

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Types Definitions

### [STM32L476G-DISCOVERY Common](#)

---

This file provides firmware functions to manage Leds, push-buttons, COM ports, SD card on SPI and temperature sensor (TS751) available on STM32L476G-DISCOVERY discoveryuaton board from STMicroelectronics. [More...](#)

This file provides firmware functions to manage Leds, push-buttons, COM ports, SD card on SPI and temperature sensor (TS751) available on STM32L476G-DISCOVERY discoveryuaton board from STMicroelectronics.

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Macros

STM32L476G-DISCOVERY Common

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Types

STM32L476G-DISCOVERY ACCELEROMETER

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Constants

STM32L476G-DISCOVERY ACCELEROMETER

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Macros

STM32L476G-DISCOVERY ACCELEROMETER

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Functions

STM32L476G-DISCOVERY ACCELEROMETER

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Exported Types

STM32L476G-DISCOVERY ACCELEROMETER

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Exported Macros

STM32L476G-DISCOVERY ACCELEROMETER

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Data Structures</a>	<a href="#">Data Structure Index</a>	<a href="#">Data Fields</a>		

[Data Fields](#)

## AUDIO\_OUT\_TypeDef Struct Reference

[Private Types](#)

---

Data Fields

AUDIO_DrvTypeDef *	AudioDrv
Audio_CallbackTypeDef	CbError
Audio_CallbackTypeDef	CbHalfTransfer
Audio_CallbackTypeDef	CbTransferComplete



## Detailed Description

Definition at line **148** of file [stm32l476g\\_discovery\\_audio.c](#).

---

## Field Documentation

### **AUDIO\_DrvTypeDef\* AUDIO\_OUT\_TypeDef::AudioDrv**

Definition at line **150** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **AUDIO\_CODEC\_Reset()**,  
**BSP\_AUDIO\_OUT\_DeInit()**, **BSP\_AUDIO\_OUT\_Init()**,  
**BSP\_AUDIO\_OUT\_Pause()**, **BSP\_AUDIO\_OUT\_Play()**,  
**BSP\_AUDIO\_OUT\_Resume()**, **BSP\_AUDIO\_OUT\_SetMute()**,  
**BSP\_AUDIO\_OUT\_SetOutputMode()**,  
**BSP\_AUDIO\_OUT\_SetVolume()**, and **BSP\_AUDIO\_OUT\_Stop()**.

### **Audio\_CallbackTypeDef AUDIO\_OUT\_TypeDef::CbError**

Definition at line **151** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_OUT\_Init()**,  
**BSP\_AUDIO\_OUT\_RegisterCallbacks()**, and  
**HAL\_SAI\_ErrorCallback()**.

### **Audio\_CallbackTypeDef AUDIO\_OUT\_TypeDef::CbHalfTransfer**

Definition at line **152** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_OUT\_Init()**,  
**BSP\_AUDIO\_OUT\_RegisterCallbacks()**, and  
**HAL\_SAI\_TxHalfCpltCallback()**.

### **Audio\_CallbackTypeDef AUDIO\_OUT\_TypeDef::CbTransferComple**

Definition at line **153** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_OUT\_Init()**,  
**BSP\_AUDIO\_OUT\_RegisterCallbacks()**, and  
**HAL\_SAI\_TxCpltCallback()**.

---

The documentation for this struct was generated from the following file:

- **stm32l476g\_discovery\_audio.c**
- 

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Data Structures</a>	<a href="#">Data Structure Index</a>	<a href="#">Data Fields</a>		

[Data Fields](#)

## AUDIO\_IN\_TypeDef Struct Reference

[Private Types](#)

---

## Data Fields

DFSDM_Channel_HandleTypeDef	<b>hDfsdmLeftChannel</b>
DMA_HandleTypeDef	<b>hDmaDfsdmLeft</b>
int32_t *	<b>LeftRecBuff</b>
uint32_t	<b>Frequency</b>
uint32_t	<b>BitResolution</b>
uint32_t	<b>ChannelNbr</b>
uint16_t *	<b>pRecBuf</b>
uint32_t	<b>RecSize</b>
<b>Audio_CallbackTypeDef</b>	<b>CbError</b>
<b>Audio_CallbackTypeDef</b>	<b>CbHalfTransfer</b>
<b>Audio_CallbackTypeDef</b>	<b>CbTransferComplete</b>

## Detailed Description

Definition at line **156** of file [stm32l476g\\_discovery\\_audio.c](#).

---

## Field Documentation

### **uint32\_t AUDIO\_IN\_TypeDef::BitResolution**

Definition at line **162** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Init()**.

### **Audio\_CallbackTypeDef AUDIO\_IN\_TypeDef::CbError**

Definition at line **166** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Init()**,  
**BSP\_AUDIO\_IN\_RegisterCallbacks()**, and  
**HAL\_DFSDM\_FilterErrorCallback()**.

### **Audio\_CallbackTypeDef AUDIO\_IN\_TypeDef::CbHalfTransfer**

Definition at line **167** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Init()**,  
**BSP\_AUDIO\_IN\_RegisterCallbacks()**, and  
**HAL\_DFSDM\_FilterRegConvHalfCpltCallback()**.

### **Audio\_CallbackTypeDef AUDIO\_IN\_TypeDef::CbTransferComplete**

Definition at line **168** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Init()**,  
**BSP\_AUDIO\_IN\_RegisterCallbacks()**, and  
**HAL\_DFSDM\_FilterRegConvCpltCallback()**.

## **uint32\_t AUDIO\_IN\_TypeDef::ChannelNbr**

Definition at line **163** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Init()**.

## **uint32\_t AUDIO\_IN\_TypeDef::Frequency**

Definition at line **161** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Init()**.

## **DFSDM\_Channel\_HandleTypeDef AUDIO\_IN\_TypeDef::hDfsdmLeft**

Definition at line **158** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **AUDIO\_DFSDMx\_DeInit()**, and **AUDIO\_DFSDMx\_Init()**.

## **DMA\_HandleTypeDef AUDIO\_IN\_TypeDef::hDmaDfsdmLeft**

Definition at line **159** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **HAL\_DFSDM\_FilterMspDeInit()**, and **HAL\_DFSDM\_FilterMspInit()**.

## **int32\_t\* AUDIO\_IN\_TypeDef::LeftRecBuff**

Definition at line **160** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Record()**, **BSP\_AUDIO\_IN\_Resume()**, **BSP\_AUDIO\_IN\_Stop()**, **HAL\_DFSDM\_FilterRegConvCpltCallback()**, and



**HAL\_DFSDM\_FilterRegConvHalfCpltCallback().**

**uint16\_t\* AUDIO\_IN\_TypeDef::pRecBuf**

Definition at line **164** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Record()**,  
**HAL\_DFSDM\_FilterRegConvCpltCallback()**, and  
**HAL\_DFSDM\_FilterRegConvHalfCpltCallback()**.

**uint32\_t AUDIO\_IN\_TypeDef::RecSize**

Definition at line **165** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **BSP\_AUDIO\_IN\_Record()**,  
**BSP\_AUDIO\_IN\_Resume()**,  
**HAL\_DFSDM\_FilterRegConvCpltCallback()**, and  
**HAL\_DFSDM\_FilterRegConvHalfCpltCallback()**.

The documentation for this struct was generated from the following file:

- **stm32l476g\_discovery\_audio.c**

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Constants

[STM32L476G-DISCOVERY AUDIO](#)

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Types

STM32L476G-DISCOVERY COMPASS

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Constants

STM32L476G-DISCOVERY COMPASS

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Macros

STM32L476G-DISCOVERY COMPASS

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Exported Types

STM32L476G-DISCOVERY COMPASS

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Exported Macros

STM32L476G-DISCOVERY COMPASS

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Functions

STM32L476G-DISCOVERY COMPASS

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Types

STM32L476G-DISCOVERY GYROSCOPE

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Constants

STM32L476G-DISCOVERY GYROSCOPE

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Macros

STM32L476G-DISCOVERY GYROSCOPE

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Functions

STM32L476G-DISCOVERY GYROSCOPE

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Exported Constants

STM32L476G-DISCOVERY GYROSCOPE

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Exported Macros

STM32L476G-DISCOVERY GYROSCOPE

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Defines

STM32L476G-DISCOVERY IDD

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Private Functions

STM32L476G-DISCOVERY IDD

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Data Structures</a>	<a href="#">Data Structure Index</a>	<a href="#">Data Fields</a>		

[Data Fields](#)

## QSPI\_Info Struct Reference

[Exported Types](#)

```
#include <stm32l476g_discovery_qspi.h>
```

Data Fields

uint32_t	FlashSize
uint32_t	EraseSectorSize
uint32_t	EraseSectorsNumber
uint32_t	ProgPageSize
uint32_t	ProgPagesNumber

## Detailed Description

Definition at line **83** of file [stm32l476g\\_discovery\\_qspi.h](#).

---

## Field Documentation

### **uint32\_t** **QSPI\_Info::EraseSectorSize**

Size of sectors for the erase operation

Definition at line **85** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_GetInfo()**.

### **uint32\_t** **QSPI\_Info::EraseSectorsNumber**

Number of sectors for the erase operation

Definition at line **86** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_GetInfo()**.

### **uint32\_t** **QSPI\_Info::FlashSize**

Size of the flash

Definition at line **84** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_GetInfo()**.

### **uint32\_t** **QSPI\_Info::ProgPageSize**

Size of pages for the program operation

Definition at line **87** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_GetInfo()**.

## **uint32\_t QSPI\_Info::ProgPagesNumber**

Number of pages for the program operation

Definition at line **88** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_GetInfo()**.

---

The documentation for this struct was generated from the following file:

- **stm32l476g\_discovery\_qspi.h**

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Data Structures</a>	<a href="#">Data Structure Index</a>	<a href="#">Data Fields</a>		
<a href="#">All</a>	<a href="#">Variables</a>			

Here is a list of all struct and union fields with links to the structures/unions they belong to:

- AudioDrv : [AUDIO\\_OUT\\_TypeDef](#)
- BitResolution : [AUDIO\\_IN\\_TypeDef](#)
- CbError : [AUDIO\\_OUT\\_TypeDef](#) , [AUDIO\\_IN\\_TypeDef](#)
- CbHalfTransfer : [AUDIO\\_IN\\_TypeDef](#) , [AUDIO\\_OUT\\_TypeDef](#)
- CbTransferComplete : [AUDIO\\_IN\\_TypeDef](#) , [AUDIO\\_OUT\\_TypeDef](#)
- ChannelNbr : [AUDIO\\_IN\\_TypeDef](#)
- EraseSectorSize : [QSPI\\_Info](#)
- EraseSectorsNumber : [QSPI\\_Info](#)
- FlashSize : [QSPI\\_Info](#)
- Frequency : [AUDIO\\_IN\\_TypeDef](#)
- hDfsdmLeftChannel : [AUDIO\\_IN\\_TypeDef](#)
- hDmaDfsdmLeft : [AUDIO\\_IN\\_TypeDef](#)
- LeftRecBuff : [AUDIO\\_IN\\_TypeDef](#)
- pRecBuf : [AUDIO\\_IN\\_TypeDef](#)
- ProgPageSize : [QSPI\\_Info](#)
- ProgPagesNumber : [QSPI\\_Info](#)
- RecSize : [AUDIO\\_IN\\_TypeDef](#)

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Data Structures</a>	<a href="#">Data Structure Index</a>	<a href="#">Data Fields</a>		
<a href="#">All</a>	<a href="#">Variables</a>			

- AudioDrv : [AUDIO\\_OUT\\_TypeDef](#)
- BitResolution : [AUDIO\\_IN\\_TypeDef](#)
- CbError : [AUDIO\\_OUT\\_TypeDef](#) , [AUDIO\\_IN\\_TypeDef](#)
- CbHalfTransfer : [AUDIO\\_IN\\_TypeDef](#) , [AUDIO\\_OUT\\_TypeDef](#)
- CbTransferComplete : [AUDIO\\_IN\\_TypeDef](#) , [AUDIO\\_OUT\\_TypeDef](#)
- ChannelNbr : [AUDIO\\_IN\\_TypeDef](#)
- EraseSectorSize : [QSPI\\_Info](#)
- EraseSectorsNumber : [QSPI\\_Info](#)
- FlashSize : [QSPI\\_Info](#)
- Frequency : [AUDIO\\_IN\\_TypeDef](#)
- hDfsdmLeftChannel : [AUDIO\\_IN\\_TypeDef](#)
- hDmaDfsdmLeft : [AUDIO\\_IN\\_TypeDef](#)
- LeftRecBuf : [AUDIO\\_IN\\_TypeDef](#)
- pRecBuf : [AUDIO\\_IN\\_TypeDef](#)
- ProgPageSize : [QSPI\\_Info](#)
- ProgPagesNumber : [QSPI\\_Info](#)
- RecSize : [AUDIO\\_IN\\_TypeDef](#)

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files											
Directories																				
File List			Globals																	
All	Functions		Variables			Typedefs		Enumerations			Enumerator									
Defines																				
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w		

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- \_ -

- `__SPI_DIRECTION_1LINE_RX` : [stm32l476g\\_discovery.c](#)
- `__SPI_DIRECTION_1LINE_TX` : [stm32l476g\\_discovery.c](#)
- `__SPI_DIRECTION_2LINES` : [stm32l476g\\_discovery.c](#)
- `__SPI_DIRECTION_2LINES_RXONLY` : [stm32l476g\\_discovery.c](#)
- `__STM32L476G_DISCOVERY_BSP_VERSION` : [stm32l476g\\_discovery.c](#)
- `__STM32L476G_DISCOVERY_BSP_VERSION_MAIN` : [stm32l476g\\_discovery.c](#)
- `__STM32L476G_DISCOVERY_BSP_VERSION_RC` : [stm32l476g\\_discovery.c](#)
- `__STM32L476G_DISCOVERY_BSP_VERSION_SUB1` : [stm32l476g\\_discovery.c](#)
- `__STM32L476G_DISCOVERY_BSP_VERSION_SUB2` : [stm32l476g\\_discovery.c](#)



# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files													
Directories																			
File List		Globals																	
All	Functions		Variables		Typedefs		Enumerations		Enumerator										
Defines																			
—	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w	

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- a -

- ACCELERO\_CS\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_HIGH : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_LOW : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_PIN : [stm32l476g\\_discovery.h](#)
- ACCELERO\_ERROR : [stm32l476g\\_discovery\\_accelerometer.h](#)
- ACCELERO\_IO\_DeInit() : [stm32l476g\\_discovery.c](#) , [stm32l476g\\_discovery\\_compass.c](#)
- ACCELERO\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- ACCELERO\_IO\_ITConfig() : [stm32l476g\\_discovery.c](#)
- ACCELERO\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- ACCELERO\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- ACCELERO\_OK : [stm32l476g\\_discovery\\_accelerometer.h](#)
- ACCELERO\_StatusTypeDef : [stm32l476g\\_discovery\\_accelerometer.h](#)
- ACCELERO\_TIMEOUT :

- [stm32l476g\\_discovery\\_accelerometer.h](#)
- ACCELERO\_XLINT\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_PIN : [stm32l476g\\_discovery.h](#)
- AccelerometerDrv : [stm32l476g\\_discovery\\_accelerometer.c](#) , [stm32l476g\\_discovery\\_compass.c](#)
- ASCII\_CHAR\_0 : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- ASCII\_CHAR\_APOSTROPHE : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- ASCII\_CHAR\_AT\_SYMBOL : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- ASCII\_CHAR\_LEFT\_OPEN\_BRACE : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- ASCII\_CHAR\_LEFT\_OPEN\_BRACKET : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- Audio\_CallbackTypeDef : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_CODEC\_Reset() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_DFSDM\_DMAX\_LEFT\_IRQHandler : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_AF : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_PORT : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CLK\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CLK\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DeInit() : [stm32l476g\\_discovery\\_audio.c](#)

- AUDIO\_DFSDMx\_DMAx\_CLK\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAx\_CLK\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAx\_LEFT\_CHANNEL : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAx\_LEFT\_IRQ : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAx\_MEM\_DATA\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAx\_PERIPH\_DATA\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMIC\_DATIN\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_Init() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_DFSDMx\_LEFT\_CHANNEL : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_LEFT\_FILTER : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_ERROR : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_I2C\_ADDRESS : [stm32l476g\\_discovery.h](#)
- AUDIO\_IN\_IRQ\_PREPRIO : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_IO\_DeInit() : [stm32l476g\\_discovery.c](#)
- AUDIO\_IO\_Delay() : [stm32l476g\\_discovery.c](#)
- AUDIO\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- AUDIO\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- AUDIO\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- AUDIO\_OK : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_OUT\_IRQ\_PREPRIO : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_RESET\_GPIO : [stm32l476g\\_discovery.h](#)
- AUDIO\_RESET\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- AUDIO\_RESET\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- AUDIO\_RESET\_PIN : [stm32l476g\\_discovery.h](#)
- AUDIO\_SAIPLLConfig() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_SAIx : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_CLK\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_CLK\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)

- AUDIO\_SAIx\_DeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_SAIx\_DMAX\_CHANNEL : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_CLK\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_CLK\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_IRQ : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_IRQHandler : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_MEM\_DATA\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_PERIPH\_DATA\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_FS\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_Init() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_SAIx\_MCK\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_AF : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_GPIO\_PORT : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_PLL\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_SCK\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_SD\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_TIMEOUT : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIODATA\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures				Files											
Directories																			
File List		Globals																	
All	Functions		Variables		Typedefs		Enumerations		Enumerator										
Defines																			
—	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w	

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- b -

- BarId\_TypeDef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERY\_DETECTION\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- BATTERY\_DETECTION\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- BATTERY\_DETECTION\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- BATTERY\_DETECTION\_PIN : [stm32l476g\\_discovery.h](#)
- BATTERYLEVEL\_1\_2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERYLEVEL\_1\_4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERYLEVEL\_3\_4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERYLEVEL\_FULL : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERYLEVEL\_OFF : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BatteryLevel\_TypeDef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- bLCDGlass\_KeyPressed : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_ACCELERO\_GetXYZ() : [stm32l476g\\_discovery\\_accelerometer.c](#)
- BSP\_ACCELERO\_Init() : [stm32l476g\\_discovery\\_accelerometer.c](#)
- BSP\_ACCELERO\_Reset() :

[stm32l476g\\_discovery\\_accelerometer.c](#)

- BSP\_AUDIO\_FREQUENCY\_11K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_16K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_22K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_32K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_44K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_48K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_8K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_96K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_hDfsdmLeftFilter : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_hSai : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_DeInit() : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Init() : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Pause() : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Record() : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_RegisterCallbacks() : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Resume() : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_SetFrequency() : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)

- BSP\_AUDIO\_IN\_Stop() : [stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_ChangeAudioConfig() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_ChangeBuffer() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_CIRCULARMODE :  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_DeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_Init() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_MONOMODE :  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_NORMALMODE :  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_Pause() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_Play() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_RegisterCallbacks() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_Resume() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_SetFrequency() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_SetMute() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_SetOutputMode() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_SetVolume() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_STEREO MODE :  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_Stop() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_COMPASS\_AccGetXYZ() :  
[stm32l476g\\_discovery\\_compass.c](#)
- BSP\_COMPASS\_DeInit() : [stm32l476g\\_discovery\\_compass.c](#)
- BSP\_COMPASS\_Init() : [stm32l476g\\_discovery\\_compass.c](#)
- BSP\_COMPASS\_LowPower() :  
[stm32l476g\\_discovery\\_compass.c](#)
- BSP\_COMPASS\_MagGetXYZ() :  
[stm32l476g\\_discovery\\_compass.c](#)



- BSP\_GetVersion() : [stm32l476g\\_discovery.c](#)
- BSP\_GYRO\_DeInit() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_DisableIT() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_EnableIT() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_GetXYZ() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_Init() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_ITConfig() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_LowPower() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_ReadID() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_Reset() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_IDD\_ClearIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_Config() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_DeInit() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_DisableIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_EnableIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorClearIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorDisableIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorEnableIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorGetCode() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorGetITStatus() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_GetITStatus() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_GetValue() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_Init() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_LowPower() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_Reset() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_StartMeasure() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_WakeUp() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_JOY\_DeInit() : [stm32l476g\\_discovery.c](#)
- BSP\_JOY\_GetState() : [stm32l476g\\_discovery.c](#)
- BSP\_JOY\_Init() : [stm32l476g\\_discovery.c](#)
- BSP\_LCD\_GLASS\_BarLevelConfig() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_BlinkConfig() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_Clear() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_ClearBar() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)



- BSP\_LCD\_GLASS\_Contrast() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DeInit() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DisplayBar() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DisplayChar() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DisplayStrDeci() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DisplayString() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_Init() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_ScrollSentence() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LED\_DeInit() : [stm32l476g\\_discovery.c](#)
- BSP\_LED\_Init() : [stm32l476g\\_discovery.c](#)
- BSP\_LED\_Off() : [stm32l476g\\_discovery.c](#)
- BSP\_LED\_On() : [stm32l476g\\_discovery.c](#)
- BSP\_LED\_Toggle() : [stm32l476g\\_discovery.c](#)
- BSP\_QSPI\_DeInit() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_EnableMemoryMappedMode() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Erase\_Block() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Erase\_Chip() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Erase\_Sector() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_GetInfo() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_GetStatus() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Init() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Read() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_ResumeErase() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_SuspendErase() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Write() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_SupplyModeDetection() : [stm32l476g\\_discovery.c](#)

BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files													
Directories																			
File List		Globals																	
All	Functions		Variables		Typedefs		Enumerations		Enumerator										
Defines																			
—	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w	

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- c -

- C\_CLOSEPARMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_DMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_FULL : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_MINUS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_MMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_NMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_OPENPARMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_PERCENT\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_PERCENT\_2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_PLUS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_SLATCH : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_STAR : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_UMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- CapLetterMap : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- CODEC\_AUDIO\_POWER\_OFF : [stm32l476g\\_discovery.h](#)
- CODEC\_AUDIO\_POWER\_ON : [stm32l476g\\_discovery.h](#)
- CODEC\_RESET\_DELAY : [stm32l476g\\_discovery\\_audio.h](#)
- COM\_PER\_DIGIT\_NB : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- COMPASS\_ERROR : [stm32l476g\\_discovery\\_compass.h](#)

- COMPASS\_OK : [stm32l476g\\_discovery\\_compass.h](#)
- COMPASS\_StatusTypeDef : [stm32l476g\\_discovery\\_compass.h](#)
- COMPASS\_TIMEOUT : [stm32l476g\\_discovery\\_compass.h](#)
- Convert() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files													
Directories																			
File List		Globals																	
All	Functions		Variables		Typedefs		Enumerations		Enumerator										
Defines																			
—	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w	

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- d -

- DEFAULT\_AUDIO\_IN\_BIT\_RESOLUTION : [stm32l476g\\_discovery\\_audio.h](#)
- DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR : [stm32l476g\\_discovery\\_audio.h](#)
- DEFAULT\_AUDIO\_IN\_FREQ : [stm32l476g\\_discovery\\_audio.h](#)
- DEFAULT\_AUDIO\_IN\_VOLUME : [stm32l476g\\_discovery\\_audio.h](#)
- DFSDMClockDivider : [stm32l476g\\_discovery\\_audio.c](#)
- DFSDMFilterOrder : [stm32l476g\\_discovery\\_audio.c](#)
- DFSDMOverSampling : [stm32l476g\\_discovery\\_audio.c](#)
- DFSDMRightBitShift : [stm32l476g\\_discovery\\_audio.c](#)
- Digit : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- DigitPosition\_Typedef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- DISCOVERY\_I2C1 : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_ER\_IRQHandler : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_ER\_IRQn : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_EV\_IRQHandler : [stm32l476g\\_discovery.h](#)

- DISCOVERY\_I2C1\_EV\_IRQn : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_FORCE\_RESET : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_RELEASE\_RESET :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_GPIO\_CLK\_DISABLE :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_GPIO\_CLK\_ENABLE :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_GPIO\_PORT :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_SDA\_AF : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SDA\_GPIO\_CLK\_DISABLE :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SDA\_GPIO\_CLK\_ENABLE :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SDA\_GPIO\_PORT :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SDA\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_TIMEOUT\_MAX : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2 : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_ER\_IRQn : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_EV\_IRQn : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_FORCE\_RESET : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_RELEASE\_RESET :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_GPIO\_CLK\_DISABLE :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_GPIO\_CLK\_ENABLE :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_GPIO\_PORT :  
[stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_SDA\_AF : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SDA\_GPIO\_CLK\_DISABLE :

- [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SDA\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SDA\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SDA\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_TIMEOUT\_MAX : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_IDD\_AMPLI\_GAIN : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT0\_STABDELAY : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT0\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT1\_STABDELAY : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT1\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT2\_STABDELAY : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT2\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT4\_STABDELAY : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT4\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_VDD\_MIN : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_SPIx : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_AF : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_CLOCK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_CLOCK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_FORCE\_RESET : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_RELEASE\_RESET :

### **stm32l476g\_discovery.h**

- DISCOVERY\_SPIx\_MISO\_PIN : **stm32l476g\_discovery.h**
- DISCOVERY\_SPIx\_MOSI\_PIN : **stm32l476g\_discovery.h**
- DISCOVERY\_SPIx\_SCK\_PIN : **stm32l476g\_discovery.h**
- DMA\_MAX : **stm32l476g\_discovery\_audio.h**
- DMA\_MAX\_SIZE : **stm32l476g\_discovery\_audio.h**
- DOT : **stm32l476g\_discovery\_glass\_lcd.h**
- DOUBLE\_DOT : **stm32l476g\_discovery\_glass\_lcd.h**
- DOUBLEPOINT\_OFF : **stm32l476g\_discovery\_glass\_lcd.h**
- DOUBLEPOINT\_ON : **stm32l476g\_discovery\_glass\_lcd.h**
- DoublePoint\_Typedef : **stm32l476g\_discovery\_glass\_lcd.h**
- DOWN\_JOY\_EXTI\_IRQn : **stm32l476g\_discovery.h**
- DOWN\_JOY\_GPIO\_CLK\_DISABLE : **stm32l476g\_discovery.h**
- DOWN\_JOY\_GPIO\_CLK\_ENABLE : **stm32l476g\_discovery.h**
- DOWN\_JOY\_GPIO\_PORT : **stm32l476g\_discovery.h**
- DOWN\_JOY\_PIN : **stm32l476g\_discovery.h**
- DUMMY\_BYTE : **stm32l476g\_discovery.h**



# STM32L476G-Discovery BSP User Manual

Main Page				Modules				Data Structures				Files									
Directories																					
File List				Globals																	
All		Functions			Variables			Typedefs			Enumerations			Enumerator							
Defines																					
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w			

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- e -

- EEPROM\_I2C\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- EEPROM\_I2C\_IO\_IsDeviceReady() : [stm32l476g\\_discovery.c](#)
- EEPROM\_I2C\_IO\_ReadData() : [stm32l476g\\_discovery.c](#)
- EEPROM\_I2C\_IO\_WriteData() : [stm32l476g\\_discovery.c](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files											
Directories																				
File List			Globals																	
All	Functions		Variables		Typedefs		Enumerations		Enumerator											
Defines																				
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w		

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- g -

- GYRO\_CS\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_HIGH : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_LOW : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_PIN : [stm32l476g\\_discovery.h](#)
- GYRO\_ERROR : [stm32l476g\\_discovery\\_gyroscope.h](#)
- GYRO\_INT1\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_PIN : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_PIN : [stm32l476g\\_discovery.h](#)
- GYRO\_IO\_DeInit() : [stm32l476g\\_discovery.c](#)
- GYRO\_IO\_Init() : [stm32l476g\\_discovery.c](#)

- GYRO\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- GYRO\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- GYRO\_OK : [stm32l476g\\_discovery\\_gyroscope.h](#)
- GYRO\_StatusTypeDef : [stm32l476g\\_discovery\\_gyroscope.h](#)
- GYRO\_TIMEOUT : [stm32l476g\\_discovery\\_gyroscope.h](#)
- GyroscopeDrv : [stm32l476g\\_discovery\\_gyroscope.c](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files											
Directories																				
File List			Globals																	
All	Functions		Variables		Typedefs		Enumerations		Enumerator											
Defines																				
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w		

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

## - h -

- HAL\_DFSDM\_ChannelMspDeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_ChannelMspInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterErrorCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterMspDeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterMspInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterRegConvCpltCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterRegConvHalfCpltCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_ErrorCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_MspDeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_MspInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_TxCpltCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_TxHalfCpltCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- hAudioIn : [stm32l476g\\_discovery\\_audio.c](#)
- hAudioOut : [stm32l476g\\_discovery\\_audio.c](#)

- hDmaSai : [stm32l476g\\_discovery\\_audio.c](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files											
Directories																				
File List			Globals																	
All	Functions		Variables			Typedefs		Enumerations			Enumerator									
Defines																				
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w		

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- i -

- I2C1\_DeInit() : [stm32l476g\\_discovery.c](#)
- I2C1\_Error() : [stm32l476g\\_discovery.c](#)
- I2C1\_Init() : [stm32l476g\\_discovery.c](#)
- I2C1\_MspDeInit() : [stm32l476g\\_discovery.c](#)
- I2C1\_MspInit() : [stm32l476g\\_discovery.c](#)
- I2C1\_ReadBuffer() : [stm32l476g\\_discovery.c](#)
- I2C1\_WriteBuffer() : [stm32l476g\\_discovery.c](#)
- I2c1Handle : [stm32l476g\\_discovery.c](#)
- I2c1Timeout : [stm32l476g\\_discovery.c](#)
- I2C2\_DeInit() : [stm32l476g\\_discovery.c](#)
- I2C2\_Error() : [stm32l476g\\_discovery.c](#)
- I2C2\_Init() : [stm32l476g\\_discovery.c](#)
- I2C2\_MspDeInit() : [stm32l476g\\_discovery.c](#)
- I2C2\_MspInit() : [stm32l476g\\_discovery.c](#)
- I2C2\_ReadBuffer() : [stm32l476g\\_discovery.c](#)
- I2C2\_ReadData() : [stm32l476g\\_discovery.c](#)
- I2C2\_WriteBuffer() : [stm32l476g\\_discovery.c](#)
- I2C2\_WriteData() : [stm32l476g\\_discovery.c](#)
- I2c2Handle : [stm32l476g\\_discovery.c](#)

- I2c2Timeout : [stm32l476g\\_discovery.c](#)
- IDD\_ERROR : [stm32l476g\\_discovery\\_idd.h](#)
- IDD\_I2C\_ADDRESS : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_PIN : [stm32l476g\\_discovery.h](#)
- IDD\_OK : [stm32l476g\\_discovery\\_idd.h](#)
- IDD\_StatusTypeDef : [stm32l476g\\_discovery\\_idd.h](#)
- IDD\_TIMEOUT : [stm32l476g\\_discovery\\_idd.h](#)
- IDD\_WAKEUP\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- IDD\_WAKEUP\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- IDD\_WAKEUP\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- IDD\_WAKEUP\_PIN : [stm32l476g\\_discovery.h](#)
- IDD\_ZERO\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- IddDrv : [stm32l476g\\_discovery\\_idd.c](#)
- IOE\_Delay() : [stm32l476g\\_discovery.c](#)
- IOE\_Init() : [stm32l476g\\_discovery.c](#)
- IOE\_ITConfig() : [stm32l476g\\_discovery.c](#)
- IOE\_Read() : [stm32l476g\\_discovery.c](#)
- IOE\_ReadMultiple() : [stm32l476g\\_discovery.c](#)
- IOE\_Write() : [stm32l476g\\_discovery.c](#)

# STM32L476G-Discovery BSP User Manual

Main Page				Modules				Data Structures				Files																													
Directories																																									
File List				Globals																																					
All		Functions				Variables				Typedefs				Enumerations				Enumerator																							
Defines																																									
_		a		b		c		d		e		g		h		i		j		l		m		n		p		q		r		s		u		w					

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- j -

- JOY\_ALL\_PINS : [stm32l476g\\_discovery.h](#)
  - JOY\_DOWN : [stm32l476g\\_discovery.h](#)
  - JOY\_IRQn : [stm32l476g\\_discovery.c](#)
  - JOY\_LEFT : [stm32l476g\\_discovery.h](#)
  - JOY\_MODE\_EXTI : [stm32l476g\\_discovery.h](#)
  - JOY\_MODE\_GPIO : [stm32l476g\\_discovery.h](#)
  - JOY\_NONE : [stm32l476g\\_discovery.h](#)
  - JOY\_PIN : [stm32l476g\\_discovery.c](#)
  - JOY\_PORT : [stm32l476g\\_discovery.c](#)
  - JOY\_RIGHT : [stm32l476g\\_discovery.h](#)
  - JOY\_SEL : [stm32l476g\\_discovery.h](#)
  - JOY\_UP : [stm32l476g\\_discovery.h](#)
  - JOYMode\_TypeDef : [stm32l476g\\_discovery.h](#)
  - JOYn : [stm32l476g\\_discovery.h](#)
  - JOYState\_TypeDef : [stm32l476g\\_discovery.h](#)
  - JOYx\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
  - JOYx\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
-



Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page				Modules				Data Structures				Files																													
Directories																																									
File List				Globals																																					
All		Functions				Variables				Typedefs				Enumerations				Enumerator																							
Defines																																									
_		a		b		c		d		e		g		h		i		j		l		m		n		p		q		r		s		u		w					

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- l -

- LCD\_BAR0\_2\_COM : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR0\_2\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR0\_SEG : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR1\_3\_COM : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR1\_3\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR1\_SEG : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR2\_SEG : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR3\_SEG : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_NONE : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM0\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM1\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- LCD\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM2\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM3\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT2\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT2\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT2\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT2\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT2\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT2\_COM2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT2\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT2\_COM3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT3\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT3\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT3\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT3\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT3\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT3\_COM2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- LCD\_DIGIT3\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT3\_COM3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM0\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM0\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM1\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM1\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM2\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM2\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM3\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM3\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM0\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM0\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM1\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM1\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- LCD\_DIGIT5\_COM1\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM2\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM2\_1\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM2\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM3\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM3\_1\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM3\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM0\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM1\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM2\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM3\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_MAX\_NUMBER : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_5 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_6 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_GPIO\_BANKA\_PINS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_GPIO\_BANKB\_PINS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_GPIO\_BANKC\_PINS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- LCD\_GPIO\_BANKD\_PINS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_1ST\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_1ST\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_2ND\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_2ND\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_3RD\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_3RD\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_4TH\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_4TH\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM1\_SEG\_1ST\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM1\_SEG\_1ST\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM1\_SEG\_2ND\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM1\_SEG\_2ND\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM1\_SEG\_3RD\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM1\_SEG\_3RD\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM1\_SEG\_4TH\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM1\_SEG\_4TH\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM2\_SEG\_1ST\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM2\_SEG\_1ST\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- LCD\_MAP\_CHAR\_COM2\_SEG\_2ND\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM2\_SEG\_2ND\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM2\_SEG\_3RD\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM2\_SEG\_3RD\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM2\_SEG\_4TH\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM2\_SEG\_4TH\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_1ST\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_1ST\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_2ND\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_2ND\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_3RD\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_3RD\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_4TH\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_4TH\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MspDeInit() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- LCD\_MspInit() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- LCD\_SEG0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG0\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG10 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG10\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG11 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG11\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)



- LCD\_SEG12 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG12\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG13 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG13\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG14 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG14\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG15 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG15\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG16 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG16\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG17 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG17\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG18 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG18\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG19 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG19\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG1\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG20 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG20\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG21 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG21\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG22 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG22\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG23 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG23\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG2\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG3\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG4\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG5 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG5\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG6 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG6\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG7 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG7\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)



- LCD\_SEG8 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG8\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG9 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG9\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCDBar : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- LCDHandle : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- LED4 : [stm32l476g\\_discovery.h](#)
- LED4\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- LED4\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- LED4\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- LED4\_PIN : [stm32l476g\\_discovery.h](#)
- LED5 : [stm32l476g\\_discovery.h](#)
- LED5\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- LED5\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- LED5\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- LED5\_PIN : [stm32l476g\\_discovery.h](#)
- LED\_GREEN : [stm32l476g\\_discovery.h](#)
- LED\_PIN : [stm32l476g\\_discovery.c](#)
- LED\_PORT : [stm32l476g\\_discovery.c](#)
- LED\_RED : [stm32l476g\\_discovery.h](#)
- Led\_TypeDef : [stm32l476g\\_discovery.h](#)
- LEDn : [stm32l476g\\_discovery.h](#)
- LEDx\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- LEDx\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_PIN : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page				Modules				Data Structures				Files										
Directories																						
File List				Globals																		
All		Functions				Variables				Typedefs				Enumerations				Enumerator				
Defines																						
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w				

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- m -

- MAGNETO\_CS\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_HIGH : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_LOW : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_PIN : [stm32l476g\\_discovery.h](#)
- MAGNETO\_DRDY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_DRDY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_DRDY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- MAGNETO\_DRDY\_PIN : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT1\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT1\_PIN : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- MAGNETO\_IO\_DeInit() : [stm32l476g\\_discovery.c](#) ,

### [stm32l476g\\_discovery\\_compass.c](#)

- MAGNETO\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- MAGNETO\_IO\_ITConfig() : [stm32l476g\\_discovery.c](#)
- MAGNETO\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- MAGNETO\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- MagnetoDrv : [stm32l476g\\_discovery\\_compass.c](#)
- MCU\_LCD\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM0\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM1\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM2\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM3\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM4\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM5 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM5\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM6 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM6\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM7 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM7\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG0\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG10 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG10\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG11 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG11\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG12 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG12\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG13 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG13\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG14 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG14\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG15 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG15\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- MCU\_LCD\_SEG16 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG16\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG17 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG17\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG18 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG18\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG19 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG19\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG1\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG20 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG20\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG21 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG21\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG22 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG22\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG23 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG23\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG24 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG24\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG25 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG25\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG26 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG26\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG27 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG27\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG28 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG28\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG29 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG29\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG2\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG30 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG30\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG31 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG31\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG32 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- MCU\_LCD\_SEG32\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG33 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG33\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG34 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG34\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG35 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG35\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG36 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG36\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG37 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG37\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG38 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG38\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG39\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG3\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG40\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG41\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG42\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG43\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG4\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG5 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG5\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG6 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG6\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG7 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG7\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG8 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG8\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG9 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG9\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MFX\_IO\_DeInit() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Delay() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_EnableWakeupPin() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_ITConfig() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Read() : [stm32l476g\\_discovery.c](#)

- MFX\_IO\_ReadMultiple() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Wakeup() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_WriteMultiple() : [stm32l476g\\_discovery.c](#)
- MULTIPLEBYTE\_CMD : [stm32l476g\\_discovery.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files		
Directories											
File List			Globals								
All		Functions		Variables		Typedefs		Enumerations		Enumerator	
Defines											
<div><div></div><div>a</div><div>b</div><div>c</div><div>d</div><div>e</div><div>g</div><div>h</div><div>i</div><div>j</div><div>l</div><div>m</div><div>n</div><div>p</div><div>q</div><div>r</div><div>s</div><div>u</div><div>w</div></div>											

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

**- n -**

- NumberMap : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)



# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files											
Directories																				
File List			Globals																	
All	Functions		Variables			Typedefs		Enumerations			Enumerator									
Defines																				
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w		

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- p -

- POINT\_OFF : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- POINT\_ON : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- Point\_Typedef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)



# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files											
Directories																				
File List			Globals																	
All	Functions		Variables			Typedefs		Enumerations			Enumerator									
Defines																				
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w		

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- q -

- [QSPI\\_AutoPollingMemReady\(\)](#) : [stm32l476g\\_discovery\\_qspi.c](#)
- [QSPI\\_BUSY](#) : [stm32l476g\\_discovery\\_qspi.h](#)
- [QSPI\\_DummyCyclesCfg\(\)](#) : [stm32l476g\\_discovery\\_qspi.c](#)
- [QSPI\\_ERROR](#) : [stm32l476g\\_discovery\\_qspi.h](#)
- [QSPI\\_MspDeInit\(\)](#) : [stm32l476g\\_discovery\\_qspi.c](#)
- [QSPI\\_MspInit\(\)](#) : [stm32l476g\\_discovery\\_qspi.c](#)
- [QSPI\\_NOT\\_SUPPORTED](#) : [stm32l476g\\_discovery\\_qspi.h](#)
- [QSPI\\_OK](#) : [stm32l476g\\_discovery\\_qspi.h](#)
- [QSPI\\_ResetMemory\(\)](#) : [stm32l476g\\_discovery\\_qspi.c](#)
- [QSPI\\_SUSPENDED](#) : [stm32l476g\\_discovery\\_qspi.h](#)
- [QSPI\\_WriteEnable\(\)](#) : [stm32l476g\\_discovery\\_qspi.c](#)
- [QSPIHandle](#) : [stm32l476g\\_discovery\\_qspi.c](#)

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files											
Directories																				
File List			Globals																	
All	Functions		Variables			Typedefs		Enumerations			Enumerator									
Defines																				
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w		

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- r -

- READWRITE\_CMD : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_PIN : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files											
Directories																				
File List			Globals																	
All	Functions		Variables			Typedefs		Enumerations		Enumerator										
Defines																				
_	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w		

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- s -

- SAIClockDivider : [stm32l476g\\_discovery\\_audio.c](#)
- SaturateLH : [stm32l476g\\_discovery\\_audio.c](#)
- SCROLL\_SPEED\_HIGH : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SCROLL\_SPEED\_LOW : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SCROLL\_SPEED\_MEDIUM :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SEG\_PER\_DIGIT\_NB : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SEL\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- SEL\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- SEL\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- SEL\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- SEL\_JOY\_PIN : [stm32l476g\\_discovery.h](#)
- SpiHandle : [stm32l476g\\_discovery.c](#)
- SPIx\_DeInit() : [stm32l476g\\_discovery.c](#)
- SPIx\_Init() : [stm32l476g\\_discovery.c](#)
- SPIx\_MspDeInit() : [stm32l476g\\_discovery.c](#)
- SPIx\_MspInit() : [stm32l476g\\_discovery.c](#)
- SPIx\_Read() : [stm32l476g\\_discovery.c](#)
- SPIx\_TIMEOUT\_MAX : [stm32l476g\\_discovery.h](#)

- SPIx\_Write() : [stm32l476g\\_discovery.c](#)
- SPIx\_WriteRead() : [stm32l476g\\_discovery.c](#)
- SpixTimeout : [stm32l476g\\_discovery.c](#)
- SUPPLY\_MODE\_BATTERY : [stm32l476g\\_discovery.h](#)
- SUPPLY\_MODE\_ERROR : [stm32l476g\\_discovery.h](#)
- SUPPLY\_MODE\_EXTERNAL : [stm32l476g\\_discovery.h](#)
- SupplyMode\_TypeDef : [stm32l476g\\_discovery.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page				Modules				Data Structures				Files																									
Directories																																					
File List				Globals																																	
All		Functions				Variables				Typedefs				Enumerations				Enumerator																			
Defines																																					
_		a		b		c		d		e		g		h		i		j		l		m		n		p		q		r		s		u		w	

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- u -

- UP\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- UP\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- UP\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- UP\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- UP\_JOY\_PIN : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures				Files										
Directories																		
File List		Globals																
All	Functions		Variables		Typedefs		Enumerations		Enumerator									
Defines																		
—	a	b	c	d	e	g	h	i	j	l	m	n	p	q	r	s	u	w

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- w -

- WriteChar() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)

# STM32L476G-Discovery BSP User Manual

Main Page				Modules				Data Structures				Files											
Directories																							
File List				Globals																			
All		Functions				Variables				Typedefs				Enumerations				Enumerator					
Defines																							
a	b	c	e	g	h	i	l	m	q	s	w												

- a -

- ACCELERO\_IO\_DeInit() : [stm32l476g\\_discovery.c](#) , [stm32l476g\\_discovery\\_compass.c](#)
- ACCELERO\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- ACCELERO\_IO\_ITConfig() : [stm32l476g\\_discovery.c](#)
- ACCELERO\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- ACCELERO\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- AUDIO\_CODEC\_Reset() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_DFSDMx\_DeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_DFSDMx\_Init() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_IO\_DeInit() : [stm32l476g\\_discovery.c](#)
- AUDIO\_IO\_Delay() : [stm32l476g\\_discovery.c](#)
- AUDIO\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- AUDIO\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- AUDIO\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- AUDIO\_SAIPLLConfig() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_SAIx\_DeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- AUDIO\_SAIx\_Init() : [stm32l476g\\_discovery\\_audio.c](#)

- b -

- BSP\_ACCELERO\_GetXYZ() :  
[stm32l476g\\_discovery\\_accelerometer.c](#)
- BSP\_ACCELERO\_Init() :  
[stm32l476g\\_discovery\\_accelerometer.c](#)
- BSP\_ACCELERO\_Reset() :  
[stm32l476g\\_discovery\\_accelerometer.c](#)
- BSP\_AUDIO\_IN\_DeInit() : [stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Init() : [stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Pause() : [stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Record() : [stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_RegisterCallbacks() :  
[stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Resume() : [stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_SetFrequency() :  
[stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_IN\_Stop() : [stm32l476g\\_discovery\\_audio.c](#) ,  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_ChangeAudioConfig() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_ChangeBuffer() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_DeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_Init() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_Pause() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_Play() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_RegisterCallbacks() :  
[stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_Resume() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_SetFrequency() :  
[stm32l476g\\_discovery\\_audio.c](#)



- BSP\_AUDIO\_OUT\_SetMute() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_SetOutputMode() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_SetVolume() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_AUDIO\_OUT\_Stop() : [stm32l476g\\_discovery\\_audio.c](#)
- BSP\_COMPASS\_AccGetXYZ() : [stm32l476g\\_discovery\\_compass.c](#)
- BSP\_COMPASS\_DeInit() : [stm32l476g\\_discovery\\_compass.c](#)
- BSP\_COMPASS\_Init() : [stm32l476g\\_discovery\\_compass.c](#)
- BSP\_COMPASS\_LowPower() : [stm32l476g\\_discovery\\_compass.c](#)
- BSP\_COMPASS\_MagGetXYZ() : [stm32l476g\\_discovery\\_compass.c](#)
- BSP\_GetVersion() : [stm32l476g\\_discovery.c](#)
- BSP\_GYRO\_DeInit() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_DisableIT() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_EnableIT() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_GetXYZ() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_Init() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_ITConfig() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_LowPower() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_ReadID() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_GYRO\_Reset() : [stm32l476g\\_discovery\\_gyroscope.c](#)
- BSP\_IDD\_ClearIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_Config() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_DeInit() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_DisableIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_EnableIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorClearIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorDisableIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorEnableIT() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorGetCode() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_ErrorGetITStatus() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_GetITStatus() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_GetValue() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_Init() : [stm32l476g\\_discovery\\_idd.c](#)

- BSP\_IDD\_LowPower() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_Reset() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_StartMeasure() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_IDD\_WakeUp() : [stm32l476g\\_discovery\\_idd.c](#)
- BSP\_JOY\_DeInit() : [stm32l476g\\_discovery.c](#)
- BSP\_JOY\_GetState() : [stm32l476g\\_discovery.c](#)
- BSP\_JOY\_Init() : [stm32l476g\\_discovery.c](#)
- BSP\_LCD\_GLASS\_BarLevelConfig() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_BlinkConfig() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_Clear() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_ClearBar() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_Contrast() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DeInit() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DisplayBar() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DisplayChar() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DisplayStrDeci() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_DisplayString() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_Init() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LCD\_GLASS\_ScrollSentence() :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_LED\_DeInit() : [stm32l476g\\_discovery.c](#)
- BSP\_LED\_Init() : [stm32l476g\\_discovery.c](#)
- BSP\_LED\_Off() : [stm32l476g\\_discovery.c](#)
- BSP\_LED\_On() : [stm32l476g\\_discovery.c](#)
- BSP\_LED\_Toggle() : [stm32l476g\\_discovery.c](#)
- BSP\_QSPI\_DeInit() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_EnableMemoryMappedMode() :  
[stm32l476g\\_discovery\\_qspi.c](#)

- BSP\_QSPI\_Erase\_Block() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Erase\_Chip() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Erase\_Sector() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_GetInfo() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_GetStatus() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Init() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Read() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_ResumeErase() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_SuspendErase() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_QSPI\_Write() : [stm32l476g\\_discovery\\_qspi.c](#)
- BSP\_SupplyModeDetection() : [stm32l476g\\_discovery.c](#)

- c -

- Convert() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)

- e -

- EEPROM\_I2C\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- EEPROM\_I2C\_IO\_IsDeviceReady() : [stm32l476g\\_discovery.c](#)
- EEPROM\_I2C\_IO\_ReadData() : [stm32l476g\\_discovery.c](#)
- EEPROM\_I2C\_IO\_WriteData() : [stm32l476g\\_discovery.c](#)

- g -

- GYRO\_IO\_DeInit() : [stm32l476g\\_discovery.c](#)
- GYRO\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- GYRO\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- GYRO\_IO\_Write() : [stm32l476g\\_discovery.c](#)

- h -

- HAL\_DFSDM\_ChannelMspDeInit() :  
[stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_ChannelMspInit() :  
[stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterErrorCallback() :  
[stm32l476g\\_discovery\\_audio.c](#)

- HAL\_DFSDM\_FilterMspDeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterMspInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterRegConvCpltCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_DFSDM\_FilterRegConvHalfCpltCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_ErrorCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_MspDeInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_MspInit() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_TxCpltCallback() : [stm32l476g\\_discovery\\_audio.c](#)
- HAL\_SAI\_TxHalfCpltCallback() : [stm32l476g\\_discovery\\_audio.c](#)

- i -

- I2C1\_DeInit() : [stm32l476g\\_discovery.c](#)
- I2C1\_Error() : [stm32l476g\\_discovery.c](#)
- I2C1\_Init() : [stm32l476g\\_discovery.c](#)
- I2C1\_MspDeInit() : [stm32l476g\\_discovery.c](#)
- I2C1\_MspInit() : [stm32l476g\\_discovery.c](#)
- I2C1\_ReadBuffer() : [stm32l476g\\_discovery.c](#)
- I2C1\_WriteBuffer() : [stm32l476g\\_discovery.c](#)
- I2C2\_DeInit() : [stm32l476g\\_discovery.c](#)
- I2C2\_Error() : [stm32l476g\\_discovery.c](#)
- I2C2\_Init() : [stm32l476g\\_discovery.c](#)
- I2C2\_MspDeInit() : [stm32l476g\\_discovery.c](#)
- I2C2\_MspInit() : [stm32l476g\\_discovery.c](#)
- I2C2\_ReadBuffer() : [stm32l476g\\_discovery.c](#)
- I2C2\_ReadData() : [stm32l476g\\_discovery.c](#)
- I2C2\_WriteBuffer() : [stm32l476g\\_discovery.c](#)
- I2C2\_WriteData() : [stm32l476g\\_discovery.c](#)
- IOE\_Delay() : [stm32l476g\\_discovery.c](#)
- IOE\_Init() : [stm32l476g\\_discovery.c](#)
- IOE\_ITConfig() : [stm32l476g\\_discovery.c](#)
- IOE\_Read() : [stm32l476g\\_discovery.c](#)
- IOE\_ReadMultiple() : [stm32l476g\\_discovery.c](#)
- IOE\_Write() : [stm32l476g\\_discovery.c](#)

- l -

- LCD\_MspDeInit() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- LCD\_MspInit() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)

- m -

- MAGNETO\_IO\_DeInit() : [stm32l476g\\_discovery.c](#) , [stm32l476g\\_discovery\\_compass.c](#)
- MAGNETO\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- MAGNETO\_IO\_ITConfig() : [stm32l476g\\_discovery.c](#)
- MAGNETO\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- MAGNETO\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_DeInit() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Delay() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_EnableWakeupPin() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Init() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_ITConfig() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Read() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_ReadMultiple() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Wakeup() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_Write() : [stm32l476g\\_discovery.c](#)
- MFX\_IO\_WriteMultiple() : [stm32l476g\\_discovery.c](#)

- q -

- QSPI\_AutoPollingMemReady() : [stm32l476g\\_discovery\\_qspi.c](#)
- QSPI\_DummyCyclesCfg() : [stm32l476g\\_discovery\\_qspi.c](#)
- QSPI\_MspDeInit() : [stm32l476g\\_discovery\\_qspi.c](#)
- QSPI\_MspInit() : [stm32l476g\\_discovery\\_qspi.c](#)
- QSPI\_ResetMemory() : [stm32l476g\\_discovery\\_qspi.c](#)
- QSPI\_WriteEnable() : [stm32l476g\\_discovery\\_qspi.c](#)

- s -

- SPIx\_DeInit() : [stm32l476g\\_discovery.c](#)
- SPIx\_Init() : [stm32l476g\\_discovery.c](#)
- SPIx\_MspDeInit() : [stm32l476g\\_discovery.c](#)
- SPIx\_MspInit() : [stm32l476g\\_discovery.c](#)

- SPIx\_Read() : [stm32l476g\\_discovery.c](#)
- SPIx\_Write() : [stm32l476g\\_discovery.c](#)
- SPIx\_WriteRead() : [stm32l476g\\_discovery.c](#)

- W -

- WriteChar() : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules	Data Structures	Files	
Directories					
File List	Globals				
All	Functions	Variables	Typedefs	Enumerations	Enumerator
Defines					

- AccelerometerDrv : [stm32l476g\\_discovery\\_accelerometer.c](#) , [stm32l476g\\_discovery\\_compass.c](#)
- bLCDGlass\_KeyPressed : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- BSP\_AUDIO\_hDfsdmLeftFilter : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_hSai : [stm32l476g\\_discovery\\_audio.c](#) , [stm32l476g\\_discovery\\_audio.h](#)
- CapLetterMap : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- Digit : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- GyroscopeDrv : [stm32l476g\\_discovery\\_gyroscope.c](#)
- hAudioIn : [stm32l476g\\_discovery\\_audio.c](#)
- hAudioOut : [stm32l476g\\_discovery\\_audio.c](#)
- hDmaSai : [stm32l476g\\_discovery\\_audio.c](#)
- I2c1Handle : [stm32l476g\\_discovery.c](#)
- I2c1Timeout : [stm32l476g\\_discovery.c](#)
- I2c2Handle : [stm32l476g\\_discovery.c](#)
- I2c2Timeout : [stm32l476g\\_discovery.c](#)
- IddDrv : [stm32l476g\\_discovery\\_idd.c](#)
- JOY\_IRQn : [stm32l476g\\_discovery.c](#)
- JOY\_PIN : [stm32l476g\\_discovery.c](#)
- JOY\_PORT : [stm32l476g\\_discovery.c](#)
- LCDBar : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- LCDHandle : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)



- LED\_PIN : [stm32l476g\\_discovery.c](#)
- LED\_PORT : [stm32l476g\\_discovery.c](#)
- MagnetoDrv : [stm32l476g\\_discovery\\_compass.c](#)
- NumberMap : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- QSPIHandle : [stm32l476g\\_discovery\\_qspi.c](#)
- SpiHandle : [stm32l476g\\_discovery.c](#)
- SpixTimeout : [stm32l476g\\_discovery.c](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1



# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files		
Directories								
File List		Globals						
All	Functions		Variables		Typedefs		Enumerations	Enumerator
Defines								

- Audio\_CallbackTypeDef : [stm32l476g\\_discovery\\_audio.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules	Data Structures	Files	
Directories					
File List	Globals				
All	Functions	Variables	Typedefs	Enumerations	Enumerator
Defines					

- ACCELERO\_StatusTypeDef : [stm32l476g\\_discovery\\_accelerometer.h](#)
- BarId\_Typedef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BatteryLevel\_Typedef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- COMPASS\_StatusTypeDef : [stm32l476g\\_discovery\\_compass.h](#)
- DigitPosition\_Typedef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- DoublePoint\_Typedef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- GYRO\_StatusTypeDef : [stm32l476g\\_discovery\\_gyroscope.h](#)
- IDD\_StatusTypeDef : [stm32l476g\\_discovery\\_idd.h](#)
- JOYMode\_TypeDef : [stm32l476g\\_discovery.h](#)
- JOYState\_TypeDef : [stm32l476g\\_discovery.h](#)
- Led\_TypeDef : [stm32l476g\\_discovery.h](#)
- Point\_Typedef : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SupplyMode\_TypeDef : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files			
Directories												
File List			Globals									
All	Functions		Variables		Typedefs		Enumerations		Enumerator			
Defines												
a	b	c	d	g	i	j	l	p	s			

- a -

- ACCELERO\_ERROR : [stm32l476g\\_discovery\\_accelerometer.h](#)
- ACCELERO\_OK : [stm32l476g\\_discovery\\_accelerometer.h](#)
- ACCELERO\_TIMEOUT : [stm32l476g\\_discovery\\_accelerometer.h](#)

- b -

- BATTERYLEVEL\_1\_2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERYLEVEL\_1\_4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERYLEVEL\_3\_4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERYLEVEL\_FULL : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- BATTERYLEVEL\_OFF : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- c -

- COMPASS\_ERROR : [stm32l476g\\_discovery\\_compass.h](#)
- COMPASS\_OK : [stm32l476g\\_discovery\\_compass.h](#)
- COMPASS\_TIMEOUT : [stm32l476g\\_discovery\\_compass.h](#)

- d -

- DOUBLEPOINT\_OFF : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- DOUBLEPOINT\_ON : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- g -

- GYRO\_ERROR : [stm32l476g\\_discovery\\_gyroscope.h](#)
- GYRO\_OK : [stm32l476g\\_discovery\\_gyroscope.h](#)
- GYRO\_TIMEOUT : [stm32l476g\\_discovery\\_gyroscope.h](#)

- i -

- IDD\_ERROR : [stm32l476g\\_discovery\\_idd.h](#)
- IDD\_OK : [stm32l476g\\_discovery\\_idd.h](#)
- IDD\_TIMEOUT : [stm32l476g\\_discovery\\_idd.h](#)
- IDD\_ZERO\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)

- j -

- JOY\_DOWN : [stm32l476g\\_discovery.h](#)
- JOY\_LEFT : [stm32l476g\\_discovery.h](#)
- JOY\_MODE\_EXTI : [stm32l476g\\_discovery.h](#)
- JOY\_MODE\_GPIO : [stm32l476g\\_discovery.h](#)
- JOY\_NONE : [stm32l476g\\_discovery.h](#)
- JOY\_RIGHT : [stm32l476g\\_discovery.h](#)
- JOY\_SEL : [stm32l476g\\_discovery.h](#)
- JOY\_UP : [stm32l476g\\_discovery.h](#)

- l -

- LCD\_BAR\_0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR\_NONE : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_MAX\_NUMBER :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- LCD\_DIGIT\_POSITION\_3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_5 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT\_POSITION\_6 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LED4 : [stm32l476g\\_discovery.h](#)
- LED5 : [stm32l476g\\_discovery.h](#)
- LED\_GREEN : [stm32l476g\\_discovery.h](#)
- LED\_RED : [stm32l476g\\_discovery.h](#)

- p -

- POINT\_OFF : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- POINT\_ON : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- s -

- SUPPLY\_MODE\_BATTERY : [stm32l476g\\_discovery.h](#)
- SUPPLY\_MODE\_ERROR : [stm32l476g\\_discovery.h](#)
- SUPPLY\_MODE\_EXTERNAL : [stm32l476g\\_discovery.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

- \_ -

- \_\_SPI\_DIRECTION\_1LINE\_RX : [stm32l476g\\_discovery.c](#)
- \_\_SPI\_DIRECTION\_1LINE\_TX : [stm32l476g\\_discovery.c](#)
- \_\_SPI\_DIRECTION\_2LINES : [stm32l476g\\_discovery.c](#)
- \_\_SPI\_DIRECTION\_2LINES\_RXONLY : [stm32l476g\\_discovery.c](#)
- \_\_STM32L476G\_DISCOVERY\_BSP\_VERSION : [stm32l476g\\_discovery.c](#)
- \_\_STM32L476G\_DISCOVERY\_BSP\_VERSION\_MAIN : [stm32l476g\\_discovery.c](#)
- \_\_STM32L476G\_DISCOVERY\_BSP\_VERSION\_RC : [stm32l476g\\_discovery.c](#)
- \_\_STM32L476G\_DISCOVERY\_BSP\_VERSION\_SUB1 : [stm32l476g\\_discovery.c](#)
- \_\_STM32L476G\_DISCOVERY\_BSP\_VERSION\_SUB2 : [stm32l476g\\_discovery.c](#)

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

- a -

- ACCELERO\_CS\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_HIGH : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_LOW : [stm32l476g\\_discovery.h](#)
- ACCELERO\_CS\_PIN : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- ACCELERO\_XLINT\_PIN : [stm32l476g\\_discovery.h](#)
- ASCII\_CHAR\_0 : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- ASCII\_CHAR\_APOSTROPHE : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- ASCII\_CHAR\_AT\_SYMBOL : [stm32l476g\\_discovery\\_glass\\_lcd.c](#)

- ASCII\_CHAR\_LEFT\_OPEN\_BRACE :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- ASCII\_CHAR\_LEFT\_OPEN\_BRACKET :  
[stm32l476g\\_discovery\\_glass\\_lcd.c](#)
- AUDIO\_DFSDM\_DMAX\_LEFT\_IRQHandler :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_AF :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_CLK\_DISABLE :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_CLK\_ENABLE :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_PORT :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CKOUT\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CLK\_DISABLE :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_CLK\_ENABLE :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAX\_CLK\_DISABLE :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAX\_CLK\_ENABLE :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAX\_LEFT\_CHANNEL :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAX\_LEFT\_IRQ :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAX\_MEM\_DATA\_SIZE :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMAX\_PERIPH\_DATA\_SIZE :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_DMIC\_DATIN\_PIN :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_LEFT\_CHANNEL :  
[stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_DFSDMx\_LEFT\_FILTER :  
[stm32l476g\\_discovery\\_audio.h](#)



- AUDIO\_ERROR : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_I2C\_ADDRESS : [stm32l476g\\_discovery.h](#)
- AUDIO\_IN\_IRQ\_PREPRIO : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_OK : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_OUT\_IRQ\_PREPRIO : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_RESET\_GPIO : [stm32l476g\\_discovery.h](#)
- AUDIO\_RESET\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- AUDIO\_RESET\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- AUDIO\_RESET\_PIN : [stm32l476g\\_discovery.h](#)
- AUDIO\_SAIx : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_CLK\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_CLK\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_CHANNEL : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_CLK\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_CLK\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_IRQ : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_IRQHandler : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_MEM\_DATA\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_DMAX\_PERIPH\_DATA\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_FS\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_AF : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_ENABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_GPIO\_PORT : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_PLL\_DISABLE : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_SAIx\_SCK\_PIN : [stm32l476g\\_discovery\\_audio.h](#)

- AUDIO\_SAIx\_SD\_PIN : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIO\_TIMEOUT : [stm32l476g\\_discovery\\_audio.h](#)
- AUDIODATA\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

- b -

- BATTERY\_DETECTION\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- BATTERY\_DETECTION\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- BATTERY\_DETECTION\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- BATTERY\_DETECTION\_PIN : [stm32l476g\\_discovery.h](#)
- BSP\_AUDIO\_FREQUENCY\_11K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_16K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_22K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_32K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_44K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_48K : [stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_FREQUENCY\_8K : [stm32l476g\\_discovery\\_audio.h](#)

- BSP\_AUDIO\_FREQUENCY\_96K :  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_CIRCULARMODE :  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_MONOMODE :  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_NORMALMODE :  
[stm32l476g\\_discovery\\_audio.h](#)
- BSP\_AUDIO\_OUT\_STEREO MODE :  
[stm32l476g\\_discovery\\_audio.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

- C -

- C\_CLOSEPARMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_DMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_FULL : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_MINUS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_MMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_NMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_OPENPARMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_PERCENT\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_PERCENT\_2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_PLUS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_SLATCH : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_STAR : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- C\_UMAP : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- CODEC\_AUDIO\_POWER\_OFF : [stm32l476g\\_discovery.h](#)
- CODEC\_AUDIO\_POWER\_ON : [stm32l476g\\_discovery.h](#)
- CODEC\_RESET\_DELAY : [stm32l476g\\_discovery\\_audio.h](#)
- COM\_PER\_DIGIT\_NB : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures			Files								
Directories															
File List		Globals													
All	Functions		Variables		Typedefs		Enumerations		Enumerator						
Defines															
_	a	b	c	d	g	i	j	l	m	q	r	s	u		

- d -

- DEFAULT\_AUDIO\_IN\_BIT\_RESOLUTION : [stm32l476g\\_discovery\\_audio.h](#)
- DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR : [stm32l476g\\_discovery\\_audio.h](#)
- DEFAULT\_AUDIO\_IN\_FREQ : [stm32l476g\\_discovery\\_audio.h](#)
- DEFAULT\_AUDIO\_IN\_VOLUME : [stm32l476g\\_discovery\\_audio.h](#)
- DFSDMClockDivider : [stm32l476g\\_discovery\\_audio.c](#)
- DFSDMFilterOrder : [stm32l476g\\_discovery\\_audio.c](#)
- DFSDMOverSampling : [stm32l476g\\_discovery\\_audio.c](#)
- DFSDMRightBitShift : [stm32l476g\\_discovery\\_audio.c](#)
- DISCOVERY\_I2C1 : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_ER\_IRQHandler : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_ER\_IRQn : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_EV\_IRQHandler : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_EV\_IRQn : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_FORCE\_RESET : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_RELEASE\_RESET :

## [stm32l476g\\_discovery.h](#)

- DISCOVERY\_I2C1\_SCL\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SCL\_SDA\_AF : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SDA\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SDA\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SDA\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_SDA\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C1\_TIMEOUT\_MAX : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2 : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_ER\_IRQn : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_EV\_IRQn : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_FORCE\_RESET : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_RELEASE\_RESET : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SCL\_SDA\_AF : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SDA\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SDA\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)



- DISCOVERY\_I2C2\_SDA\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_SDA\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_I2C2\_TIMEOUT\_MAX : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_IDD\_AMPLI\_GAIN : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT0\_STABDELAY : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT0\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT1\_STABDELAY : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT1\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT2\_STABDELAY : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT2\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT4\_STABDELAY : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_SHUNT4\_VALUE : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_IDD\_VDD\_MIN : [stm32l476g\\_discovery\\_idd.h](#)
- DISCOVERY\_SPIx : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_AF : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_CLOCK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_CLOCK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_FORCE\_RESET : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_GPIO\_RELEASE\_RESET : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_MISO\_PIN : [stm32l476g\\_discovery.h](#)
- DISCOVERY\_SPIx\_MOSI\_PIN : [stm32l476g\\_discovery.h](#)

- DISCOVERY\_SPIx\_SCK\_PIN : [stm32l476g\\_discovery.h](#)
- DMA\_MAX : [stm32l476g\\_discovery\\_audio.h](#)
- DMA\_MAX\_SIZE : [stm32l476g\\_discovery\\_audio.h](#)
- DOT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- DOUBLE\_DOT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- DOWN\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- DOWN\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- DOWN\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- DOWN\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- DOWN\_JOY\_PIN : [stm32l476g\\_discovery.h](#)
- DUMMY\_BYTE : [stm32l476g\\_discovery.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

- g -

- GYRO\_CS\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_HIGH : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_LOW : [stm32l476g\\_discovery.h](#)
- GYRO\_CS\_PIN : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- GYRO\_INT1\_PIN : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- GYRO\_INT2\_PIN : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

- i -

- IDD\_I2C\_ADDRESS : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- IDD\_INT\_PIN : [stm32l476g\\_discovery.h](#)
- IDD\_WAKEUP\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- IDD\_WAKEUP\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- IDD\_WAKEUP\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- IDD\_WAKEUP\_PIN : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

- j -

- JOY\_ALL\_PINS : [stm32l476g\\_discovery.h](#)
- JOYn : [stm32l476g\\_discovery.h](#)
- JOYx\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- JOYx\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

- | -

- LCD\_BAR0\_2\_COM : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR0\_2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR0\_SEG : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR1\_3\_COM : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR1\_3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR1\_SEG : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR2\_SEG : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_BAR3\_SEG : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM0\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM1\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM2\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_COM3\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM0\_SEG\_MASK :

- [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT1\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT1\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT1\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT1\_COM2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT1\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT1\_COM3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT2\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT2\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT2\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT2\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT2\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT2\_COM2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT2\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT2\_COM3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT3\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT3\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT3\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT3\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT3\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT3\_COM2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT3\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT3\_COM3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT4\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT4\_COM0\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_DIGIT4\_COM0\_1\_SEG\_MASK :

- [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM1\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM1\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM2\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM2\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM2\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM3\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM3\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT4\_COM3\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM0\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM0\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM0\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM1\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM1\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM2\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM2\_1\_SEG\_MASK :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)



- LCD\_DIGIT5\_COM2\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM3\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM3\_1\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT5\_COM3\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM0\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM1\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM2\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_DIGIT6\_COM3\_SEG\_MASK : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_GPIO\_BANKA\_PINS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_GPIO\_BANKB\_PINS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_GPIO\_BANKC\_PINS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_GPIO\_BANKD\_PINS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_1ST\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_1ST\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_2ND\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_2ND\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_3RD\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_3RD\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_4TH\_POS :

- [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM0\_SEG\_4TH\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM1\_SEG\_1ST\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM1\_SEG\_1ST\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM1\_SEG\_2ND\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM1\_SEG\_2ND\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM1\_SEG\_3RD\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM1\_SEG\_3RD\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM1\_SEG\_4TH\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM1\_SEG\_4TH\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM2\_SEG\_1ST\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM2\_SEG\_1ST\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM2\_SEG\_2ND\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM2\_SEG\_2ND\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM2\_SEG\_3RD\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM2\_SEG\_3RD\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM2\_SEG\_4TH\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM2\_SEG\_4TH\_SHIFT :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
  - LCD\_MAP\_CHAR\_COM3\_SEG\_1ST\_POS :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- LCD\_MAP\_CHAR\_COM3\_SEG\_1ST\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_2ND\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_2ND\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_3RD\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_3RD\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_4TH\_POS : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_MAP\_CHAR\_COM3\_SEG\_4TH\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG0\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG10 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG10\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG11 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG11\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG12 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG12\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG13 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG13\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG14 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG14\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG15 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG15\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG16 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG16\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG17 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG17\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG18 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG18\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG19 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG19\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- LCD\_SEG1\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG20 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG20\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG21 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG21\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG22 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG22\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG23 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG23\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG2\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG3\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG4\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG5 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG5\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG6 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG6\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG7 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG7\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG8 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG8\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG9 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LCD\_SEG9\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- LED4\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- LED4\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- LED4\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- LED4\_PIN : [stm32l476g\\_discovery.h](#)
- LED5\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- LED5\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- LED5\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- LED5\_PIN : [stm32l476g\\_discovery.h](#)
- LEDn : [stm32l476g\\_discovery.h](#)
- LEDx\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- LEDx\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)

- LEFT\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- LEFT\_JOY\_PIN : [stm32l476g\\_discovery.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files								
Directories														
File List		Globals												
All	Functions	Variables	Typedefs	Enumerations	Enumerator									
Defines														
—	a	b	c	d	g	i	j	l	m	q	r	s	u	

## - m -

- MAGNETO\_CS\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_HIGH : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_LOW : [stm32l476g\\_discovery.h](#)
- MAGNETO\_CS\_PIN : [stm32l476g\\_discovery.h](#)
- MAGNETO\_DRDY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_DRDY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_DRDY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- MAGNETO\_DRDY\_PIN : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT1\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT1\_PIN : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- MAGNETO\_INT\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- MCU\_LCD\_COM0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM0\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)

- MCU\_LCD\_COM1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM1\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM2\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM3\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM4\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM5 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM5\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM6 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM6\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM7 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_COM7\_1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG0 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG0\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG1 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG10 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG10\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG11 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG11\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG12 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG12\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG13 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG13\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG14 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG14\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG15 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG15\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG16 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG16\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG17 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG17\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG18 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG18\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG19 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG19\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)



- MCU\_LCD\_SEG1\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG2 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG20 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG20\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG21 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG21\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG22 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG22\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG23 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG23\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG24 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG24\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG25 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG25\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG26 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG26\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG27 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG27\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG28 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG28\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG29 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG29\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG2\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG3 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG30 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG30\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG31 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG31\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG32 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG32\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG33 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG33\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG34 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG34\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG35 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG35\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG36 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)



- MCU\_LCD\_SEG36\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG37 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG37\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG38 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG38\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG39\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG3\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG4 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG40\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG41\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG42\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG43\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG4\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG5 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG5\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG6 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG6\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG7 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG7\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG8 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG8\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG9 : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MCU\_LCD\_SEG9\_SHIFT : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- MULTIPLEBYTE\_CMD : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files							
Directories													
File List		Globals											
All	Functions	Variables		Typedefs	Enumerations		Enumerator						
Defines													
_	a	b	c	d	g	i	j	l	m	q	r	s	u

- q -

- QSPI\_BUSY : [stm32l476g\\_discovery\\_qspi.h](#)
- QSPI\_ERROR : [stm32l476g\\_discovery\\_qspi.h](#)
- QSPI\_NOT\_SUPPORTED : [stm32l476g\\_discovery\\_qspi.h](#)
- QSPI\_OK : [stm32l476g\\_discovery\\_qspi.h](#)
- QSPI\_SUSPENDED : [stm32l476g\\_discovery\\_qspi.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files									
Directories															
File List		Globals													
All	Functions	Variables		Typedefs	Enumerations		Enumerator								
Defines															
_	a	b	c	d	g	i	j	l	m	q	r	s	u		

- r -

- READWRITE\_CMD : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- RIGHT\_JOY\_PIN : [stm32l476g\\_discovery.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

Main Page			Modules			Data Structures			Files		
Directories											
File List			Globals								
All		Functions		Variables		Typedefs		Enumerations		Enumerator	
Defines											
<div><div></div><div>a</div><div>b</div><div>c</div><div>d</div><div>g</div><div>i</div><div>j</div><div>l</div><div>m</div><div>q</div><div>r</div><div>s</div><div>u</div></div>											

- S -

- SAIClockDivider : [stm32l476g\\_discovery\\_audio.c](#)
- SaturateLH : [stm32l476g\\_discovery\\_audio.c](#)
- SCROLL\_SPEED\_HIGH : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SCROLL\_SPEED\_LOW : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SCROLL\_SPEED\_MEDIUM :  
[stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SEG\_PER\_DIGIT\_NB : [stm32l476g\\_discovery\\_glass\\_lcd.h](#)
- SEL\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- SEL\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- SEL\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- SEL\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- SEL\_JOY\_PIN : [stm32l476g\\_discovery.h](#)
- SPIx\_TIMEOUT\_MAX : [stm32l476g\\_discovery.h](#)

# STM32L476G-Discovery BSP User Manual

Main Page		Modules		Data Structures		Files							
Directories													
File List		Globals											
All	Functions	Variables	Typedefs	Enumerations	Enumerator								
Defines													
—	a	b	c	d	g	i	j	l	m	q	r	s	u

- u -

- UP\_JOY\_EXTI\_IRQn : [stm32l476g\\_discovery.h](#)
- UP\_JOY\_GPIO\_CLK\_DISABLE : [stm32l476g\\_discovery.h](#)
- UP\_JOY\_GPIO\_CLK\_ENABLE : [stm32l476g\\_discovery.h](#)
- UP\_JOY\_GPIO\_PORT : [stm32l476g\\_discovery.h](#)
- UP\_JOY\_PIN : [stm32l476g\\_discovery.h](#)

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
<a href="#">Defines</a>   <a href="#">Functions</a>   <a href="#">Variables</a>				

## stm32l476g\_discovery.c File Reference

This file provides a set of firmware functions to manage Leds, push-button and joystick of STM32L476G-Discovery board (MB1184) [More...](#)

```
#include "stm32l476g_discovery.h"
```

[Go to the source code of this file.](#)

## Defines

#define	<b>__STM32L476G_DISCOVERY_BSP_VERSION_MAIN</b>	(0x00)
	STM32L476G DISCOVERY BSP Driver version number \$VERSION\$.	
#define	<b>__STM32L476G_DISCOVERY_BSP_VERSION_SUB1</b>	(0x00)
#define	<b>__STM32L476G_DISCOVERY_BSP_VERSION_SUB2</b>	(0x00)
#define	<b>__STM32L476G_DISCOVERY_BSP_VERSION_RC</b>	(0x01)
#define	<b>__STM32L476G_DISCOVERY_BSP_VERSION</b>	
#define	<b>__SPI_DIRECTION_2LINES(__HANDLE__)</b>	
#define	<b>__SPI_DIRECTION_2LINES_RXONLY(__HANDLE__)</b>	
#define	<b>__SPI_DIRECTION_1LINE_TX(__HANDLE__)</b>	
#define	<b>__SPI_DIRECTION_1LINE_RX(__HANDLE__)</b>	

## Functions

static void	<b>I2C2_Init</b> (void) Discovery I2C2 Bus initialization.
static void	<b>I2C2_MspInit</b> (I2C_HandleTypeDef *hi2c) Discovery I2C2 MSP Initialization.
static void	<b>I2C2_DeInit</b> (void) Discovery I2C2 Bus Deinitialization.
static void	<b>I2C2_MspDeInit</b> (I2C_HandleTypeDef *hi2c) Discovery I2C2 MSP DeInitialization.
static void	<b>I2C2_WriteData</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t Value) Write a value in a register of the device through BUS.
static HAL_StatusTypeDef	<b>I2C2_WriteBuffer</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length) Write a value in a register of the device through BUS.
static uint8_t	<b>I2C2_ReadData</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize) Read a register of the device through BUS.
static HAL_StatusTypeDef	<b>I2C2_ReadBuffer</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length) Reads multiple data on the BUS.
static void	<b>I2C2_Error</b> (void) Discovery I2C2 error treatment function.
static void	<b>I2C1_Init</b> (void) Discovery I2C1 Bus initialization.
static void	<b>I2C1_MspInit</b> (I2C_HandleTypeDef *hi2c)



	Discovery I2C1 MSP Initialization.
static void	<b>I2C1_DeInit</b> (void) Discovery I2C1 Bus Deinitialization.
static void	<b>I2C1_MspDeInit</b> (I2C_HandleTypeDef *hi2c) Discovery I2C1 MSP Deinitialization.
static HAL_StatusTypeDef	<b>I2C1_WriteBuffer</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length) Write a value in a register of the device through BUS.
static HAL_StatusTypeDef	<b>I2C1_ReadBuffer</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length) Reads multiple data on the BUS.
static void	<b>I2C1_Error</b> (void) Discovery I2C1 error treatment function.
static void	<b>SPIx_Init</b> (void) SPIx Bus initialization.
static void	<b>SPIx_MspInit</b> (SPI_HandleTypeDef *hspi) SPI MSP Init.
static void	<b>SPIx_DeInit</b> (void) SPIx Bus Deinitialization.
static void	<b>SPIx_MspDeInit</b> (void) SPI MSP DeInit.
static uint8_t	<b>SPIx_WriteRead</b> (uint8_t Byte) Sends a Byte through the SPI interface and return the Byte received from the SPI bus.
static void	<b>SPIx_Write</b> (uint8_t Byte) Sends a Byte through the SPI interface.
static uint8_t	<b>SPIx_Read</b> (void) Receives a Byte from the SPI bus.
void	<b>EEPROM_I2C_IO_Init</b> (void)

HAL_StatusTypeDef	<b>EEPROM_I2C_IO_WriteData</b> (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize)
HAL_StatusTypeDef	<b>EEPROM_I2C_IO_ReadData</b> (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize)
HAL_StatusTypeDef	<b>EEPROM_I2C_IO_IsDeviceReady</b> (uint16_t DevAddress, uint32_t Trials)
void	<b>AUDIO_IO_Init</b> (void) Initializes Audio low level.
void	<b>AUDIO_IO_DeInit</b> (void) Deinitializes Audio low level.
void	<b>AUDIO_IO_Write</b> (uint8_t Addr, uint8_t Reg, uint8_t Value) Writes a single data.
uint8_t	<b>AUDIO_IO_Read</b> (uint8_t Addr, uint8_t Reg) Reads a single data.
void	<b>AUDIO_IO_Delay</b> (uint32_t Delay) AUDIO Codec delay.
void	<b>ACCELERO_IO_Init</b> (void) Configures COMPASS/ACCELEROMETER io interface.
void	<b>ACCELERO_IO_DeInit</b> (void) De-Configures COMPASS/ACCELEROMETER io interface.
void	<b>ACCELERO_IO_ITConfig</b> (void) Configures COMPASS / ACCELERO click IT.
void	<b>ACCELERO_IO_Write</b> (uint8_t RegisterAddr, uint8_t Value) Writes one byte to the COMPASS / ACCELEROMETER.

uint8_t	<b>ACCELERO_IO_Read</b> (uint8_t RegisterAddr) Reads a block of data from the COMPASS / ACCELEROMETER.
void	<b>MAGNETO_IO_Init</b> (void) Configures COMPASS/MAGNETO SPI interface.
void	<b>MAGNETO_IO_DeInit</b> (void) de-Configures COMPASS/MAGNETO SPI interface.
void	<b>MAGNETO_IO_ITConfig</b> (void)
void	<b>MAGNETO_IO_Write</b> (uint8_t RegisterAddr, uint8_t Value) Writes one byte to the COMPASS/MAGNETO.
uint8_t	<b>MAGNETO_IO_Read</b> (uint8_t RegisterAddr) Reads a block of data from the COMPASS/MAGNETO.
void	<b>GYRO_IO_Init</b> (void) Configures the GYRO SPI interface.
void	<b>GYRO_IO_DeInit</b> (void) de-Configures GYRO SPI interface.
void	<b>GYRO_IO_Write</b> (uint8_t *pBuffer, uint8_t WriteAddr, uint16_t NumByteToWrite) Writes one byte to the GYRO.
void	<b>GYRO_IO_Read</b> (uint8_t *pBuffer, uint8_t ReadAddr, uint16_t NumByteToRead) Reads a block of data from the GYROSCOPE.
void	<b>IOE_Init</b> (void)
void	<b>IOE_ITConfig</b> (void)
void	<b>IOE_Delay</b> (uint32_t Delay)
void	<b>IOE_Write</b> (uint8_t Addr, uint8_t Reg,

	uint8_t Value)
uint8_t	<b>IOE_Read</b> (uint8_t Addr, uint8_t Reg)
uint16_t	<b>IOE_ReadMultiple</b> (uint8_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length)
void	<b>MFX_IO_Init</b> (void) Initializes MFX low level.
void	<b>MFX_IO_DeInit</b> (void) Deinitializes MFX low level.
void	<b>MFX_IO_ITConfig</b> (void) Configures MFX low level interrupt.
void	<b>MFX_IO_EnableWakeupPin</b> (void) Configures MFX wke up pin.
void	<b>MFX_IO_Wakeup</b> (void) Wakeup MFX.
void	<b>MFX_IO_Delay</b> (uint32_t Delay) MFX delay.
void	<b>MFX_IO_Write</b> (uint16_t Addr, uint8_t Reg, uint8_t Value) MFX writes single data.
uint8_t	<b>MFX_IO_Read</b> (uint16_t Addr, uint8_t Reg) MFX reads single data.
void	<b>MFX_IO_WriteMultiple</b> (uint16_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length) MFX writes multiple data.
uint16_t	<b>MFX_IO_ReadMultiple</b> (uint16_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length) MFX reads multiple data.
uint32_t	<b>BSP_GetVersion</b> (void) This method returns the STM32L476 DISCOVERY BSP Driver revision.

<b>SupplyMode_TypeDef</b>	<b>BSP_SupplyModeDetection</b> (void) This method returns the STM32L476 DISCOVERY supply mode.
void	<b>BSP_LED_Init</b> (Led_TypeDef Led) Configures LED GPIOs.
void	<b>BSP_LED_DeInit</b> (Led_TypeDef Led) Unconfigures LED GPIOs.
void	<b>BSP_LED_On</b> (Led_TypeDef Led) Turns selected LED On.
void	<b>BSP_LED_Off</b> (Led_TypeDef Led) Turns selected LED Off.
void	<b>BSP_LED_Toggle</b> (Led_TypeDef Led) Toggles the selected LED.
uint8_t	<b>BSP_JOY_Init</b> (JOYMode_TypeDef Joy_Mode) Configures all buttons of the joystick in GPIO or EXTI modes.
void	<b>BSP_JOY_DeInit</b> (void) Unconfigures all GPIOs used as buttons of the joystick.
<b>JOYState_TypeDef</b>	<b>BSP_JOY_GetState</b> (void) Returns the current joystick status.

## Variables

GPIO_TypeDef *	<b>LED_PORT [LEDn]</b> LED variables.
const uint16_t	<b>LED_PIN [LEDn]</b>
GPIO_TypeDef *	<b>JOY_PORT [JOYn]</b> JOYSTICK variables.
const uint16_t	<b>JOY_PIN [JOYn]</b>
const uint8_t	<b>JOY_IRQn [JOYn]</b>
uint32_t	<b>I2c1Timeout =</b> <b>DISCOVERY_I2C2_TIMEOUT_MAX</b> BUS variables.
uint32_t	<b>I2c2Timeout =</b> <b>DISCOVERY_I2C2_TIMEOUT_MAX</b>
static I2C_HandleTypeDef	<b>I2c1Handle</b>
static I2C_HandleTypeDef	<b>I2c2Handle</b>
uint32_t	<b>SpixTimeout = SPIx_TIMEOUT_MAX</b>
static SPI_HandleTypeDef	<b>SpiHandle</b>

## Detailed Description

This file provides a set of firmware functions to manage Leds, push-button and joystick of STM32L476G-Discovery board (MB1184)

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery.c](#).



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
<div>Defines   Enumerations   Functions</div>				

## stm32l476g\_discovery.h File Reference

This file contains definitions for STM32L476G\_DISCOVERY's LEDs, push-buttons hardware resources (MB1184). [More...](#)

```
#include "stm32l4xx_hal.h"
```

[Go to the source code of this file.](#)

## Defines

#define	<b>BATTERY_DETECTION_PIN</b>	GPIO_PIN_3
#define	<b>BATTERY_DETECTION_GPIO_PORT</b>	GPIOB
#define	<b>BATTERY_DETECTION_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOB_CLK_ENABLE()
#define	<b>BATTERY_DETECTION_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOB_CLK_DISABLE()
#define	<b>LEDn</b>	2
#define	<b>LED4_PIN</b>	GPIO_PIN_2
#define	<b>LED4_GPIO_PORT</b>	GPIOB
#define	<b>LED4_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOB_CLK_ENABLE()
#define	<b>LED4_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOB_CLK_DISABLE()
#define	<b>LED5_PIN</b>	GPIO_PIN_8
#define	<b>LED5_GPIO_PORT</b>	GPIOE
#define	<b>LED5_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOE_CLK_ENABLE()
#define	<b>LED5_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOE_CLK_DISABLE()
#define	<b>LEDx_GPIO_CLK_ENABLE(__LED__)</b>	
#define	<b>LEDx_GPIO_CLK_DISABLE(__LED__)</b>	
#define	<b>JOYn</b>	5
#define	<b>RIGHT_JOY_PIN</b>	GPIO_PIN_2 /* PA.02 */ Joystick Right push-button.
#define	<b>RIGHT_JOY_GPIO_PORT</b>	GPIOA
#define	<b>RIGHT_JOY_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOA_CLK_ENABLE()
#define	<b>RIGHT_JOY_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOA_CLK_DISABLE()
#define	<b>RIGHT_JOY_EXTI_IRQn</b>	EXTI2_IRQn
#define	<b>LEFT_JOY_PIN</b>	GPIO_PIN_1 /* PA.01 */ Joystick Left push-button.
#define	<b>LEFT_JOY_GPIO_PORT</b>	GPIOA
#define	<b>LEFT_JOY_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOA_CLK_ENABLE()
#define	<b>LEFT_JOY_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOA_CLK_DISABLE()
#define	<b>LEFT_JOY_EXTI_IRQn</b>	EXTI1_IRQn
#define	<b>UP_JOY_PIN</b>	GPIO_PIN_3 /* PA.03 */ Joystick Up push-button.
#define	<b>UP_JOY_GPIO_PORT</b>	GPIOA

```

#define UP_JOY_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK
#define UP_JOY_GPIO_CLK_DISABLE() __HAL_RCC_GPIOA_CLK
#define UP_JOY_EXTI_IRQn EXTI3_IRQn
#define DOWN_JOY_PIN GPIO_PIN_5 /* PA.05 */
    Joystick Down push-button.

#define DOWN_JOY_GPIO_PORT GPIOA
#define DOWN_JOY_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_
#define DOWN_JOY_GPIO_CLK_DISABLE() __HAL_RCC_GPIOA_
#define DOWN_JOY_EXTI_IRQn EXTI9_5_IRQn
#define SEL_JOY_PIN GPIO_PIN_0 /* PA.00 */
    Joystick Sel push-button.

#define SEL_JOY_GPIO_PORT GPIOA
#define SEL_JOY_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CL
#define SEL_JOY_GPIO_CLK_DISABLE() __HAL_RCC_GPIOA_CL
#define SEL_JOY_EXTI_IRQn EXTI0_IRQn
#define JOYx_GPIO_CLK_ENABLE(__JOY__)
#define JOYx_GPIO_CLK_DISABLE(__JOY__)
#define JOY_ALL_PINS (RIGHT_JOY_PIN | LEFT_JOY_PIN | UP_J
    | SEL_JOY_PIN)

#define DISCOVERY_SPIx SPI2
#define DISCOVERY_SPIx_CLOCK_ENABLE() __HAL_RCC_SPI2_
#define DISCOVERY_SPIx_CLOCK_DISABLE() __HAL_RCC_SPI2_
#define DISCOVERY_SPIx_GPIO_PORT GPIOD /* GPIOD */
#define DISCOVERY_SPIx_AF GPIO_AF5_SPI2
#define DISCOVERY_SPIx_GPIO_CLK_ENABLE() __HAL_RCC_GI
#define DISCOVERY_SPIx_GPIO_CLK_DISABLE() __HAL_RCC_G
#define DISCOVERY_SPIx_GPIO_FORCE_RESET() __HAL_RCC_S
#define DISCOVERY_SPIx_GPIO_RELEASE_RESET() __HAL_RCC
#define DISCOVERY_SPIx_SCK_PIN GPIO_PIN_1 /* PD.01 */
#define DISCOVERY_SPIx_MISO_PIN GPIO_PIN_3 /* PD.03 */
#define DISCOVERY_SPIx_MOSI_PIN GPIO_PIN_4 /* PD.04 */
#define SPIx_TIMEOUT_MAX ((uint32_t)0x1000)
#define READWRITE_CMD ((uint8_t)0x80)
#define MULTIPLEBYTE_CMD ((uint8_t)0x40)

```

```

#define DUMMY_BYTE ((uint8_t)0x00)
#define DISCOVERY_I2C1_SCL_GPIO_PORT GPIOB
#define DISCOVERY_I2C1_SDA_GPIO_PORT GPIOB
#define DISCOVERY_I2C1_SCL_PIN GPIO_PIN_6
#define DISCOVERY_I2C1_SDA_PIN GPIO_PIN_7
#define DISCOVERY_I2C1_SCL_SDA_AF GPIO_AF4_I2C1
#define DISCOVERY_I2C1 I2C1
#define DISCOVERY_I2C1_CLK_ENABLE() __HAL_RCC_I2C1_CLK_ENABLE()
#define DISCOVERY_I2C1_CLK_DISABLE() __HAL_RCC_I2C1_CLK_DISABLE()
#define DISCOVERY_I2C1_SDA_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define DISCOVERY_I2C1_SCL_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define DISCOVERY_I2C1_SDA_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define DISCOVERY_I2C1_SCL_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define DISCOVERY_I2C1_FORCE_RESET() __HAL_RCC_I2C1_FORCE_RESET()
#define DISCOVERY_I2C1_RELEASE_RESET() __HAL_RCC_I2C1_RELEASE_RESET()
#define DISCOVERY_I2C1_EV_IRQn I2C1_EV_IRQn
#define DISCOVERY_I2C1_EV_IRQHandler I2C1_EV_IRQHandler
#define DISCOVERY_I2C1_ER_IRQn I2C1_ER_IRQn
#define DISCOVERY_I2C1_ER_IRQHandler I2C1_ER_IRQHandler
#define AUDIO_I2C_ADDRESS ((uint16_t) 0x94)
#define DISCOVERY_I2C1_TIMEOUT_MAX 3000
#define DISCOVERY_I2C2_SCL_PIN GPIO_PIN_10
#define DISCOVERY_I2C2_SCL_GPIO_PORT GPIOB
#define DISCOVERY_I2C2_SDA_PIN GPIO_PIN_11
#define DISCOVERY_I2C2_SDA_GPIO_PORT GPIOB
#define DISCOVERY_I2C2_SCL_SDA_AF GPIO_AF4_I2C2
#define DISCOVERY_I2C2 I2C2
#define DISCOVERY_I2C2_CLK_ENABLE() __HAL_RCC_I2C2_CLK_ENABLE()
#define DISCOVERY_I2C2_CLK_DISABLE() __HAL_RCC_I2C2_CLK_DISABLE()
#define DISCOVERY_I2C2_SDA_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define DISCOVERY_I2C2_SCL_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define DISCOVERY_I2C2_SDA_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define DISCOVERY_I2C2_SCL_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define DISCOVERY_I2C2_FORCE_RESET() __HAL_RCC_I2C2_FORCE_RESET()

```

```

#define DISCOVERY_I2C2_RELEASE_RESET() __HAL_RCC_I2C2_
#define DISCOVERY_I2C2_EV_IRQn I2C2_EV_IRQn
#define DISCOVERY_I2C2_ER_IRQn I2C2_ER_IRQn
#define IDD_I2C_ADDRESS ((uint16_t) 0x84)
#define DISCOVERY_I2C2_TIMEOUT_MAX 3000
#define ACCELERO_CS_LOW() HAL_GPIO_WritePin(ACCELERO_
ACCELERO_CS_PIN, GPIO_PIN_RESET)
Accelerometer Chip Select macro definition.

#define ACCELERO_CS_HIGH() HAL_GPIO_WritePin(ACCELERO_
ACCELERO_CS_PIN, GPIO_PIN_SET)

#define ACCELERO_CS_GPIO_PORT GPIOE /* GPIOE */
Accelerometer SPI Interface pins.

#define ACCELERO_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIO
#define ACCELERO_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPI
#define ACCELERO_CS_PIN GPIO_PIN_0 /* PE.00 */
#define ACCELERO_XLINT_GPIO_PORT GPIOE /* GPIOE */
Accelerometer Interrupt pins.

#define ACCELERO_XLINT_GPIO_CLK_ENABLE() __HAL_RCC_G
#define ACCELERO_XLINT_GPIO_CLK_DISABLE() __HAL_RCC_(
#define ACCELERO_XLINT_PIN GPIO_PIN_1 /* PE.01 */
#define ACCELERO_XLINT_EXTI_IRQn EXTI1_IRQn
#define MAGNETO_CS_LOW() HAL_GPIO_WritePin(MAGNETO_CS
MAGNETO_CS_PIN, GPIO_PIN_RESET)
Magnetometer Chip Select macro definition.

#define MAGNETO_CS_HIGH() HAL_GPIO_WritePin(MAGNETO_CS
MAGNETO_CS_PIN, GPIO_PIN_SET)

#define MAGNETO_CS_GPIO_PORT GPIOC /* GPIOC */
Magnetometer SPI Interface pins.

#define MAGNETO_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIO
#define MAGNETO_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPI
#define MAGNETO_CS_PIN GPIO_PIN_0 /* PC.00 */
#define MAGNETO_INT_GPIO_PORT GPIOC /* GPIOC */
Magnetometer Interrupt pins.

#define MAGNETO_INT_GPIO_CLK_ENABLE() __HAL_RCC_GPI

```

```

#define MAGNETO_INT_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC_CLK_DISABLE()
#define MAGNETO_INT1_PIN GPIO_PIN_1 /* PC.01 */
#define MAGNETO_INT1_EXTI_IRQn EXTI1_IRQn
#define MAGNETO_DRDY_GPIO_PORT GPIOC /* GPIOC */
#define MAGNETO_DRDY_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC_CLK_ENABLE()
#define MAGNETO_DRDY_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC_CLK_DISABLE()
#define MAGNETO_DRDY_PIN GPIO_PIN_2 /* PC.01 */
#define CODEC_AUDIO_POWER_OFF() HAL_GPIO_WritePin(AUDIO_RESET_GPIO_PORT, AUDIO_RESET_PIN, GPIO_PIN_RESET)
/*
 * Audio codec chip reset definition.
 */
#define CODEC_AUDIO_POWER_ON() HAL_GPIO_WritePin(AUDIO_RESET_GPIO_PORT, AUDIO_RESET_PIN, GPIO_PIN_SET)
#define AUDIO_RESET_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC_CLK_ENABLE()
#define AUDIO_RESET_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC_CLK_DISABLE()
#define AUDIO_RESET_PIN GPIO_PIN_3
#define AUDIO_RESET_GPIO GPIOC
#define GYRO_CS_LOW() HAL_GPIO_WritePin(GYRO_CS_GPIO_PORT, GYRO_CS_PIN, GPIO_PIN_RESET)
/*
 * Gyroscope Chip Select macro definition.
 */
#define GYRO_CS_HIGH() HAL_GPIO_WritePin(GYRO_CS_GPIO_PORT, GYRO_CS_PIN, GPIO_PIN_SET)
#define GYRO_CS_GPIO_PORT GPIOD /* GPIOD */
/*
 * Gyroscope SPI Interface pins.
 */
#define GYRO_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define GYRO_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPIOD_CLK_DISABLE()
#define GYRO_CS_PIN GPIO_PIN_7 /* PD.07 */
#define GYRO_INT1_GPIO_PORT GPIOD /* GPIOD */
/*
 * Gyroscope Interrupt pins.
 */
#define GYRO_INT1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define GYRO_INT1_GPIO_CLK_DISABLE() __HAL_RCC_GPIOD_CLK_DISABLE()
#define GYRO_INT1_PIN GPIO_PIN_2 /* PD.02 */
#define GYRO_INT1_EXTI_IRQn EXTI2_IRQn
#define GYRO_INT2_GPIO_PORT GPIOB /* GPIOB */
#define GYRO_INT2_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()

```

```

#define GYRO_INT2_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_
#define GYRO_INT2_PIN GPIO_PIN_8 /* PB.08 */
#define GYRO_INT2_EXTI_IRQn EXTI9_5_IRQn
#define IDD_INT_GPIO_PORT GPIOC /* GPIOC */
    Idd current measurement interface pins.
#define IDD_INT_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC_CLK
#define IDD_INT_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC_CLK
#define IDD_INT_PIN GPIO_PIN_13 /* PC.13 */
#define IDD_INT_EXTI_IRQn EXTI15_10_IRQn
#define IDD_WAKEUP_GPIO_PORT GPIOA /* GPIOA */
#define IDD_WAKEUP_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_
#define IDD_WAKEUP_GPIO_CLK_DISABLE() __HAL_RCC_GPIOA_
#define IDD_WAKEUP_PIN GPIO_PIN_4 /* PA.04 */

```

## Enumerations

```
enum Led_TypeDef { LED4 = 0, LED5 = 1, LED_RED = LED4,  
                  LED_GREEN = LED5 }
```

LED Types Definition. [More...](#)

```
enum JOYState_TypeDef {
    JOY_SEL = 0, JOY_LEFT = 1, JOY_RIGHT = 2,
    JOY_DOWN = 3,
    JOY_UP = 4, JOY_NONE = 5
}
```

JOYSTICK Types Definition. [More...](#)

```
enum JOYMode_TypeDef { JOY_MODE_GPIO = 0,  
                      JOY_MODE_EXTI = 1 }
```

```
enum SupplyMode_TypeDef { SUPPLY_MODE_ERROR = 0,  
    SUPPLY_MODE_EXTERNAL = 1,  
    SUPPLY_MODE_BATTERY = 2 }
```



## Functions

uint32_t	<b>BSP_GetVersion</b> (void) This method returns the STM32L476 DISCOVERY BSP Driver revision.
<b>SupplyMode_TypeDef</b>	<b>BSP_SupplyModeDetection</b> (void) This method returns the STM32L476 DISCOVERY supply mode.
void	<b>BSP_LED_Init</b> ( <b>Led_TypeDef</b> Led) Configures LED GPIOs.
void	<b>BSP_LED_DeInit</b> ( <b>Led_TypeDef</b> Led) Unconfigures LED GPIOs.
void	<b>BSP_LED_On</b> ( <b>Led_TypeDef</b> Led) Turns selected LED On.
void	<b>BSP_LED_Off</b> ( <b>Led_TypeDef</b> Led) Turns selected LED Off.
void	<b>BSP_LED_Toggle</b> ( <b>Led_TypeDef</b> Led) Toggles the selected LED.
uint8_t	<b>BSP_JOY_Init</b> ( <b>JOYMode_TypeDef</b> Joy_Mode) Configures all buttons of the joystick in GPIO or EXTI modes.
void	<b>BSP_JOY_DeInit</b> (void) Unconfigures all GPIOs used as buttons of the joystick.
<b>JOYState_TypeDef</b>	<b>BSP_JOY_GetState</b> (void) Returns the current joystick status.

## Detailed Description

This file contains definitions for STM32L476G\_DISCOVERY's LEDs, push-buttons hardware resources (MB1184).

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery.h](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
			<a href="#">Functions</a>   <a href="#">Variables</a>	

## stm32l476g\_discovery\_accelerometer.c File Reference

This file provides a set of functions needed to manage the ACCELEROMETER MEMS available on STM32L476G-Discovery Kit.  
[More...](#)

```
#include "stm32l476g_discovery_accelerometer.h"
```

[Go to the source code of this file.](#)

## Functions

uint8\_t **BSP\_ACCELERO\_Init** (void)

Initialize Accelerometer.

void **BSP\_ACCELERO\_Reset** (void)

Reboot memory content of Accelerometer.

void **BSP\_ACCELERO\_GetXYZ** (int16\_t \*pDataXYZ)

Get XYZ angular accelerations from the Accelerometer.

## Variables

---

```
static ACCELERO_DrvTypeDef * AccelerometerDrv
```

---

## Detailed Description

This file provides a set of functions needed to manage the ACCELEROMETER MEMS available on STM32L476G-Discovery Kit.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_accelerometer.c](#).



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
			<a href="#">Enumerations</a>	<a href="#">Functions</a>

## stm32l476g\_discovery\_accelerometer.h File Reference

This file contains definitions for [stm32l476g\\_discovery\\_accelerometer.c](#) firmware driver. [More...](#)

```
#include "stm32l476g_discovery.h" #include
"../Components/lsm303dlhc/lsm303dlhc.h"
```

[Go to the source code of this file.](#)

## Enumerations

```
enum ACCELERO_StatusTypeDef { ACCELERO_OK = 0,  
                               ACCELERO_ERROR = 1, ACCELERO_TIMEOUT = 2 }
```

## Functions

uint8\_t **BSP\_ACCELERO\_Init** (void)

Initialize Accelerometer.

void **BSP\_ACCELERO\_Reset** (void)

Reboot memory content of Accelerometer.

void **BSP\_ACCELERO\_GetXYZ** (int16\_t \*pDataXYZ)

Get XYZ angular accelerations from the Accelerometer.

## Detailed Description

This file contains definitions for [stm32l476g\\_discovery\\_accelerometer.c](#) firmware driver.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**| Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_accelerometer.h](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
<a href="#">Data Structures</a>   <a href="#">Defines</a>   <a href="#">Functions</a>   <a href="#">Variables</a>				

## stm32l476g\_discovery\_audio.c File Reference

This file provides a set of functions needed to manage the Audio driver for the STM32L476G-Discovery board. [More...](#)

```
#include <string.h> #include "stm32l476g_discovery_audio.h"
```

[Go to the source code of this file.](#)

## Data Structures

struct	AUDIO_OUT_TypeDef
--------	-------------------

struct	AUDIO_IN_TypeDef
--------	------------------

## Defines

#define	<b>SAIClockDivider</b> (__FREQUENCY__)
#define	<b>DFSDMOverSampling</b> (__FREQUENCY__)
#define	<b>DFSDMClockDivider</b> (__FREQUENCY__)
#define	<b>DFSDMFilterOrder</b> (__FREQUENCY__)
#define	<b>DFSDMRightBitShift</b> (__FREQUENCY__)
#define	<b>SaturaLH</b> (N, L, H) (((N)<(L))? (L):(((N)>(H))? (H):(N)))



## Functions

static void	<b>AUDIO_CODEC_Reset</b> (void)	Resets the audio codec.
static uint8_t	<b>AUDIO_SAIx_Init</b> (uint32_t AudioFreq)	Initializes the Audio Codec audio interface (SAI).
static uint8_t	<b>AUDIO_SAIx_DeInit</b> (void)	De-initializes the Audio Codec audio interface (SAI).
static uint8_t	<b>AUDIO_DFSDMx_Init</b> (uint32_t AudioFreq)	Initializes the Digital Filter for Sigma-Delta Modulators interface (DFSDM).
static uint8_t	<b>AUDIO_DFSDMx_DeInit</b> (void)	De-initializes the Digital Filter for Sigma-Delta Modulators interface (DFSDM).
static uint8_t	<b>AUDIO_SAIPLLConfig</b> (uint32_t Frequency)	Configures the SAI PLL clock according to the required audio frequency.
uint8_t	<b>BSP_AUDIO_OUT_Init</b> (uint16_t OutputDevice, uint8_t Volume, uint32_t AudioFreq)	Configures the audio codec related peripherals.
uint8_t	<b>BSP_AUDIO_OUT_DeInit</b> (void)	De-Initializes audio codec related peripherals.
uint8_t	<b>BSP_AUDIO_OUT_Play</b> (uint16_t *pData, uint32_t Size)	Starts playing audio stream from a data buffer for a determined size.
uint8_t	<b>BSP_AUDIO_OUT_ChangeBuffer</b> (uint16_t *pData, uint16_t Size)	Sends n-Bytes on the SAI interface.
uint8_t	<b>BSP_AUDIO_OUT_Pause</b> (void)	This function Pauses the audio file stream.
uint8_t	<b>BSP_AUDIO_OUT_Resume</b> (void)	This function Resumes the audio file stream.

uint8_t	<b>BSP_AUDIO_OUT_Stop</b> (uint32_t Option) Stops audio playing and Power down the Audio Codec.
uint8_t	<b>BSP_AUDIO_OUT_SetVolume</b> (uint8_t Volume) Controls the current audio volume level.
uint8_t	<b>BSP_AUDIO_OUT_SetMute</b> (uint32_t Cmd) Enables or disables the MUTE mode by software.
uint8_t	<b>BSP_AUDIO_OUT_SetOutputMode</b> (uint8_t Output) Switch dynamically (while audio file is being played) the output target (speaker or headphone).
uint8_t	<b>BSP_AUDIO_OUT_SetFrequency</b> (uint32_t AudioFreq) Updates the audio frequency.
void	<b>BSP_AUDIO_OUT_ChangeAudioConfig</b> (uint32_t AudioOutOption) Changes the Audio Out Configuration.
void	<b>BSP_AUDIO_OUT_RegisterCallbacks</b> (Audio_CallbackTypeDef ErrorCallback, Audio_CallbackTypeDef HalfTransferCallback, Audio_CallbackTypeDef TransferCompleteCallback) register user callback functions
void	<b>HAL_SAI_TxCpltCallback</b> (SAI_HandleTypeDef *hsai) Tx Transfer completed callbacks.
void	<b>HAL_SAI_TxHalfCpltCallback</b> (SAI_HandleTypeDef *hsai) Tx Half Transfer completed callbacks.
void	<b>HAL_SAI_ErrorCallback</b> (SAI_HandleTypeDef *hsai) SAI error callbacks.
uint8_t	<b>BSP_AUDIO_IN_Init</b> (uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnINbr) Initializes micropone related peripherals.
uint8_t	<b>BSP_AUDIO_IN_DeInit</b> (void) De-Initializes microphone related peripherals.
uint8_t	<b>BSP_AUDIO_IN_Record</b> (uint16_t *pbuf, uint32_t size)

	Starts audio recording.
uint8_t	<b>BSP_AUDIO_IN_SetFrequency</b> (uint32_t AudioFreq) Updates the audio frequency.
void	<b>HAL_DFSDM_FilterRegConvCpltCallback</b> (DFSDM_Filter_HandleTypeDef *hdfsdm_filter) Regular conversion complete callback.
void	<b>HAL_DFSDM_FilterRegConvHalfCpltCallback</b> (DFSDM_Filter_HandleTypeDef *hdfsdm_filter) Half regular conversion complete callback.
void	<b>HAL_DFSDM_FilterErrorCallback</b> (DFSDM_Filter_HandleTypeDef *hdfsdm_filter) Error callback.
uint8_t	<b>BSP_AUDIO_IN_Stop</b> (void) Stops audio recording.
uint8_t	<b>BSP_AUDIO_IN_Pause</b> (void) Pauses the audio file stream.
uint8_t	<b>BSP_AUDIO_IN_Resume</b> (void) Resumes the audio file stream.
void	<b>BSP_AUDIO_IN_RegisterCallbacks</b> (Audio_CallbackTypeDef ErrorCallback, Audio_CallbackTypeDef HalfTransferCallback, Audio_CallbackTypeDef TransferCompleteCallback) register user callback functions
void	<b>HAL_SAI_MspltInit</b> (SAI_HandleTypeDef *hsai) SAI MSP Init.
void	<b>HAL_SAI_MspDeInit</b> (SAI_HandleTypeDef *hsai) SAI MSP De-init.
void	<b>HAL_DFSDM_ChannelMspInit</b> (DFSDM_Channel_HandleTypeDef *hdfsdm_channel) Initializes the DFSDM channel MSP.
void	<b>HAL_DFSDM_ChannelMspDeInit</b> (DFSDM_Channel_HandleTypeDef *hdfsdm_channel) De-initializes the DFSDM channel MSP.
void	<b>HAL_DFSDM_FilterMspInit</b>

(DFSDM\_Filter\_HandleTypeDef \*hdfsdm\_filter)  
Initializes the DFSDM filter MSP.

---

void **HAL\_DFSDM\_FilterMspDeInit**  
(DFSDM\_Filter\_HandleTypeDef \*hdfsdm\_filter)  
De-initializes the DFSDM filter MSP.

## Variables

static <b>AUDIO_OUT_TypeDef</b>	<b>hAudioOut</b>
static <b>AUDIO_IN_TypeDef</b>	<b>hAudioIn</b>
static <b>DMA_HandleTypeDef</b>	<b>hDmaSai</b>
static <b>SAI_HandleTypeDef</b>	<b>BSP_AUDIO_hSai</b>
static <b>DFSDM_Filter_HandleTypeDef</b>	<b>BSP_AUDIO_hDfsdmLeftFilter</b>

## Detailed Description

This file provides a set of functions needed to manage the Audio driver for the STM32L476G-Discovery board.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**| Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_audio.c](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

[Defines](#) | [Typedefs](#) | [Functions](#) | [Variables](#)

## stm32l476g\_discovery\_audio.h File Reference

This file contains the common defines and functions prototypes for the **stm32l476g\_discovery\_audio.c** driver. [More...](#)

```
#include <stdlib.h> #include "../Components/cs43l22/cs43l22.h"
#include "stm32l476g_discovery.h"
```

[Go to the source code of this file.](#)



## Defines

```
#define BSP_AUDIO_OUT_CIRCULARMODE ((uint32_t)0x00000000)
#define BSP_AUDIO_OUT_NORMALMODE ((uint32_t)0x00000002)
#define BSP_AUDIO_OUT_STEREOCODE ((uint32_t)0x00000004)
#define BSP_AUDIO_OUT_MONOMODE ((uint32_t)0x00000008) /*
#define BSP_AUDIO_FREQUENCY_96K SAI_AUDIO_FREQUENCY
#define BSP_AUDIO_FREQUENCY_48K SAI_AUDIO_FREQUENCY
#define BSP_AUDIO_FREQUENCY_44K SAI_AUDIO_FREQUENCY
#define BSP_AUDIO_FREQUENCY_32K SAI_AUDIO_FREQUENCY
#define BSP_AUDIO_FREQUENCY_22K SAI_AUDIO_FREQUENCY
#define BSP_AUDIO_FREQUENCY_16K SAI_AUDIO_FREQUENCY
#define BSP_AUDIO_FREQUENCY_11K SAI_AUDIO_FREQUENCY
#define BSP_AUDIO_FREQUENCY_8K SAI_AUDIO_FREQUENCY
#define AUDIO_SAIx SAI1_Block_A
#define AUDIO_SAIx_CLK_ENABLE() __HAL_RCC_SAI1_CLK_EN
#define AUDIO_SAIx_CLK_DISABLE() __HAL_RCC_SAI1_CLK_DI
#define AUDIO_SAIx_MCK_SCK_SD_FS_AF GPIO_AF13_SAI1
#define AUDIO_SAIx_MCK_SCK_SD_FS_ENABLE() __HAL_RCC_
#define AUDIO_SAIx_MCK_SCK_SD_FS_DISABLE() __HAL_RCC_
#define AUDIO_SAIx_FS_PIN GPIO_PIN_4
#define AUDIO_SAIx_SCK_PIN GPIO_PIN_5
#define AUDIO_SAIx_SD_PIN GPIO_PIN_6
#define AUDIO_SAIx_MCK_PIN GPIO_PIN_2
#define AUDIO_SAIx_MCK_SCK_SD_FS_GPIO_PORT GPIOE
#define AUDIO_SAIx_DMAX_CLK_ENABLE() __HAL_RCC_DMA2_
#define AUDIO_SAIx_DMAX_CLK_DISABLE() __HAL_RCC_DMA2_
#define AUDIO_SAIx_DMAX_CHANNEL DMA2_Channel1
#define AUDIO_SAIx_DMAX_IRQ DMA2_Channel1_IRQn
#define AUDIO_SAIx_DMAX_PERIPH_DATA_SIZE DMA_PDATAAL
#define AUDIO_SAIx_DMAX_MEM_DATA_SIZE DMA_MDATAALIGI
#define DMA_MAX_SZE (uint32_t)0xFFFF
#define AUDIO_SAIx_DMAX_IRQHandler DMA2_Channel1_IRQHar
```

```

#define AUDIO_OUT_IRQ_PREPRIO 5 /* Select the preemption priority */
#define AUDIO_SAIx_PLL_DISABLE() HAL_RCCEx_DisablePLLSAI
#define AUDIO_DFSDMx_LEFT_CHANNEL DFSDM_Channel2
#define AUDIO_DFSDMx_LEFT_FILTER DFSDM_Filter0
#define AUDIO_DFSDMx_CLK_ENABLE() __HAL_RCC_DFSDM_CLK_ENABLE()
#define AUDIO_DFSDMx_CLK_DISABLE() __HAL_RCC_DFSDM_CLK_DISABLE()
#define AUDIO_DFSDMx_CKOUT_PIN GPIO_PIN_9
#define AUDIO_DFSDMx_DMIC_DATIN_PIN GPIO_PIN_7
#define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_PORT GPIOA
#define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()
#define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_CLK_DISABLE() __HAL_RCC_GPIOA_CLK_DISABLE()
#define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_AF GPIO_AF6_DFSMD
#define AUDIO_DFSDMx_DMAx_CLK_ENABLE() __HAL_RCC_DMA1_CLK_ENABLE()
#define AUDIO_DFSDMx_DMAx_CLK_DISABLE() __HAL_RCC_DMA1_CLK_DISABLE()
#define AUDIO_DFSDMx_DMAx_LEFT_CHANNEL DMA1_Channel4
#define AUDIO_DFSDMx_DMAx_LEFT_IRQ DMA1_Channel4_IRQn
#define AUDIO_DFSDMx_DMAx_PERIPH_DATA_SIZE DMA_PDATA1_SIZE
#define AUDIO_DFSDMx_DMAx_MEM_DATA_SIZE DMA_MDATA1_SIZE
#define AUDIO_DFSDM_DMAx_LEFT_IRQHandler DMA1_Channel4_IRQHandler
#define AUDIO_IN_IRQ_PREPRIO 6 /* Select the preemption priority */
#define AUDIODATA_SIZE 2 /* 16-bits audio data size */
#define AUDIO_OK 0
#define AUDIO_ERROR 1
#define AUDIO_TIMEOUT 2
#define DEFAULT_AUDIO_IN_FREQ BSP_AUDIO_FREQUENCY_16K
#define DEFAULT_AUDIO_IN_BIT_RESOLUTION 16
#define DEFAULT_AUDIO_IN_CHANNEL_NBR 1 /* Mono = 1, Stereo = 2 */
#define DEFAULT_AUDIO_IN_VOLUME 64
#define CODEC_RESET_DELAY 5
#define DMA_MAX(_X_) (((_X_) <= DMA_MAX_SIZE)? (_X_):DMA_MAX_SIZE)

```

## Typedefs

```
typedef void(* Audio_CallbackTypeDef)(void)
```

## Functions

uint8_t	<b>BSP_AUDIO_OUT_Init</b> (uint16_t OutputDevice, uint8_t Volume, uint32_t AudioFreq) Configures the audio codec related peripherals.
uint8_t	<b>BSP_AUDIO_OUT_DeInit</b> (void) De-Initializes audio codec related peripherals.
uint8_t	<b>BSP_AUDIO_OUT_Play</b> (uint16_t *pData, uint32_t Size) Starts playing audio stream from a data buffer for a determined size.
uint8_t	<b>BSP_AUDIO_OUT_ChangeBuffer</b> (uint16_t *pData, uint16_t Size) Sends n-Bytes on the SAI interface.
uint8_t	<b>BSP_AUDIO_OUT_Pause</b> (void) This function Pauses the audio file stream.
uint8_t	<b>BSP_AUDIO_OUT_Resume</b> (void) This function Resumes the audio file stream.
uint8_t	<b>BSP_AUDIO_OUT_Stop</b> (uint32_t Option) Stops audio playing and Power down the Audio Codec.
uint8_t	<b>BSP_AUDIO_OUT_SetVolume</b> (uint8_t Volume) Controls the current audio volume level.
uint8_t	<b>BSP_AUDIO_OUT_SetFrequency</b> (uint32_t AudioFreq) Updates the audio frequency.
void	<b>BSP_AUDIO_OUT_ChangeAudioConfig</b> (uint32_t AudioOutOption) Changes the Audio Out Configuration.
uint8_t	<b>BSP_AUDIO_OUT_SetMute</b> (uint32_t Cmd) Enables or disables the MUTE mode by software.
uint8_t	<b>BSP_AUDIO_OUT_SetOutputMode</b> (uint8_t Output) Switch dynamically (while audio file is being played) the output target (speaker or headphone).
void	<b>BSP_AUDIO_OUT_RegisterCallbacks</b> (Audio_CallbackTypeDef ErrorCallback,

**Audio\_CallbackTypeDef** HalfTransferCallback,  
**Audio\_CallbackTypeDef** TransferCompleteCallback)  
register user callback functions

uint8\_t **BSP\_AUDIO\_IN\_Init** (uint32\_t AudioFreq, uint32\_t BitRes,  
uint32\_t ChnINbr)  
Initializes micropone related peripherals.

uint8\_t **BSP\_AUDIO\_IN\_DeInit** (void)  
De-Initializes microphone related peripherals.

uint8\_t **BSP\_AUDIO\_IN\_Record** (uint16\_t \*pData, uint32\_t Size)  
Starts audio recording.

uint8\_t **BSP\_AUDIO\_IN\_SetFrequency** (uint32\_t AudioFreq)  
Updates the audio frequency.

uint8\_t **BSP\_AUDIO\_IN\_Stop** (void)  
Stops audio recording.

uint8\_t **BSP\_AUDIO\_IN\_Pause** (void)  
Pauses the audio file stream.

uint8\_t **BSP\_AUDIO\_IN\_Resume** (void)  
Resumes the audio file stream.

void **BSP\_AUDIO\_IN\_RegisterCallbacks**  
(**Audio\_CallbackTypeDef** ErrorCallback,  
**Audio\_CallbackTypeDef** HalfTransferCallback,  
**Audio\_CallbackTypeDef** TransferCompleteCallback)  
register user callback functions

## Variables

SAI_HandleTypeDef	<b>BSP_AUDIO_hSai</b>
DFSDM_Filter_HandleTypeDef	<b>BSP_AUDIO_hDfsdmLeftFilter</b>

## Detailed Description

This file contains the common defines and functions prototypes for the [stm32l476g\\_discovery\\_audio.c](#) driver.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_audio.h](#).



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
			<a href="#">Functions</a>   <a href="#">Variables</a>	

## stm32l476g\_discovery\_compass.c File Reference

This file provides a set of functions needed to manage the E-Compass (ACCELEROMETER + MAGNETOMETER) MEMS LSM303C available on STM32L476G-Discovery board. [More...](#)

```
#include "stm32l476g_discovery.h" #include
"stm32l476g_discovery_compass.h"
#include "../Components/lsm303c/lsm303c.h"
#include <math.h>
```

[Go to the source code of this file.](#)

## Functions

	void <b>ACCELERO_IO_DeInit</b> (void) De-Configures COMPASS/ACCELEROMETER io interface.
	void <b>MAGNETO_IO_DeInit</b> (void) de-Configures COMPASS/MAGNETO SPI interface.
<b>COMPASS_StatusTypeDef</b>	<b>BSP_COMPASS_Init</b> (void) Initialize the COMPASS.
	void <b>BSP_COMPASS_DeInit</b> (void) DeInitialize the COMPASS.
	void <b>BSP_COMPASS_LowPower</b> (void) Put the COMPASS in low power mode.
	void <b>BSP_COMPASS_AccGetXYZ</b> (int16_t *pDataXYZ) Get XYZ acceleration values.
	void <b>BSP_COMPASS_MagGetXYZ</b> (int16_t *pDataXYZ) Get XYZ magnetometer values.

## Variables

static ACCELERO_DrvTypeDef *	<b>AccelerometerDrv</b>
------------------------------	-------------------------

static MAGNETO_DrvTypeDef *	<b>MagnetoDrv</b>
-----------------------------	-------------------

## Detailed Description

This file provides a set of functions needed to manage the E-Compass (ACCELEROMETER + MAGNETOMETER) MEMS LSM303C available on STM32L476G-Discovery board.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_compass.c](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
			<a href="#">Enumerations</a>	<a href="#">Functions</a>

## stm32l476g\_discovery\_compass.h File Reference

This file contains definitions for [stm32l476g\\_discovery\\_compass.c](#) firmware driver. [More...](#)

```
#include "stm32l476g_discovery.h"
```

[Go to the source code of this file.](#)

## Enumerations

```
enum COMPASS_StatusTypeDef { COMPASS_OK = 0,  
                             COMPASS_ERROR = 1, COMPASS_TIMEOUT = 2 }
```

## Functions

<b>COMPASS_StatusTypeDef</b>	<b>BSP_COMPASS_Init</b> (void) Initialize the COMPASS.
void	<b>BSP_COMPASS_DeInit</b> (void) DeInitialize the COMPASS.
void	<b>BSP_COMPASS_LowPower</b> (void) Put the COMPASS in low power mode.
void	<b>BSP_COMPASS_MagGetXYZ</b> (int16_t *pDataXYZ) Get XYZ magnetometer values.
void	<b>BSP_COMPASS_AccGetXYZ</b> (int16_t *pDataXYZ) Get XYZ acceleration values.



## Detailed Description

This file contains definitions for [stm32l476g\\_discovery\\_compass.c](#) firmware driver.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_compass.h](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
<a href="#">Defines</a>   <a href="#">Functions</a>   <a href="#">Variables</a>				

## stm32l476g\_discovery\_glass\_lcd.c File Reference

This file provides a set of functions needed to manage the LCD Glass driver for the STM32L476G-Discovery board. [More...](#)

```
#include "stm32l476g_discovery_glass_lcd.h"
```

[Go to the source code of this file.](#)

## Defines

```
#define ASCII_CHAR_0 0x30 /* 0 */
```

```
#define ASCII_CHAR_AT_SYMBOL 0x40 /* @ */
```

```
#define ASCII_CHAR_LEFT_OPEN_BRACKET 0x5B /* [ */
```

```
#define ASCII_CHAR_APOSTROPHE 0x60 /* ` */
```

```
#define ASCII_CHAR_LEFT_OPEN_BRACE 0x7B /* ( */
```

## Functions

static void	<b>Convert</b> (uint8_t *Char, <b>Point_Typedef</b> Point, <b>DoublePoint_Typedef</b> Colon)	Convert an ascii char to the a LCD digit.
static void	<b>WriteChar</b> (uint8_t *ch, <b>Point_Typedef</b> Point, <b>DoublePoint_Typedef</b> Colon, <b>DigitPosition_Typedef</b> Position)	Write a character in the LCD frame buffer.
static void	<b>LCD_MspInit</b> (LCD_HandleTypeDef *hlcd)	Initialize the LCD MSP.
static void	<b>LCD_MspDeInit</b> (LCD_HandleTypeDef *hlcd)	DeInitialize the LCD MSP.
void	<b>BSP_LCD_GLASS_Init</b> (void)	Initialize the LCD GLASS relative GPIO port IOs and LCD peripheral.
void	<b>BSP_LCD_GLASS_DeInit</b> (void)	DeInitialize the LCD GLASS relative GPIO port IOs and LCD peripheral.
void	<b>BSP_LCD_GLASS_BlinkConfig</b> (uint32_t BlinkMode, uint32_t BlinkFrequency)	Configure the LCD Blink mode and Blink frequency.
void	<b>BSP_LCD_GLASS_Contrast</b> (uint32_t Contrast)	Configure the LCD contrast.
void	<b>BSP_LCD_GLASS_DisplayBar</b> (uint32_t BarId)	Display one or several bar in LCD frame buffer.
void	<b>BSP_LCD_GLASS_ClearBar</b> (uint32_t BarId)	Clear one or several bar in LCD frame buffer.
void	<b>BSP_LCD_GLASS_BarLevelConfig</b> (uint8_t BarLevel)	Configure the bar level on LCD by writing bar value in LCD frame buffer.
void	<b>BSP_LCD_GLASS_DisplayChar</b> (uint8_t *ch, <b>Point_Typedef</b> Point, <b>DoublePoint_Typedef</b> Colon,	

**DigitPosition\_Typedef** Position)

Write a character in the LCD RAM buffer.

void **BSP\_LCD\_GLASS\_DisplayString** (uint8\_t \*ptr)

Write a character string in the LCD RAM buffer.

void **BSP\_LCD\_GLASS\_DisplayStrDeci** (uint16\_t \*ptr)

Write a character string with decimal point in the LCD RAM buffer.

void **BSP\_LCD\_GLASS\_Clear** (void)

Clear the whole LCD RAM buffer.

void **BSP\_LCD\_GLASS\_ScrollSentence** (uint8\_t \*ptr, uint16\_t  
nScroll, uint16\_t ScrollSpeed)

Display a string in scrolling mode.

## Variables

__IO uint8_t	<b>bLCDGlass_KeyPressed</b> = 0
LCD_HandleTypeDef	<b>LCDHandle</b>
const uint16_t	<b>CapLetterMap</b> [26]
const uint16_t	<b>NumberMap</b> [10]
uint32_t	<b>Digit</b> [4]
uint8_t	<b>LCDBar</b> = <b>BATTERYLEVEL_FULL</b>

## Detailed Description

This file provides a set of functions needed to manage the LCD Glass driver for the STM32L476G-Discovery board.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**| Attention:**



## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_glass\\_lcd.c](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
<div>Defines   Enumerations   Functions</div>				

## stm32l476g\_discovery\_glass\_lcd.h File Reference

Header file for [stm32l476g\\_discovery\\_glass\\_lcd.c](#) module. [More...](#)

```
#include "stm32l476g_discovery.h"
```

[Go to the source code of this file.](#)

## Defines

#define	<b>COM_PER_DIGIT_NB</b>	4	LCD digit defintion.
#define	<b>SEG_PER_DIGIT_NB</b>	4	
#define	<b>LCD_MAP_CHAR_COM0_SEG_1ST_POS</b>	(1 << LCD_MAP_CHAR_COM0_SEG_1ST_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM0_SEG_2ND_POS</b>	(1 << LCD_MAP_CHAR_COM0_SEG_2ND_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM0_SEG_3RD_POS</b>	(1 << LCD_MAP_CHAR_COM0_SEG_3RD_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM0_SEG_4TH_POS</b>	(1 << LCD_MAP_CHAR_COM0_SEG_4TH_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM1_SEG_1ST_POS</b>	(1 << LCD_MAP_CHAR_COM1_SEG_1ST_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM1_SEG_2ND_POS</b>	(1 << LCD_MAP_CHAR_COM1_SEG_2ND_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM1_SEG_3RD_POS</b>	(1 << LCD_MAP_CHAR_COM1_SEG_3RD_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM1_SEG_4TH_POS</b>	(1 << LCD_MAP_CHAR_COM1_SEG_4TH_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM2_SEG_1ST_POS</b>	(1 << LCD_MAP_CHAR_COM2_SEG_1ST_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM2_SEG_2ND_POS</b>	(1 << LCD_MAP_CHAR_COM2_SEG_2ND_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM2_SEG_3RD_POS</b>	(1 << LCD_MAP_CHAR_COM2_SEG_3RD_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM2_SEG_4TH_POS</b>	(1 << LCD_MAP_CHAR_COM2_SEG_4TH_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM3_SEG_1ST_POS</b>	(1 << LCD_MAP_CHAR_COM3_SEG_1ST_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM3_SEG_2ND_POS</b>	(1 << LCD_MAP_CHAR_COM3_SEG_2ND_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM3_SEG_3RD_POS</b>	(1 <<	

	LCD_MAP_CHAR_COM3_SEG_3RD_SHIFT)	
#define	LCD_MAP_CHAR_COM3_SEG_4TH_POS (1 << LCD_MAP_CHAR_COM3_SEG_4TH_SHIFT)	
#define	LCD_MAP_CHAR_COM0_SEG_1ST_SHIFT	0x00000000
#define	LCD_MAP_CHAR_COM0_SEG_2ND_SHIFT	0x00000001
#define	LCD_MAP_CHAR_COM0_SEG_3RD_SHIFT	0x00000002
#define	LCD_MAP_CHAR_COM0_SEG_4TH_SHIFT	0x00000003
#define	LCD_MAP_CHAR_COM1_SEG_1ST_SHIFT	0x00000004
#define	LCD_MAP_CHAR_COM1_SEG_2ND_SHIFT	0x00000005
#define	LCD_MAP_CHAR_COM1_SEG_3RD_SHIFT	0x00000006
#define	LCD_MAP_CHAR_COM1_SEG_4TH_SHIFT	0x00000007
#define	LCD_MAP_CHAR_COM2_SEG_1ST_SHIFT	0x00000008
#define	LCD_MAP_CHAR_COM2_SEG_2ND_SHIFT	0x00000009
#define	LCD_MAP_CHAR_COM2_SEG_3RD_SHIFT	0x00000010
#define	LCD_MAP_CHAR_COM2_SEG_4TH_SHIFT	0x00000011
#define	LCD_MAP_CHAR_COM3_SEG_1ST_SHIFT	0x00000012
#define	LCD_MAP_CHAR_COM3_SEG_2ND_SHIFT	0x00000013
#define	LCD_MAP_CHAR_COM3_SEG_3RD_SHIFT	0x00000014
#define	LCD_MAP_CHAR_COM3_SEG_4TH_SHIFT	0x00000015
#define	LCD_DIGIT1_COM0 LCD_COM0 LCD Digit defines.	
#define	LCD_DIGIT1_COM0_SEG_MASK ~(LCD_SEG0   LCD_SEG1   LCD_SEG22   LCD_SEG23)	
#define	LCD_DIGIT1_COM1 LCD_COM1	
#define	LCD_DIGIT1_COM1_SEG_MASK ~(LCD_SEG0   LCD_SEG1   LCD_SEG22   LCD_SEG23)	
#define	LCD_DIGIT1_COM2 LCD_COM2	
#define	LCD_DIGIT1_COM2_SEG_MASK ~(LCD_SEG0   LCD_SEG1   LCD_SEG22   LCD_SEG23)	
#define	LCD_DIGIT1_COM3 LCD_COM3	
#define	LCD_DIGIT1_COM3_SEG_MASK ~(LCD_SEG0   LCD_SEG1   LCD_SEG22   LCD_SEG23)	
#define	LCD_DIGIT2_COM0 LCD_COM0	
#define	LCD_DIGIT2_COM0_SEG_MASK ~(LCD_SEG2   LCD_SEG3   LCD_SEG20   LCD_SEG21)	

```

#define LCD_DIGIT2_COM1 LCD_COM1

#define LCD_DIGIT2_COM1_SEG_MASK ~(LCD_SEG2 |
LCD_SEG3 | LCD_SEG20 | LCD_SEG21)

#define LCD_DIGIT2_COM2 LCD_COM2

#define LCD_DIGIT2_COM2_SEG_MASK ~(LCD_SEG2 |
LCD_SEG3 | LCD_SEG20 | LCD_SEG21)

#define LCD_DIGIT2_COM3 LCD_COM3

#define LCD_DIGIT2_COM3_SEG_MASK ~(LCD_SEG2 |
LCD_SEG3 | LCD_SEG20 | LCD_SEG21)

#define LCD_DIGIT3_COM0 LCD_COM0

#define LCD_DIGIT3_COM0_SEG_MASK ~(LCD_SEG4 |
LCD_SEG5 | LCD_SEG18 | LCD_SEG19)

#define LCD_DIGIT3_COM1 LCD_COM1

#define LCD_DIGIT3_COM1_SEG_MASK ~(LCD_SEG4 |
LCD_SEG5 | LCD_SEG18 | LCD_SEG19)

#define LCD_DIGIT3_COM2 LCD_COM2

#define LCD_DIGIT3_COM2_SEG_MASK ~(LCD_SEG4 |
LCD_SEG5 | LCD_SEG18 | LCD_SEG19)

#define LCD_DIGIT3_COM3 LCD_COM3

#define LCD_DIGIT3_COM3_SEG_MASK ~(LCD_SEG4 |
LCD_SEG5 | LCD_SEG18 | LCD_SEG19)

#define LCD_DIGIT4_COM0 LCD_COM0

#define LCD_DIGIT4_COM0_SEG_MASK ~(LCD_SEG6 |
LCD_SEG17)

#define LCD_DIGIT4_COM0_1 LCD_COM0_1

#define LCD_DIGIT4_COM0_1_SEG_MASK ~(LCD_SEG7 |
LCD_SEG16)

#define LCD_DIGIT4_COM1 LCD_COM1

#define LCD_DIGIT4_COM1_SEG_MASK ~(LCD_SEG6 |
LCD_SEG17)

#define LCD_DIGIT4_COM1_1 LCD_COM1_1

#define LCD_DIGIT4_COM1_1_SEG_MASK ~(LCD_SEG7 |
LCD_SEG16)

#define LCD_DIGIT4_COM2 LCD_COM2

```

```

#define LCD_DIGIT4_COM2_SEG_MASK ~(LCD_SEG6 |
LCD_SEG17)
#define LCD_DIGIT4_COM2_1 LCD_COM2_1
#define LCD_DIGIT4_COM2_1_SEG_MASK ~(LCD_SEG7 |
LCD_SEG16)
#define LCD_DIGIT4_COM3 LCD_COM3
#define LCD_DIGIT4_COM3_SEG_MASK ~(LCD_SEG6 |
LCD_SEG17)
#define LCD_DIGIT4_COM3_1 LCD_COM3_1
#define LCD_DIGIT4_COM3_1_SEG_MASK ~(LCD_SEG7 |
LCD_SEG16)
#define LCD_DIGIT5_COM0 LCD_COM0
#define LCD_DIGIT5_COM0_SEG_MASK ~(LCD_SEG9 |
LCD_SEG14)
#define LCD_DIGIT5_COM0_1 LCD_COM0_1
#define LCD_DIGIT5_COM0_1_SEG_MASK ~(LCD_SEG8 |
LCD_SEG15)
#define LCD_DIGIT5_COM1 LCD_COM1
#define LCD_DIGIT5_COM1_SEG_MASK ~(LCD_SEG9 |
LCD_SEG14)
#define LCD_DIGIT5_COM1_1 LCD_COM1_1
#define LCD_DIGIT5_COM1_1_SEG_MASK ~(LCD_SEG8 |
LCD_SEG15)
#define LCD_DIGIT5_COM2 LCD_COM2
#define LCD_DIGIT5_COM2_SEG_MASK ~(LCD_SEG9 |
LCD_SEG14)
#define LCD_DIGIT5_COM2_1 LCD_COM2_1
#define LCD_DIGIT5_COM2_1_SEG_MASK ~(LCD_SEG8 |
LCD_SEG15)
#define LCD_DIGIT5_COM3 LCD_COM3
#define LCD_DIGIT5_COM3_SEG_MASK ~(LCD_SEG9 |
LCD_SEG14)
#define LCD_DIGIT5_COM3_1 LCD_COM3_1
#define LCD_DIGIT5_COM3_1_SEG_MASK ~(LCD_SEG8 |
LCD_SEG15)

```

```

#define LCD_DIGIT6_COM0 LCD_COM0
#define LCD_DIGIT6_COM0_SEG_MASK ~(LCD_SEG10 |
LCD_SEG11 | LCD_SEG12 | LCD_SEG13)

#define LCD_DIGIT6_COM1 LCD_COM1
#define LCD_DIGIT6_COM1_SEG_MASK ~(LCD_SEG10 |
LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
#define LCD_DIGIT6_COM2 LCD_COM2
#define LCD_DIGIT6_COM2_SEG_MASK ~(LCD_SEG10 |
LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
#define LCD_DIGIT6_COM3 LCD_COM3
#define LCD_DIGIT6_COM3_SEG_MASK ~(LCD_SEG10 |
LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
#define LCD_BAR0_2_COM LCD_COM3
LCD Bar location.

#define LCD_BAR1_3_COM LCD_COM2
#define LCD_BAR0_SEG LCD_SEG11
#define LCD_BAR1_SEG LCD_SEG11
#define LCD_BAR2_SEG LCD_SEG9
#define LCD_BAR3_SEG LCD_SEG9
#define LCD_BAR0_2_SEG_MASK ~(LCD_BAR0_SEG |
LCD_BAR2_SEG)
#define LCD_BAR1_3_SEG_MASK ~(LCD_BAR1_SEG |
LCD_BAR3_SEG)
#define LCD_COM0 MCU_LCD_COM0
LCD segments & coms redefinition.

#define LCD_COM0_1 MCU_LCD_COM0_1
#define LCD_COM1 MCU_LCD_COM1
#define LCD_COM1_1 MCU_LCD_COM1_1
#define LCD_COM2 MCU_LCD_COM2
#define LCD_COM2_1 MCU_LCD_COM2_1
#define LCD_COM3 MCU_LCD_COM3
#define LCD_COM3_1 MCU_LCD_COM3_1
#define LCD_SEG0 MCU_LCD_SEG4
#define LCD_SEG1 MCU_LCD_SEG23

```

#define	LCD_SEG2	MCU_LCD_SEG6
#define	LCD_SEG3	MCU_LCD_SEG13
#define	LCD_SEG4	MCU_LCD_SEG15
#define	LCD_SEG5	MCU_LCD_SEG29
#define	LCD_SEG6	MCU_LCD_SEG31
#define	LCD_SEG7	MCU_LCD_SEG33
#define	LCD_SEG8	MCU_LCD_SEG35
#define	LCD_SEG9	MCU_LCD_SEG25
#define	LCD_SEG10	MCU_LCD_SEG17
#define	LCD_SEG11	MCU_LCD_SEG8
#define	LCD_SEG12	MCU_LCD_SEG9
#define	LCD_SEG13	MCU_LCD_SEG26
#define	LCD_SEG14	MCU_LCD_SEG24
#define	LCD_SEG15	MCU_LCD_SEG34
#define	LCD_SEG16	MCU_LCD_SEG32
#define	LCD_SEG17	MCU_LCD_SEG30
#define	LCD_SEG18	MCU_LCD_SEG28
#define	LCD_SEG19	MCU_LCD_SEG14
#define	LCD_SEG20	MCU_LCD_SEG12
#define	LCD_SEG21	MCU_LCD_SEG5
#define	LCD_SEG22	MCU_LCD_SEG22
#define	LCD_SEG23	MCU_LCD_SEG3
#define	LCD_SEG0_SHIFT	MCU_LCD_SEG4_SHIFT
#define	LCD_SEG1_SHIFT	MCU_LCD_SEG23_SHIFT
#define	LCD_SEG2_SHIFT	MCU_LCD_SEG6_SHIFT
#define	LCD_SEG3_SHIFT	MCU_LCD_SEG13_SHIFT
#define	LCD_SEG4_SHIFT	MCU_LCD_SEG15_SHIFT
#define	LCD_SEG5_SHIFT	MCU_LCD_SEG29_SHIFT
#define	LCD_SEG6_SHIFT	MCU_LCD_SEG31_SHIFT
#define	LCD_SEG7_SHIFT	MCU_LCD_SEG33_SHIFT
#define	LCD_SEG8_SHIFT	MCU_LCD_SEG35_SHIFT
#define	LCD_SEG9_SHIFT	MCU_LCD_SEG25_SHIFT
#define	LCD_SEG10_SHIFT	MCU_LCD_SEG17_SHIFT
#define	LCD_SEG11_SHIFT	MCU_LCD_SEG8_SHIFT



```

#define LCD_SEG12_SHIFT MCU_LCD_SEG9_SHIFT
#define LCD_SEG13_SHIFT MCU_LCD_SEG26_SHIFT
#define LCD_SEG14_SHIFT MCU_LCD_SEG24_SHIFT
#define LCD_SEG15_SHIFT MCU_LCD_SEG34_SHIFT
#define LCD_SEG16_SHIFT MCU_LCD_SEG32_SHIFT
#define LCD_SEG17_SHIFT MCU_LCD_SEG30_SHIFT
#define LCD_SEG18_SHIFT MCU_LCD_SEG28_SHIFT
#define LCD_SEG19_SHIFT MCU_LCD_SEG14_SHIFT
#define LCD_SEG20_SHIFT MCU_LCD_SEG12_SHIFT
#define LCD_SEG21_SHIFT MCU_LCD_SEG5_SHIFT
#define LCD_SEG22_SHIFT MCU_LCD_SEG22_SHIFT
#define LCD_SEG23_SHIFT MCU_LCD_SEG3_SHIFT
#define MCU_LCD_COM0 LCD_RAM_REGISTER0
STM32 LCD segments & coms definitions.
#define MCU_LCD_COM0_1 LCD_RAM_REGISTER1
#define MCU_LCD_COM1 LCD_RAM_REGISTER2
#define MCU_LCD_COM1_1 LCD_RAM_REGISTER3
#define MCU_LCD_COM2 LCD_RAM_REGISTER4
#define MCU_LCD_COM2_1 LCD_RAM_REGISTER5
#define MCU_LCD_COM3 LCD_RAM_REGISTER6
#define MCU_LCD_COM3_1 LCD_RAM_REGISTER7
#define MCU_LCD_COM4 LCD_RAM_REGISTER8
#define MCU_LCD_COM4_1 LCD_RAM_REGISTER9
#define MCU_LCD_COM5 LCD_RAM_REGISTER10
#define MCU_LCD_COM5_1 LCD_RAM_REGISTER11
#define MCU_LCD_COM6 LCD_RAM_REGISTER12
#define MCU_LCD_COM6_1 LCD_RAM_REGISTER13
#define MCU_LCD_COM7 LCD_RAM_REGISTER14
#define MCU_LCD_COM7_1 LCD_RAM_REGISTER15
#define MCU_LCD_SEG0 (1U << MCU_LCD_SEG0_SHIFT)
#define MCU_LCD_SEG1 (1U << MCU_LCD_SEG1_SHIFT)
#define MCU_LCD_SEG2 (1U << MCU_LCD_SEG2_SHIFT)
#define MCU_LCD_SEG3 (1U << MCU_LCD_SEG3_SHIFT)
#define MCU_LCD_SEG4 (1U << MCU_LCD_SEG4_SHIFT)

```

#define	<b>MCU_LCD_SEG5</b>	(1U << MCU_LCD_SEG5_SHIFT)
#define	<b>MCU_LCD_SEG6</b>	(1U << MCU_LCD_SEG6_SHIFT)
#define	<b>MCU_LCD_SEG7</b>	(1U << MCU_LCD_SEG7_SHIFT)
#define	<b>MCU_LCD_SEG8</b>	(1U << MCU_LCD_SEG8_SHIFT)
#define	<b>MCU_LCD_SEG9</b>	(1U << MCU_LCD_SEG9_SHIFT)
#define	<b>MCU_LCD_SEG10</b>	(1U << MCU_LCD_SEG10_SHIFT)
#define	<b>MCU_LCD_SEG11</b>	(1U << MCU_LCD_SEG11_SHIFT)
#define	<b>MCU_LCD_SEG12</b>	(1U << MCU_LCD_SEG12_SHIFT)
#define	<b>MCU_LCD_SEG13</b>	(1U << MCU_LCD_SEG13_SHIFT)
#define	<b>MCU_LCD_SEG14</b>	(1U << MCU_LCD_SEG14_SHIFT)
#define	<b>MCU_LCD_SEG15</b>	(1U << MCU_LCD_SEG15_SHIFT)
#define	<b>MCU_LCD_SEG16</b>	(1U << MCU_LCD_SEG16_SHIFT)
#define	<b>MCU_LCD_SEG17</b>	(1U << MCU_LCD_SEG17_SHIFT)
#define	<b>MCU_LCD_SEG18</b>	(1U << MCU_LCD_SEG18_SHIFT)
#define	<b>MCU_LCD_SEG19</b>	(1U << MCU_LCD_SEG19_SHIFT)
#define	<b>MCU_LCD_SEG20</b>	(1U << MCU_LCD_SEG20_SHIFT)
#define	<b>MCU_LCD_SEG21</b>	(1U << MCU_LCD_SEG21_SHIFT)
#define	<b>MCU_LCD_SEG22</b>	(1U << MCU_LCD_SEG22_SHIFT)
#define	<b>MCU_LCD_SEG23</b>	(1U << MCU_LCD_SEG23_SHIFT)
#define	<b>MCU_LCD_SEG24</b>	(1U << MCU_LCD_SEG24_SHIFT)
#define	<b>MCU_LCD_SEG25</b>	(1U << MCU_LCD_SEG25_SHIFT)
#define	<b>MCU_LCD_SEG26</b>	(1U << MCU_LCD_SEG26_SHIFT)
#define	<b>MCU_LCD_SEG27</b>	(1U << MCU_LCD_SEG27_SHIFT)
#define	<b>MCU_LCD_SEG28</b>	(1U << MCU_LCD_SEG28_SHIFT)
#define	<b>MCU_LCD_SEG29</b>	(1U << MCU_LCD_SEG29_SHIFT)
#define	<b>MCU_LCD_SEG30</b>	(1U << MCU_LCD_SEG30_SHIFT)
#define	<b>MCU_LCD_SEG31</b>	(1U << MCU_LCD_SEG31_SHIFT)
#define	<b>MCU_LCD_SEG32</b>	(1U << MCU_LCD_SEG32_SHIFT)
#define	<b>MCU_LCD_SEG33</b>	(1U << MCU_LCD_SEG33_SHIFT)
#define	<b>MCU_LCD_SEG34</b>	(1U << MCU_LCD_SEG34_SHIFT)
#define	<b>MCU_LCD_SEG35</b>	(1U << MCU_LCD_SEG35_SHIFT)
#define	<b>MCU_LCD_SEG36</b>	(1U << MCU_LCD_SEG36_SHIFT)
#define	<b>MCU_LCD_SEG37</b>	(1U << MCU_LCD_SEG37_SHIFT)

#define	MCU_LCD_SEG38	(1U << MCU_LCD_SEG38_SHIFT)
#define	MCU_LCD_SEG0_SHIFT	0
#define	MCU_LCD_SEG1_SHIFT	1
#define	MCU_LCD_SEG2_SHIFT	2
#define	MCU_LCD_SEG3_SHIFT	3
#define	MCU_LCD_SEG4_SHIFT	4
#define	MCU_LCD_SEG5_SHIFT	5
#define	MCU_LCD_SEG6_SHIFT	6
#define	MCU_LCD_SEG7_SHIFT	7
#define	MCU_LCD_SEG8_SHIFT	8
#define	MCU_LCD_SEG9_SHIFT	9
#define	MCU_LCD_SEG10_SHIFT	10
#define	MCU_LCD_SEG11_SHIFT	11
#define	MCU_LCD_SEG12_SHIFT	12
#define	MCU_LCD_SEG13_SHIFT	13
#define	MCU_LCD_SEG14_SHIFT	14
#define	MCU_LCD_SEG15_SHIFT	15
#define	MCU_LCD_SEG16_SHIFT	16
#define	MCU_LCD_SEG17_SHIFT	17
#define	MCU_LCD_SEG18_SHIFT	18
#define	MCU_LCD_SEG19_SHIFT	19
#define	MCU_LCD_SEG20_SHIFT	20
#define	MCU_LCD_SEG21_SHIFT	21
#define	MCU_LCD_SEG22_SHIFT	22
#define	MCU_LCD_SEG23_SHIFT	23
#define	MCU_LCD_SEG24_SHIFT	24
#define	MCU_LCD_SEG25_SHIFT	25
#define	MCU_LCD_SEG26_SHIFT	26
#define	MCU_LCD_SEG27_SHIFT	27
#define	MCU_LCD_SEG28_SHIFT	28
#define	MCU_LCD_SEG29_SHIFT	29
#define	MCU_LCD_SEG30_SHIFT	30
#define	MCU_LCD_SEG31_SHIFT	31
#define	MCU_LCD_SEG32_SHIFT	0

```

#define MCU_LCD_SEG33_SHIFT 1
#define MCU_LCD_SEG34_SHIFT 2
#define MCU_LCD_SEG35_SHIFT 3
#define MCU_LCD_SEG36_SHIFT 4
#define MCU_LCD_SEG37_SHIFT 5
#define MCU_LCD_SEG38_SHIFT 6
#define MCU_LCD_SEG39_SHIFT 7
#define MCU_LCD_SEG40_SHIFT 8
#define MCU_LCD_SEG41_SHIFT 9
#define MCU_LCD_SEG42_SHIFT 10
#define MCU_LCD_SEG43_SHIFT 11
#define LCD_GPIO_BANKA_PINS
    LCD Pins definition.
#define LCD_GPIO_BANKB_PINS
#define LCD_GPIO_BANKC_PINS
#define LCD_GPIO_BANKD_PINS
#define SCROLL_SPEED_HIGH 150
#define SCROLL_SPEED_MEDIUM 300
#define SCROLL_SPEED_LOW 450
#define DOT ((uint16_t) 0x8000 ) /* for add decimal point in string */
#define DOUBLE_DOT ((uint16_t) 0x4000) /* for add decimal point
    in string */
#define C_OPENPARMAP ((uint16_t) 0x0028)
#define C_CLOSEPARMAP ((uint16_t) 0x0011)
#define C_DMAP ((uint16_t) 0xf300)
#define C_MMAP ((uint16_t) 0xb210)
#define C_NMAP ((uint16_t) 0x2210)
#define C_UMAP ((uint16_t) 0x6084)
#define C_STAR ((uint16_t) 0xA0DD)
#define C_MINUS ((uint16_t) 0xA000)
#define C_PLUS ((uint16_t) 0xA014)
#define C_SLATCH ((uint16_t) 0x00c0)
#define C_PERCENT_1 ((uint16_t) 0xec00)
#define C_PERCENT_2 ((uint16_t) 0xb300)

```

```
#define C_FULL ((uint16_t) 0xffdd)
```

## Enumerations

enum	<b>DigitPosition_Typedef</b> { <b>LCD_DIGIT_POSITION_1</b> = 0, <b>LCD_DIGIT_POSITION_2</b> = 1, <b>LCD_DIGIT_POSITION_3</b> = 2, <b>LCD_DIGIT_POSITION_4</b> = 3, <b>LCD_DIGIT_POSITION_5</b> = 4, <b>LCD_DIGIT_POSITION_6</b> = 5, <b>LCD_DIGIT_MAX_NUMBER</b> = 6 }	LCD Glass digit position. <a href="#">More...</a>
enum	<b>Point_Typedef</b> { <b>POINT_OFF</b> = 0, <b>POINT_ON</b> = 1 }	LCD Glass point Warning: element values correspond to LCD Glass point. <a href="#">More...</a>
enum	<b>DoublePoint_Typedef</b> { <b>DOUBLEPOINT_OFF</b> = 0, <b>DOUBLEPOINT_ON</b> = 1 }	LCD Glass Double point Warning: element values correspond to LCD Glass Double point. <a href="#">More...</a>
enum	<b>BatteryLevel_Typedef</b> { <b>BATTERYLEVEL_OFF</b> = 0, <b>BATTERYLEVEL_1_4</b> = 1, <b>BATTERYLEVEL_1_2</b> = 2, <b>BATTERYLEVEL_3_4</b> = 3, <b>BATTERYLEVEL_FULL</b> = 4 }	LCD Glass Battery Level element values correspond to different LCD Glass battery levels. <a href="#">More...</a>
enum	<b>BarId_Typedef</b> { <b>LCD_BAR_NONE</b> = 0, <b>LCD_BAR_0</b> = (1 << 0), <b>LCD_BAR_1</b> = (1 << 1), <b>LCD_BAR_2</b> = (1 << 2), <b>LCD_BAR_3</b> = (1 << 3) }	LCD Glass Bar Id. <a href="#">More...</a>

## Functions

void	<b>BSP_LCD_GLASS_Init</b> (void)	Initialize the LCD GLASS relative GPIO port IOs and LCD peripheral.
void	<b>BSP_LCD_GLASS_DeInit</b> (void)	DeInitialize the LCD GLASS relative GPIO port IOs and LCD peripheral.
void	<b>BSP_LCD_GLASS_BlinkConfig</b> (uint32_t BlinkMode, uint32_t BlinkFrequency)	Configure the LCD Blink mode and Blink frequency.
void	<b>BSP_LCD_GLASS_Contrast</b> (uint32_t Contrast)	Configure the LCD contrast.
void	<b>BSP_LCD_GLASS_DisplayChar</b> (uint8_t *ch, <b>Point_Typedef</b> Point, <b>DoublePoint_Typedef</b> Colon, <b>DigitPosition_Typedef</b> Position)	Write a character in the LCD RAM buffer.
void	<b>BSP_LCD_GLASS_DisplayString</b> (uint8_t *ptr)	Write a character string in the LCD RAM buffer.
void	<b>BSP_LCD_GLASS_DisplayStrDeci</b> (uint16_t *ptr)	Write a character string with decimal point in the LCD RAM buffer.
void	<b>BSP_LCD_GLASS_ScrollSentence</b> (uint8_t *ptr, uint16_t nScroll, uint16_t ScrollSpeed)	Display a string in scrolling mode.
void	<b>BSP_LCD_GLASS_DisplayBar</b> (uint32_t BarId)	Display one or several bar in LCD frame buffer.
void	<b>BSP_LCD_GLASS_ClearBar</b> (uint32_t BarId)	Clear one or several bar in LCD frame buffer.
void	<b>BSP_LCD_GLASS_BarLevelConfig</b> (uint8_t BarLevel)	Configure the bar level on LCD by writing bar value in LCD frame buffer.
void	<b>BSP_LCD_GLASS_Clear</b> (void)	

Clear the whole LCD RAM buffer.

---



## Detailed Description

Header file for [stm32l476g\\_discovery\\_glass\\_lcd.c](#) module.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
			<a href="#">Functions</a>   <a href="#">Variables</a>	

## stm32l476g\_discovery\_gyroscope.c File Reference

This file provides a set of functions needed to manage the L3GD20 MEMS accelerometer available on STM32L476G-Discovery board.  
[More...](#)

```
#include "stm32l476g_discovery_gyroscope.h"
```

[Go to the source code of this file.](#)

## Functions

uint8_t	<b>BSP_GYRO_Init</b> (void)	Initialize Gyroscope.
void	<b>BSP_GYRO_DeInit</b> (void)	DeInitialize Gyroscope.
void	<b>BSP_GYRO_LowPower</b> (void)	Put Gyroscope in low power mode.
uint8_t	<b>BSP_GYRO_ReadID</b> (void)	Read ID of Gyroscope component.
void	<b>BSP_GYRO_Reset</b> (void)	Reboot memory content of Gyroscope.
void	<b>BSP_GYRO_ITConfig</b> (GYRO_InterruptConfigTypeDef *pIntConfig)	Configure Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_EnableIT</b> (uint8_t IntPin)	Enable Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_DisableIT</b> (uint8_t IntPin)	Disable Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_GetXYZ</b> (float *pfData)	Get XYZ angular acceleration from the Gyroscope.

## Variables

---

static GYRO_DrvTypeDef *	<b>GyroscopeDrv</b>
--------------------------	---------------------

---

## Detailed Description

This file provides a set of functions needed to manage the L3GD20 MEMS accelerometer available on STM32L476G-Discovery board.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_gyroscope.c](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
			<a href="#">Enumerations</a>	<a href="#">Functions</a>

## stm32l476g\_discovery\_gyroscope.h File Reference

This file contains definitions for [stm32l476g\\_discovery\\_gyroscope.c](#) firmware driver. [More...](#)

```
#include "stm32l476g_discovery.h" #include
"../Components/l3gd20/l3gd20.h"
```

[Go to the source code of this file.](#)



## Enumerations

```
enum GYRO_StatusTypeDef { GYRO_OK = 0, GYRO_ERROR = 1,  
GYRO_TIMEOUT = 2 }
```

## Functions

uint8_t	<b>BSP_GYRO_Init</b> (void)	Initialize Gyroscope.
void	<b>BSP_GYRO_DeInit</b> (void)	DeInitialize Gyroscope.
void	<b>BSP_GYRO_LowPower</b> (void)	Put Gyroscope in low power mode.
void	<b>BSP_GYRO_Reset</b> (void)	Reboot memory content of Gyroscope.
uint8_t	<b>BSP_GYRO_ReadID</b> (void)	Read ID of Gyroscope component.
void	<b>BSP_GYRO_ITConfig</b> (GYRO_InterruptConfigTypeDef *pIntConfig)	Configure Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_EnableIT</b> (uint8_t IntPin)	Enable Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_DisableIT</b> (uint8_t IntPin)	Disable Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_GetXYZ</b> (float *pfData)	Get XYZ angular acceleration from the Gyroscope.

## Detailed Description

This file contains definitions for [stm32l476g\\_discovery\\_gyroscope.c](#) firmware driver.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_gyroscope.h](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
			<a href="#">Functions</a>   <a href="#">Variables</a>	

## stm32l476g\_discovery\_idd.c File Reference

This file provides a set of firmware functions to manage the Idd measurement driver for STM32L476G-Discovery board. [More...](#)

```
#include "stm32l476g_discovery_idd.h"
```

[Go to the source code of this file.](#)

## Functions

uint8_t	<b>BSP_IDD_Init</b> (void)	Configures IDD measurement component.
void	<b>BSP_IDD_DeInit</b> (void)	Unconfigures IDD measurement component.
void	<b>BSP_IDD_Reset</b> (void)	Reset Idd measurement component.
void	<b>BSP_IDD_LowPower</b> (void)	Turn Idd measurement component in low power (standby/sleep) mode.
void	<b>BSP_IDD_StartMeasure</b> (void)	Start Measurement campaign.
void	<b>BSP_IDD_Config</b> (IDD_ConfigTypeDef IddConfig)	Configure Idd component.
void	<b>BSP_IDD_GetValue</b> (uint32_t *IddValue)	Get Idd current value.
void	<b>BSP_IDD_EnableIT</b> (void)	Enable Idd interrupt that warn end of measurement.
void	<b>BSP_IDD_ClearIT</b> (void)	Clear Idd interrupt that warn end of measurement.
uint8_t	<b>BSP_IDD_GetITStatus</b> (void)	Get Idd interrupt status.
void	<b>BSP_IDD_DisableIT</b> (void)	Disable Idd interrupt that warn end of measurement.
uint8_t	<b>BSP_IDD_ErrorGetCode</b> (void)	Get Error Code .
void	<b>BSP_IDD_ErrorEnableIT</b> (void)	Enable error interrupt that warn end of measurement.
void	<b>BSP_IDD_ErrorClearIT</b> (void)	Clear Error interrupt that warn end of measurement.
uint8_t	<b>BSP_IDD_ErrorGetITStatus</b> (void)	

Get Error interrupt status.

void **BSP\_IDD\_ErrorDisableIT** (void)

Disable Error interrupt.

void **BSP\_IDD\_WakeUp** (void)

Wake up Idd measurement component.

## Variables

```
static IDD_DrvTypeDef * IddDrv
```

---



## Detailed Description

This file provides a set of firmware functions to manage the Idd measurement driver for STM32L476G-Discovery board.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_idd.c](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
<div>Defines   Enumerations   Functions</div>				

## stm32l476g\_discovery\_idd.h File Reference

Header file for [stm32l476g\\_discovery\\_idd.c](#) module. [More...](#)

```
#include "stm32l476g_discovery.h" #include
"../Components/mfxstm32l152/mfxstm32l152.h"
```

[Go to the source code of this file.](#)

## Defines

#define	<b>DISCOVERY_IDD_SHUNT0_VALUE</b>	((uint16_t) 1000)	Shunt values on discovery in milli ohms.
#define	<b>DISCOVERY_IDD_SHUNT1_VALUE</b>	((uint16_t) 24)	
#define	<b>DISCOVERY_IDD_SHUNT2_VALUE</b>	((uint16_t) 620)	
#define	<b>DISCOVERY_IDD_SHUNT4_VALUE</b>	((uint16_t) 10000)	
#define	<b>DISCOVERY_IDD_SHUNT0_STABDELAY</b>	((uint8_t) 149)	Shunt stabilization delay on discovery in milli ohms.
#define	<b>DISCOVERY_IDD_SHUNT1_STABDELAY</b>	((uint8_t) 149)	
#define	<b>DISCOVERY_IDD_SHUNT2_STABDELAY</b>	((uint8_t) 149)	
#define	<b>DISCOVERY_IDD_SHUNT4_STABDELAY</b>	((uint8_t) 255)	
#define	<b>DISCOVERY_IDD_AMPLI_GAIN</b>	((uint16_t) 4967)	IDD Ampli Gain on discovery.
#define	<b>DISCOVERY_IDD_VDD_MIN</b>	((uint16_t) 2000)	IDD Vdd Min on discovery.

## Enumerations

```
enum  IDD_StatusTypeDef { IDD_OK = 0, IDD_TIMEOUT = 1,  
    IDD_ZERO_VALUE = 2, IDD_ERROR = 0xFF }
```

## Functions

uint8_t	<b>BSP_IDD_Init</b> (void)	Configures IDD measurement component.
void	<b>BSP_IDD_DeInit</b> (void)	Unconfigures IDD measurement component.
void	<b>BSP_IDD_Reset</b> (void)	Reset Idd measurement component.
void	<b>BSP_IDD_LowPower</b> (void)	Turn Idd measurement component in low power (standby/sleep) mode.
void	<b>BSP_IDD_WakeUp</b> (void)	Wake up Idd measurement component.
void	<b>BSP_IDD_StartMeasure</b> (void)	Start Measurement campaign.
void	<b>BSP_IDD_Config</b> (IDD_ConfigTypeDef IddConfig)	Configure Idd component.
void	<b>BSP_IDD_GetValue</b> (uint32_t *IddValue)	Get Idd current value.
void	<b>BSP_IDD_EnableIT</b> (void)	Enable Idd interrupt that warn end of measurement.
void	<b>BSP_IDD_ClearIT</b> (void)	Clear Idd interrupt that warn end of measurement.
uint8_t	<b>BSP_IDD_GetITStatus</b> (void)	Get Idd interrupt status.
void	<b>BSP_IDD_DisableIT</b> (void)	Disable Idd interrupt that warn end of measurement.
uint8_t	<b>BSP_IDD_ErrorGetCode</b> (void)	Get Error Code .
void	<b>BSP_IDD_ErrorEnableIT</b> (void)	Enable error interrupt that warn end of measurement.
void	<b>BSP_IDD_ErrorClearIT</b> (void)	

Clear Error interrupt that warn end of measurement.

uint8\_t **BSP\_IDD\_ErrorGetITStatus** (void)

Get Error interrupt status.

void **BSP\_IDD\_ErrorDisableIT** (void)

Disable Error interrupt.

---

## Detailed Description

Header file for [stm32l476g\\_discovery\\_idd.c](#) module.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**



## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_idd.h](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		
			<a href="#">Functions</a>   <a href="#">Variables</a>	

## stm32l476g\_discovery\_qspi.c File Reference

This file includes a standard driver for the N25Q128A QSPI memory mounted on STM32L476G-Discovery board. [More...](#)

```
#include "stm32l476g_discovery_qspi.h"
```

[Go to the source code of this file.](#)

## Functions

static void	<b>QSPI_MsplInit</b> (void) Initializes the QSPI MSP.
static void	<b>QSPI_MspDeInit</b> (void) De-Initializes the QSPI MSP.
static uint8_t	<b>QSPI_ResetMemory</b> (QSPI_HandleTypeDef *hqspi) This function reset the QSPI memory.
static uint8_t	<b>QSPI_DummyCyclesCfg</b> (QSPI_HandleTypeDef *hqspi) This function configure the dummy cycles on memory side.
static uint8_t	<b>QSPI_WriteEnable</b> (QSPI_HandleTypeDef *hqspi) This function send a Write Enable and wait it is effective.
static uint8_t	<b>QSPI_AutoPollingMemReady</b> (QSPI_HandleTypeDef *hqspi, uint32_t Timeout) This function read the SR of the memory and wait the EOP.
uint8_t	<b>BSP_QSPI_Init</b> (void) Initializes the QSPI interface.
uint8_t	<b>BSP_QSPI_DeInit</b> (void) De-Initializes the QSPI interface.
uint8_t	<b>BSP_QSPI_Read</b> (uint8_t *pData, uint32_t ReadAddr, uint32_t Size) Reads an amount of data from the QSPI memory.
uint8_t	<b>BSP_QSPI_Write</b> (uint8_t *pData, uint32_t WriteAddr, uint32_t Size) Writes an amount of data to the QSPI memory.
uint8_t	<b>BSP_QSPI_Erase_Block</b> (uint32_t BlockAddress) Erases the specified block of the QSPI memory.
uint8_t	<b>BSP_QSPI_Erase_Sector</b> (uint32_t Sector) Erases the specified sector of the QSPI memory.

uint8_t	<b>BSP_QSPI_Erase_Chip</b> (void)	Erases the entire QSPI memory.
uint8_t	<b>BSP_QSPI_GetStatus</b> (void)	Reads current status of the QSPI memory.
uint8_t	<b>BSP_QSPI_GetInfo</b> (QSPI_Info *pInfo)	Return the configuration of the QSPI memory.
uint8_t	<b>BSP_QSPI_EnableMemoryMappedMode</b> (void)	Configure the QSPI in memory-mapped mode.
uint8_t	<b>BSP_QSPI_SuspendErase</b> (void)	This function suspends an ongoing erase command.
uint8_t	<b>BSP_QSPI_ResumeErase</b> (void)	This function resumes a paused erase command.

## Variables

QSPI_HandleTypeDef	<b>QSPIHandle</b>
--------------------	-------------------

---

## Detailed Description

This file includes a standard driver for the N25Q128A QSPI memory mounted on STM32L476G-Discovery board.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

```
=====
=====
##### How to use this driver #####
=====
=====
[.]
( # ) This driver is used to drive the N25Q128A QSPI external
      memory mounted on STM32L476G-DISCO evaluation board.

( # ) This driver need a specific component driver (N25Q128A) to be included with.

( # ) Initialization steps:
      (++) Initialize the QPSI external memory using the BSP_QSPI_Init() function. This
           function includes the MSP layer hardware resources initialization and the
           QSPI interface with the external memory.
```

(#) QSPI memory operations

- (++) QSPI memory can be accessed with read/write operations once it is initialized.  
Read/write operation can be performed with AHB access using the functions BSP\_QSPI\_Read()/BSP\_QSPI\_Write().
- (++) The function BSP\_QSPI\_GetInfo() returns the configuration of the QSPI memory.  
(see the QSPI memory data sheet)
- (++) Perform erase block operation using the function BSP\_QSPI\_Erase\_Block() and by specifying the block address. You can perform an erase operation of the whole chip by calling the function BSP\_QSPI\_Erase\_Chip().
- (++) The function BSP\_QSPI\_GetStatus() returns the current status of the QSPI memory.  
  
(see the QSPI memory data sheet)

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_qspi.c](#).



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

[Data Structures](#) | [Defines](#) | [Functions](#)

## stm32l476g\_discovery\_qspi.h File Reference

This file contains the common defines and functions prototypes for the **stm32l476g\_discovery\_qspi.c** driver. [More...](#)

```
#include "stm32l4xx_hal.h" #include
"../Components/n25q128a/n25q128a.h"
```

[Go to the source code of this file.](#)

## Data Structures

```
struct QSPI_Info
```

## Defines

#define	<b>QSPI_OK</b>	((uint8_t)0x00)
#define	<b>QSPI_ERROR</b>	((uint8_t)0x01)
#define	<b>QSPI_BUSY</b>	((uint8_t)0x02)
#define	<b>QSPI_NOT_SUPPORTED</b>	((uint8_t)0x04)
#define	<b>QSPI_SUSPENDED</b>	((uint8_t)0x08)

## Functions

uint8_t	<b>BSP_QSPI_Init</b> (void)	Initializes the QSPI interface.
uint8_t	<b>BSP_QSPI_DeInit</b> (void)	De-Initializes the QSPI interface.
uint8_t	<b>BSP_QSPI_Read</b> (uint8_t *pData, uint32_t ReadAddr, uint32_t Size)	Reads an amount of data from the QSPI memory.
uint8_t	<b>BSP_QSPI_Write</b> (uint8_t *pData, uint32_t WriteAddr, uint32_t Size)	Writes an amount of data to the QSPI memory.
uint8_t	<b>BSP_QSPI_Erase_Block</b> (uint32_t BlockAddress)	Erases the specified block of the QSPI memory.
uint8_t	<b>BSP_QSPI_Erase_Sector</b> (uint32_t Sector)	Erases the specified sector of the QSPI memory.
uint8_t	<b>BSP_QSPI_Erase_Chip</b> (void)	Erases the entire QSPI memory.
uint8_t	<b>BSP_QSPI_GetStatus</b> (void)	Reads current status of the QSPI memory.
uint8_t	<b>BSP_QSPI_GetInfo</b> (QSPI_Info *pInfo)	Return the configuration of the QSPI memory.
uint8_t	<b>BSP_QSPI_EnableMemoryMappedMode</b> (void)	Configure the QSPI in memory-mapped mode.
uint8_t	<b>BSP_QSPI_SuspendErase</b> (void)	This function suspends an ongoing erase command.
uint8_t	<b>BSP_QSPI_ResumeErase</b> (void)	This function resumes a paused erase command.

## Detailed Description

This file contains the common defines and functions prototypes for the [stm32l476g\\_discovery\\_qspi.c](#) driver.

**Author:**

MCD Application Team

**Version:**

\$VERSION\$

**Date:**

\$DATE\$

**Attention:**

## © COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32l476g\\_discovery\\_qspi.h](#).

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

## Modules

Here is a list of all modules:

- **BSP**
  - **STM32L476G-DISCOVERY**
    - **STM32L476G-DISCOVERY Common**
      - Private Types Definitions
      - Private Defines
      - Private Macros
      - Exported Variables
      - Private Functions
      - Exported Functions
      - Bus Operations Functions
      - Exported Types
      - Exported Constants
        - BATTERY Detection Constants
        - LED Constants
        - BUTTON Constants
        - BUS Constants
    - **STM32L476G-DISCOVERY ACCELEROMETER**
      - Private Types
      - Private Constants
      - Private Macros
      - Private Variables
      - Private Functions
      - Exported Types
      - Exported Constants
      - Exported Macros

- Exported Functions
- **STM32L476G-DISCOVERY AUDIO**
  - Private Types
  - Private Constants
  - Private Macros
  - Private Variables
  - Exported Variables
  - Private Functions
  - Exported Types
  - Exported Constants
    - BSP Audio Out Option
    - BSP Audio Sample Rate
  - Exported Macros
  - Exported Functions
  - STM32L476G\_EVAL\_AUDIO\_Exported\_Functions
- **STM32L476G-DISCOVERY COMPASS**
  - Private Types
  - Private Constants
  - Private Macros
  - Private Variables
  - Exported Types
  - Exported Constants
  - Exported Macros
  - Exported Functions
  - Private Functions
- **STM32L476G-DISCOVERY GLASS LCD**
  - Private Constants
  - Private Variables
  - Private Functions
  - Exported Types
  - Exported Constants
  - Exported Functions
- **STM32L476G-DISCOVERY GYROSCOPE**
  - Private Types
  - Private Constants
  - Private Macros
  - Private Variables



- Private Functions
- Exported Types
- Exported Constants
- Exported Macros
- Exported Functions
- STM32L476G-DISCOVERY IDD
  - Private Defines
  - Private Variables
  - Private Functions
  - Exported Functions
  - Exported Types
    - IDD Config
  - Exported Defines
- STM32L476G-DISCOVERY QSPI
  - Private Variables
  - Private Functions
  - Exported Constants
  - Exported Types
  - Exported Functions

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Data Structures</a>	<a href="#">Data Structure Index</a>	<a href="#">Data Fields</a>		

## Data Structures

Here are the data structures with brief descriptions:

<a href="#">AUDIO_IN_TypeDef</a>	
<a href="#">AUDIO_OUT_TypeDef</a>	
<a href="#">QSPI_Info</a>	

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		

## File List

Here is a list of all files with brief descriptions:

<a href="#">stm32l476g_discovery.c</a> [code]	This file provides a functions to manage joystick of STM board (MB1184)
<a href="#">stm32l476g_discovery.h</a> [code]	This file contains definitions of STM32L476G_DISC push-buttons hardware (MB1184)
<a href="#">stm32l476g_discovery_accelerometer.c</a> [code]	This file provides a functions needed to manage ACCELEROMETER of STM32L476G-Discovery
<a href="#">stm32l476g_discovery_accelerometer.h</a> [code]	This file contains definitions of <a href="#">stm32l476g_discovery_accelerometer.c</a> firmware driver
<a href="#">stm32l476g_discovery_audio.c</a> [code]	This file provides a functions needed to manage AUDIO of STM32L476G-Discovery
<a href="#">stm32l476g_discovery_audio.h</a> [code]	This file contains the functions prototype of <a href="#">stm32l476g_discovery_audio.c</a>
<a href="#">stm32l476g_discovery_compass.c</a> [code]	This file provides a functions needed to manage COMPASS of STM32L476G-Discovery

	needed to manage (ACCELEROMETER MAGNETOMETER available on STM32L476G-Discovery board
<a href="#">stm32l476g_discovery_compass.h [code]</a>	This file contains driver for <a href="#">stm32l476g_discovery_compass</a> firmware driver
<a href="#">stm32l476g_discovery_glass_lcd.c [code]</a>	This file provides a driver for the LCD needed to manage the LCD for the STM32L476G-Discovery board
<a href="#">stm32l476g_discovery_glass_lcd.h [code]</a>	Header file for <a href="#">stm32l476g_discovery_glass_lcd</a> module
<a href="#">stm32l476g_discovery_gyroscope.c [code]</a>	This file provides a driver for the gyroscope needed to manage the gyroscope accelerometer available on the STM32L476G-Discovery board
<a href="#">stm32l476g_discovery_gyroscope.h [code]</a>	This file contains driver for <a href="#">stm32l476g_discovery_gyroscope</a> firmware driver
<a href="#">stm32l476g_discovery_idd.c [code]</a>	This file provides a driver for the I2C needed to manage the I2C functions to manage the I2C driver for STM32L476G-Discovery board
<a href="#">stm32l476g_discovery_idd.h [code]</a>	Header file for <a href="#">stm32l476g_discovery_idd</a> module
<a href="#">stm32l476g_discovery_qspi.c [code]</a>	This file includes a driver for the N25Q128A QSPI memory on the STM32L476G-Discovery board
<a href="#">stm32l476g_discovery_qspi.h [code]</a>	This file contains the functions prototype for <a href="#">stm32l476g_discovery_qspi</a> module

BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)[Directories](#)

## Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

- **Drivers**
  - **BSP**
    - **STM32L476G-Discovery**

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Modules](#)

## STM32L476G- DISCOVERY Common

[STM32L476G-DISCOVERY](#)

## Modules

### Private Types Definitions

This file provides firmware functions to manage Leds, push-buttons, COM ports, SD card on SPI and temperature sensor (TS751) available on STM32L476G-DISCOVERY discoveryuaton board from STMicroelectronics.

### Private Defines

### Private Macros

### Exported Variables

### Private Functions

### Exported Functions

### Bus Operations Functions

### Exported Types

### Exported Constants

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

Modules

## STM32L476G- DISCOVERY ACCELEROMETER

[STM32L476G-DISCOVERY](#)

## Modules

<b>Private Types</b>
<b>Private Constants</b>
<b>Private Macros</b>
<b>Private Variables</b>
<b>Private Functions</b>
<b>Exported Types</b>
<b>Exported Constants</b>
<b>Exported Macros</b>
<b>Exported Functions</b>

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Data Structures</a>	<a href="#">Data Structure Index</a>	<a href="#">Data Fields</a>		

## Data Structure Index

[A](#) | [Q](#)

**A**

[AUDIO\\_IN\\_TypeDef](#)

[AUDIO\\_OUT\\_TypeDef](#)

**Q**

[QSPI\\_Info](#)

[A](#) | [Q](#)

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Data Structures](#)

## Private Types

[STM32L476G-DISCOVERY AUDIO](#)

## Data Structures

struct	<b>AUDIO_OUT_TypeDef</b>
--------	--------------------------

struct	<b>AUDIO_IN_TypeDef</b>
--------	-------------------------

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Typedefs](#)

## Exported Types

[STM32L476G-DISCOVERY AUDIO](#)

## Typedefs

```
typedef void(* Audio_CallbackTypeDef )(void)
```

---

## Typedef Documentation

**typedef void(\* [Audio\\_CallbackTypeDef](#))(void)**

Definition at line **72** of file **[stm32l476g\\_discovery\\_audio.h](#)**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_audio.c

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery_audio.c
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file provides a set of fun
00008                  ctions needed to manage the
00009                  Audio driver for the STM32L476G
00010                  -Discovery board.
00011      *
00012      ****
00013      * @attention
00014      *
00015      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00016      microelectronics</center></h2>
00017      *
00018      * Redistribution and use in source and bin
00019      ary forms, with or without modification,
00020      * are permitted provided that the followin
00021      g conditions are met:
```

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

```

00035      *
00036      ****
*****
*****
00037      */
00038
00039 /*=====
=====
00040                                     User NOTES
00041
00042 1. How To use this driver:
00043 -----
00044      + This driver supports STM32L4xx devices
on STM32L476G-Discovery (MB1184) Discovery boards.
00045          a) to play an audio file (all functi
ons names start by BSP_AUDIO_OUT_xxx)
00046          b) to record an audio file through M
P34DT01TR, ST MEMS (all functions names start by B
SP_AUDIO_IN_xxx)
00047
00048 a) PLAY A FILE:
00049 =====
00050      + Call the function BSP_AUDIO_OUT_Init(
00051                                     OutputDe
vice: physical output mode (OUTPUT_DEVICE_SPEAKER,
00052
OUTPUT_DEVICE_HEADPHONE or OUTPUT_DEVICE_BOT
H)
00053                                     Volume
: Initial volume to be set (0 is min (mute), 1
00 is max (100%)
00054                                     AudioFre
q : Audio frequency in Hz (8000, 16000, 22500, 3
2000...)
00055
this parameter is relative to the audio file
/stream type.

```

```

00056                                     )
00057         This function configures all the hardware
are required for the audio application (codec, I2C
, SAI,
00058         GPIOs, DMA and interrupt if needed). This
function returns AUDIO_OK if configuration is
OK.
00059         If the returned value is different from
AUDIO_OK or the function is stuck then the communication
with
00060         the audio codec has failed.
00061         - OUTPUT_DEVICE_SPEAKER : only speaker
will be set as output for the audio stream.
00062         - OUTPUT_DEVICE_HEADPHONE: only headphones
will be set as output for the audio stream.
00063         - OUTPUT_DEVICE_BOTH : both Speaker
and Headphone are used as outputs for the audio
stream
00064                                     at the same
time.
00065
00066         + Call the function BSP_AUDIO_OUT_Register
Callbacks to register user callbacks
00067         required to manage audio data streaming
towards the audio codec (ErrorCallback(),
00068         HalfTransfer_CallBack() and TransferCom
plete_CallBack()).
00069
00070         + Call the function BSP_AUDIO_OUT_Play()
to start audio playback (for the first time).
00071         + Call the function BSP_AUDIO_OUT_Pause()
to pause audio playback
00072         + Call the function BSP_AUDIO_OUT_Resume(
) to resume audio playback.
00073         Note. After calling BSP_AUDIO_OUT_Pau
se() function for pause, only BSP_AUDIO_OUT_Resume
() should be called

```

00074           for resume (it is not allowed to call BSP\_AUDIO\_OUT\_Play() in this case).

00075           Note. This function should be called only when the audio file is played or paused (not stopped).

00076       + Call the function BSP\_AUDIO\_OUT\_Stop() to stop audio playback.

00077       + To modify the volume level, the sampling frequency, the device output mode,

00078           the mute status or the audio configuration or the stop, use the functions: BSP\_AUDIO\_OUT\_SetVolume(),

00079           AUDIO\_OUT\_SetFrequency(), BSP\_AUDIO\_OUT\_SetOutputMode(), BSP\_AUDIO\_OUT\_SetMute() and

00080           BSP\_AUDIO\_OUT\_ChangeAudioConfig().

00081

00082 Driver architecture:

00083 -----

00084       + This driver provides the audio layer high level API: it consists in functions

00085           exported in the stm32l476g\_discovery\_audio.h file (e.g. BSP\_AUDIO\_OUT\_Init(),

00086           BSP\_AUDIO\_OUT\_Play(), ...).

00087       + This driver also includes the Media Access Layer (MAL): it consists in

00088           functions allowing to access setup the audio devices. These functions

00089           are included as local functions into the stm32l476g\_discovery\_audio.c file

00090           (e.g. AUDIO\_SAIx\_Init()).

00091

00092 Known Limitations:

00093 -----

00094       1- Communication with the audio codec (through I2C) may be corrupted if it is interrupted by some

00095           user interrupt routines (in this case,

interrupts could be disabled just before the start of

00096 communication then re-enabled when it is over). Note that this communication is only done at

00097 the configuration phase (BSP\_AUDIO\_OUT\_Init() or BSP\_AUDIO\_OUT\_Stop()) and when Volume control modification is

00098 performed (BSP\_AUDIO\_OUT\_SetVolume() or BSP\_AUDIO\_OUT\_SetMute() or BSP\_AUDIO\_OUT\_SetOutputMode()).

00099 When the audio data is played, no communication is required with the audio codec.

00100 2- Parsing of audio file is not implemented (in order to determine audio file properties: Mono/Stereo, Data size,

00101 File size, Audio Frequency, Audio Data header size ...). The configuration is fixed for the given audio file.

00102 3- Supports only 16-bits audio data size.

00103

00104 b) RECORD A FILE:

00105 =====

00106 + Call the function BSP\_AUDIO\_IN\_Init(  
00107 AudioFrequency: Audio frequency in Hz (8000, 16000, 22500, 32000 ...)

00108 )

00109 This function configures all the hardware required for the audio application (DFSDM,

00110 GPIOs, DMA and interrupt if needed). This function returns AUDIO\_OK if the

00111 configuration completes successfully.

00112

00113 + Call the function BSP\_AUDIO\_IN\_RegisterCallbacks to register user callbacks

00114 used to stream audio data toward the re

```

cord buffer (ErrorCallback(),
00115      HalfTransfer_CallBack() and TransferCom
plete_CallBack()).
00116
00117      + Call the function BSP_AUDIO_IN_Record(
00118          pbuf Main buffer
pointer for the recorded data storing
00119          size Current siz
e of the recorded buffer
00120          )
00121      to start recording from the microphone.

00122
00123      + Call the function AUDIO_IN_STOP() to st
op recording
00124 =====
===== */
00125
00126 /* Includes -----
----- */
00127 #include <string.h>
00128 #include "stm32l476g_discovery_audio.h"
00129
00130 /** @addtogroup BSP
00131     * @{
00132     */
00133
00134 /** @addtogroup STM32L476G_DISCOVERY
00135     * @{
00136     */
00137
00138 /** @defgroup STM32L476G_DISCOVERY_AUDIO STM
32L476G-DISCOVERY AUDIO
00139     * @brief This file includes the low layer
driver for cs43l22 Audio Codec
00140     *         available on STM32L476G-Discovery
board(MB1184).

```

```

00141     * @{
00142     */
00143
00144 /* Private typedef -----
-----*/
00145 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Private_Types Private Types
00146     * @{
00147     */
00148 typedef struct
00149 {
00150     AUDIO_DrvTypeDef *      AudioDrv;
00151     /* Audio codec driver */
00152     Audio_CallbackTypeDef CbError;
00153     /* pointer to the callback function invoked when .
    .. */
00154     Audio_CallbackTypeDef CbHalfTransfer;
00155     /* pointer to the callback function invoked when .
    .. */
00156     Audio_CallbackTypeDef CbTransferComplete;
00157     /* pointer to the callback function invoked when .
    .. */
00158 } AUDIO_OUT_TypeDef;
00159
00160 typedef struct
00161 {
00162     DFSDM_Channel_HandleTypeDef hDfsdmLeftChannel; /* DFSDM channel handle used for left channel */
00163     DMA_HandleTypeDef           hDmaDfsdmLeft;
00164     /* DMA handle used for DFSDM regular conversions on left channel */
00165     int32_t *                   LeftRecBuff;
00166     /* Buffers for left samples */
00167     uint32_t                    Frequency;
00168     /* Record Frequency */
00169     uint32_t                    BitResolution;

```



```

/* Record bit resolution */
00163     uint32_t                                ChannelNbr;
/* Record Channel Number */
00164     uint16_t *                               pRecBuf;
/* Pointer to record user buffer */
00165     uint32_t                                RecSize;
/* Size to record in mono, double size to record i
n stereo */
00166     Audio_CallbackTypeDef                    CbError;
/* pointer to the callback function invoked
when a DMA transfer fails */
00167     Audio_CallbackTypeDef                    CbHalfTransfer
; /* pointer to the callback function invoked
when half of the DMA transfer is completed */
00168     Audio_CallbackTypeDef                    CbTransferComp
lete; /* pointer to the callback function invoked
when the DMA transfer is completed */
00169 } AUDIO_IN_TypeDef;
00170
00171 /**
00172  * @}
00173  */
00174
00175 /* Private defines -----
-----*/
00176 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Pri
vate_Constants Private Constants
00177  * @{
00178  */
00179 /**
00180  * @}
00181  */
00182
00183 /* Private macros -----
-----*/
00184 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Pri
vate_Macros Private Macros

```

```

00185     * @{
00186     */
00187 /*### PLAY ###*/
00188 /* SCK(kHz) = SAI_CK_x/(SAIClockDivider*2*25
00189 6) */
00189 #define SAIClockDivider(__FREQUENCY__) \
00190     (__FREQUENCY__ == AUDIO_FREQUENCY_8K
00191 ) ? 12 \
00192     : (__FREQUENCY__ == AUDIO_FREQUENCY_11
00193 K) ? 2 \
00194     : (__FREQUENCY__ == AUDIO_FREQUENCY_16
00195 K) ? 6 \
00196     : (__FREQUENCY__ == AUDIO_FREQUENCY_22
00197 K) ? 1 \
00198     : (__FREQUENCY__ == AUDIO_FREQUENCY_32
00199 K) ? 3 \
00200     : (__FREQUENCY__ == AUDIO_FREQUENCY_44
00201 K) ? 0 \
00202     : (__FREQUENCY__ == AUDIO_FREQUENCY_48
00203 K) ? 2 : 1 \
00204
00205 /*### RECORD ###*/
00206 #define DFSDMOverSampling(__FREQUENCY__) \
00207     (__FREQUENCY__ == AUDIO_FREQUENCY_8K
00208 ) ? 256 \
00209     : (__FREQUENCY__ == AUDIO_FREQUENCY_11
00210 K) ? 256 \
00211     : (__FREQUENCY__ == AUDIO_FREQUENCY_16
00212 K) ? 128 \
00213     : (__FREQUENCY__ == AUDIO_FREQUENCY_22
00214 K) ? 128 \
00215     : (__FREQUENCY__ == AUDIO_FREQUENCY_32
00216 K) ? 64 \
00217     : (__FREQUENCY__ == AUDIO_FREQUENCY_44
00218 K) ? 64 \
00219     : (__FREQUENCY__ == AUDIO_FREQUENCY_48
00220 K) ? 32 : 16 \

```

```

00207
00208 #define DFSDMClockDivider(__FREQUENCY__) \
00209     (__FREQUENCY__ == AUDIO_FREQUENCY_8K
00210 ) ? 24 \
00211     : (__FREQUENCY__ == AUDIO_FREQUENCY_11
00212 K) ? 4 \
00213     : (__FREQUENCY__ == AUDIO_FREQUENCY_16
00214 K) ? 24 \
00215     : (__FREQUENCY__ == AUDIO_FREQUENCY_22
00216 K) ? 4 \
00217     : (__FREQUENCY__ == AUDIO_FREQUENCY_32
00218 K) ? 24 \
00219     : (__FREQUENCY__ == AUDIO_FREQUENCY_44
00220 K) ? 4 \
00221     : (__FREQUENCY__ == AUDIO_FREQUENCY_48
00222 K) ? 32 : 32 \
00223
00224 #define DFSDMFilterOrder(__FREQUENCY__) \
00225     (__FREQUENCY__ == AUDIO_FREQUENCY_8K
00226 ) ? DFSDM_FILTER_SINC3_ORDER \
00227     : (__FREQUENCY__ == AUDIO_FREQUENCY_11
00228 K) ? DFSDM_FILTER_SINC3_ORDER \
00229     : (__FREQUENCY__ == AUDIO_FREQUENCY_16
00230 K) ? DFSDM_FILTER_SINC3_ORDER \
00231     : (__FREQUENCY__ == AUDIO_FREQUENCY_22
00232 K) ? DFSDM_FILTER_SINC3_ORDER \
00233     : (__FREQUENCY__ == AUDIO_FREQUENCY_32
00234 K) ? DFSDM_FILTER_SINC4_ORDER \
00235     : (__FREQUENCY__ == AUDIO_FREQUENCY_44
00236 K) ? DFSDM_FILTER_SINC4_ORDER \
00237     : (__FREQUENCY__ == AUDIO_FREQUENCY_48
00238 K) ? DFSDM_FILTER_SINC4_ORDER : DFSDM_FILTER_SINC5
00239 _ORDER \
00240
00241 #define DFSDMRightBitShift(__FREQUENCY__) \
00242     (__FREQUENCY__ == AUDIO_FREQUENCY_8K
00243 ) ? 2 \

```

```

00228      : (__FREQUENCY__ == AUDIO_FREQUENCY_11
K) ? 3 \
00229      : (__FREQUENCY__ == AUDIO_FREQUENCY_16
K) ? 2 \
00230      : (__FREQUENCY__ == AUDIO_FREQUENCY_22
K) ? 0 \
00231      : (__FREQUENCY__ == AUDIO_FREQUENCY_32
K) ? 3 \
00232      : (__FREQUENCY__ == AUDIO_FREQUENCY_44
K) ? 3 \
00233      : (__FREQUENCY__ == AUDIO_FREQUENCY_48
K) ? 7 : 0 \
00234
00235 /* Saturate the record PCM sample */
00236 #define SaturateLH(N, L, H) (((N)<(L))?(L):(((
N)>(H))?(H):(N)))
00237
00238 /**
00239  * @}
00240  */
00241
00242 /* Private variables -----
-----*/
00243 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Pri
vate_Variables Private Variables
00244  * @{
00245  */
00246 /* Audio output context information */
00247 static AUDIO_OUT_TypeDef hAudioOut;
00248
00249 /* Audio input context information */
00250 static AUDIO_IN_TypeDef hAudioIn;
00251
00252 /* SAI DMA handle */
00253 static DMA_HandleTypeDef hDmaSai;
00254 /**
00255  * @}

```

```

00256     */
00257
00258 /* Exported variables -----
-----*/
00259 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Exp
orted_Variables Exported Variables
00260     * @{
00261     */
00262 /* SAIx handle */
00263 SAI_HandleTypeDef BSP_AUDIO_hS
ai;
00264
00265 /* DFSDM filter handle */
00266 DFSDM_Filter_HandleTypeDef BSP_AUDIO_hD
fsdmLeftFilter;
00267 /**
00268     * @}
00269     */
00270
00271 /* Private function prototypes -----
-----*/
00272 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Pri
vate_Functions Private Functions
00273     * @{
00274     */
00275 static void AUDIO_CODEC_Reset(void);
00276 static uint8_t AUDIO_SAIx_Init(uint32_t Audi
oFreq);
00277 static uint8_t AUDIO_SAIx_DeInit(void);
00278 static uint8_t AUDIO_DFSDMx_Init(uint32_t Au
dioFreq);
00279 static uint8_t AUDIO_DFSDMx_DeInit(void);
00280 static uint8_t AUDIO_SAIPLLConfig(uint32_t A
udioFreq);
00281 /**
00282     * @}
00283     */

```

```

00284
00285 /* Exported functions -----
-----*/
00286 /** @addtogroup STM32L476G_DISCOVERY_AUDIO_E
xported_Functions
00287     * @{
00288     */
00289
00290 /**
00291     * @brief Configures the audio codec relat
ed peripherals.
00292     * @param OutputDevice: OUTPUT_DEVICE_SPEA
KER, OUTPUT_DEVICE_HEADPHONE,
00293     *                               or OUTPUT_DEVICE_B
OTH.
00294     * @param Volume: Initial volume level (fr
om 0 (Mute) to 100 (Max))
00295     * @param AudioFreq: Audio frequency used
to play the audio stream.ion.
00296     * @retval BSP AUDIO status
00297     * @note The SAI PLL input clock must be
configure in the user application.
00298     * The SAI PLL configuration done w
ithin this function assumes that
00299     * the SAI PLL input clock runs at
8 MHz.
00300     */
00301 uint8_t BSP_AUDIO_OUT_Init(uint16_t OutputDe
vice,
00302                             uint8_t Volume,
00303                             uint32_t AudioFre
q)
00304 {
00305     /* Initialize the audio output context */
00306     hAudioOut.AudioDrv = &cs43l22_dr
v;
00307     hAudioOut.CbError = (Audio_Call

```

```

backTypeDef)NULL;
00308     hAudioOut.CbHalfTransfer      = (Audio_Call
backTypeDef)NULL;
00309     hAudioOut.CbTransferComplete = (Audio_Call
backTypeDef)NULL;
00310
00311     /* Configure the SAI PLL according to the
requested audio frequency */
00312     if (AUDIO_SAIPLLConfig(AudioFreq) != AUDIO
_OK)
00313     {
00314         return AUDIO_ERROR;
00315     }
00316
00317     /* SAI data transfer preparation: prepare
the Media to be used for the audio
00318         transfer from memory to SAI peripheral.
*/
00319     if (AUDIO_SAIx_Init(AudioFreq) != AUDIO_OK
)
00320     {
00321         return AUDIO_ERROR;
00322     }
00323
00324     /* Retrieve audio codec identifier */
00325     if (cs43l22_drv.ReadID(AUDIO_I2C_ADDRESS)
!= CS43L22_ID)
00326     {
00327         return AUDIO_ERROR;
00328     }
00329
00330     /* Reset the audio codec Registers */
00331     AUDIO_CODEC_Reset();
00332
00333     /* Initialize the audio codec internal reg
isters */
00334     if (hAudioOut.AudioDrv->Init(AUDIO_I2C_ADD

```

```

RESS,
00335                                     OutputDevice,

00336                                     Volume,
00337                                     AudioFreq) !=
    0)
00338     {
00339         return AUDIO_ERROR;
00340     }
00341
00342     /* Set the requested volume */
00343     BSP_AUDIO_OUT_SetVolume(Volume);
00344
00345     return AUDIO_OK;
00346 }
00347
00348 /**
00349  * @brief De-Initializes audio codec relat
00350  * @retval BSP AUDIO status
00351  */
00352
00353 uint8_t BSP_AUDIO_OUT_DeInit(void)
00354 {
00355     /* De-initializes the Audio Codec audio in
00356     terface */
00357     if (AUDIO_SAIx_DeInit() != AUDIO_OK)
00358     {
00359         return AUDIO_ERROR;
00360     }
00361
00362     /* DeInit Audio component interface */
00363     hAudioOut.AudioDrv->DeInit();
00364
00365     /* Reset the audio output context */
00366     memset(&hAudioOut, 0, sizeof(hAudioOut));
00367

```



```

00367     return AUDIO_OK;
00368 }
00369
00370 /**
00371  * @brief Starts playing audio stream from
00372  * a data buffer for a determined size.
00373  * @param pData: pointer on PCM samples bu
00374  * ffer
00375  * @param Size: Number of audio data BYTES.
00376
00377  * @retval BSP AUDIO status
00378  */
00379 uint8_t BSP_AUDIO_OUT_Play(uint16_t* pData,
00380 uint32_t Size)
00381 {
00382     /* Call the audio Codec Play function */
00383     if (hAudioOut.AudioDrv->Play(AUDIO_I2C_ADD
00384 RESS, pData, Size) != 0)
00385     {
00386         return AUDIO_ERROR;
00387     }
00388
00389     /* Initiate a DMA transfer of PCM samples
00390     towards the serial audio interface */
00391     if (HAL_SAI_Transmit_DMA(&BSP_AUDIO_hSai,
00392 (uint8_t *)pData, DMA_MAX(Size))!= HAL_OK)
00393     {
00394         return AUDIO_ERROR;
00395     }
00396
00397     return AUDIO_OK;
00398 }
00399
00400 /**
00401  * @brief Sends n-Bytes on the SAI interfa
00402  * ce.
00403  * @param pData: pointer on PCM samples bu

```

```

ffer
00396     * @param  Size: number of data to be written
00397     * @retval BSP_AUDIO status
00398     */
00399 uint8_t BSP_AUDIO_OUT_ChangeBuffer(uint16_t
*pData, uint16_t Size)
00400 {
00401     /* Initiate a DMA transfer of PCM samples
towards the serial audio interface */
00402     if (HAL_SAI_Transmit_DMA(&BSP_AUDIO_hSai,
(uint8_t *)pData, Size) != HAL_OK)
00403     {
00404         return AUDIO_ERROR;
00405     }
00406
00407     return AUDIO_OK;
00408 }
00409
00410 /**
00411     * @brief This function Pauses the audio f
ile stream. In case
00412     *         of using DMA, the DMA Pause feat
ure is used.
00413     * @note When calling BSP_AUDIO_OUT_Pause()
function for pause, only
00414     *         BSP_AUDIO_OUT_Resume() function sh
ould be called for resume
00415     *         (use of BSP_AUDIO_OUT_Play() funct
ion for resume could lead
00416     *         to unexpected behavior).
00417     * @retval BSP_AUDIO status
00418     */
00419 uint8_t BSP_AUDIO_OUT_Pause(void)
00420 {
00421     /* Call the Audio Codec Pause function */
00422     if (hAudioOut.AudioDrv->Pause(AUDIO_I2C_AD

```

```

DDRESS) != 0)
00423     {
00424         return AUDIO_ERROR;
00425     }
00426
00427     /* Pause DMA transfer of PCM samples towards the serial audio interface */
00428     if (HAL_SAI_DMAPause(&BSP_AUDIO_hSai) != HAL_OK)
00429     {
00430         return AUDIO_ERROR;
00431     }
00432
00433     return AUDIO_OK;
00434 }
00435
00436 /**
00437  * @brief This function Resumes the audio file stream.
00438  * @note When calling BSP_AUDIO_OUT_Pause() function for pause, only
00439  *        BSP_AUDIO_OUT_Resume() function should be called for resume
00440  *        (use of BSP_AUDIO_OUT_Play() function for resume could lead to
00441  *        unexpected behavior).
00442  * @retval BSP AUDIO status
00443  */
00444 uint8_t BSP_AUDIO_OUT_Resume(void)
00445 {
00446     /* Call the Audio Codec Resume function */
00447     if (hAudioOut.AudioDrv->Resume(AUDIO_I2C_ADDRESS) != 0)
00448     {
00449         return AUDIO_ERROR;
00450     }
00451

```

```

00452  /* Resume DMA transfer of PCM samples towards the serial audio interface */
00453  if (HAL_SAI_DMAResume(&BSP_AUDIO_hSai) != HAL_OK)
00454  {
00455      return AUDIO_ERROR;
00456  }
00457
00458  return AUDIO_OK;
00459 }
00460
00461 /**
00462  * @brief Stops audio playing and Power down the Audio Codec.
00463  * @param Option: could be one of the following parameters
00464  *             - CODEC_PDWN_SW: for software power off (by writing registers).
00465  *             Then no need to reconfigure the Codec after power on.
00466  *             - CODEC_PDWN_HW: completely shut down the codec (physically).
00467  *             Then need to reconfigure the Codec after power on.
00468  * @retval BSP AUDIO status
00469  */
00470 uint8_t BSP_AUDIO_OUT_Stop(uint32_t Option)
00471 {
00472     /* Stop DMA transfer of PCM samples towards the serial audio interface */
00473     if (HAL_SAI_DMAStop(&BSP_AUDIO_hSai) != HAL_OK)
00474     {
00475         return AUDIO_ERROR;
00476     }
00477
00478     /* Call Audio Codec Stop function */

```

```

00479     if (hAudioOut.AudioDrv->Stop(AUDIO_I2C_ADD
RESS, Option) != 0)
00480     {
00481         return AUDIO_ERROR;
00482     }
00483
00484     if(Option == CODEC_PDWN_HW)
00485     {
00486         /* Wait at least 100us */
00487         HAL_Delay(1);
00488     }
00489
00490     return AUDIO_OK;
00491 }
00492
00493 /**
00494  * @brief Controls the current audio volum
e level.
00495  * @param Volume: Volume level to be set i
n percentage from 0% to 100% (0 for
00496  *             Mute and 100 for Max volume leve
l).
00497  * @retval BSP AUDIO status
00498  */
00499 uint8_t BSP_AUDIO_OUT_SetVolume(uint8_t Volu
me)
00500 {
00501     /* Call the codec volume control function
with converted volume value */
00502     if (hAudioOut.AudioDrv->SetVolume(AUDIO_I2
C_ADDRESS, Volume) != 0)
00503     {
00504         return AUDIO_ERROR;
00505     }
00506
00507     return AUDIO_OK;
00508 }

```

```

00509
00510 /**
00511  * @brief Enables or disables the MUTE mode by software
00512  * @param Cmd: Could be AUDIO_MUTE_ON to mute sound or AUDIO_MUTE_OFF to
00513  *             unmute the codec and restore previous volume level.
00514  * @retval BSP_AUDIO status
00515  */
00516 uint8_t BSP_AUDIO_OUT_SetMute(uint32_t Cmd)
00517 {
00518     /* Call the Codec Mute function */
00519     if (hAudioOut.AudioDrv->SetMute(AUDIO_I2C_ADDRESS, Cmd) != 0)
00520     {
00521         return AUDIO_ERROR;
00522     }
00523
00524     return AUDIO_OK;
00525 }
00526
00527 /**
00528  * @brief Switch dynamically (while audio file is being played) the output
00529  *             target (speaker or headphone).
00530  * @param Output: The audio output target:
00531  *             OUTPUT_DEVICE_SPEAKER,
00532  *             OUTPUT_DEVICE_HEADPHONE or OUTPUT_DEVICE_BOTH
00533  * @retval BSP_AUDIO status
00534  */
00535 uint8_t BSP_AUDIO_OUT_SetOutputMode(uint8_t Output)
00536 {
00537     /* Call the Codec output device function */

```

```

00537     if (hAudioOut.AudioDrv->SetOutputMode(AUDI
0_I2C_ADDRESS, Output) != 0)
00538     {
00539         return AUDIO_ERROR;
00540     }
00541
00542     return AUDIO_OK;
00543 }
00544
00545 /**
00546  * @brief Updates the audio frequency.
00547  * @param AudioFreq: Audio frequency used
00548  * to play the audio stream.
00549  * @note The SAI PLL input clock must be
00550  * configure in the user application.
00551  * The SAI PLL configuration done w
00552  * ithin this function assumes that
00553  * the SAI PLL input clock runs at
00554  * 8 MHz.
00555  * @retval BSP AUDIO status
00556  */
00557 uint8_t BSP_AUDIO_OUT_SetFrequency(uint32_t
00558 AudioFreq)
00559 {
00560     /* Configure the SAI PLL according to the
00561     requested audio frequency */
00562     if (AUDIO_SAIPLLConfig(AudioFreq) != AUDIO
00563     _OK)
00564     {
00565         return AUDIO_ERROR;
00566     }
00567
00568     /* Disable SAI peripheral to allow access
00569     to SAI internal registers */
00570     __HAL_SAI_DISABLE(&BSP_AUDIO_hSai);
00571
00572     /* Update the SAI audio frequency configur

```

```

ation */
00565     BSP_AUDIO_hSai.Init.Mckdiv = SAIClockDivid
er(AudioFreq);
00566     HAL_SAI_Init(&BSP_AUDIO_hSai);
00567
00568     /* Enable SAI peripheral to generate MCLK
*/
00569     __HAL_SAI_ENABLE(&BSP_AUDIO_hSai);
00570
00571     return AUDIO_OK;
00572 }
00573
00574 /**
00575  * @brief  Changes the Audio Out Configurati
ion.
00576  * @param  AudioOutOption: specifies the au
dio out new configuration
00577  *          This parameter can be any value
of @ref BSP_Audio_Out_Option
00578  * @note   This API should be called after
the BSP_AUDIO_OUT_Init() to adjust the
00579  *          audio out configuration.
00580  * @retval None
00581  */
00582 void BSP_AUDIO_OUT_ChangeAudioConfig(uint32_
t AudioOutOption)
00583 {
00584     /****** Playback Buffer circular/norma
l mode *****/
00585     if(AudioOutOption & BSP_AUDIO_OUT_CIRCULAR
MODE)
00586     {
00587         /* Deinitialize the Stream to update DMA
mode */
00588         HAL_DMA_DeInit(BSP_AUDIO_hSai.hdmatx);
00589
00590         /* Update the SAI audio Transfer DMA mod

```



```

e */
00591     BSP_AUDIO_hSai.hdmatx->Init.Mode = DMA_C
IRCULAR;
00592
00593     /* Configure the DMA Stream with new Tra
nsfer DMA mode */
00594     HAL_DMA_Init(BSP_AUDIO_hSai.hdmatx);

00595 }
00596 else /* BSP_AUDIO_OUT_NORMALMODE */
00597 {
00598     /* Deinitialize the Stream to update DMA
mode */
00599     HAL_DMA_DeInit(BSP_AUDIO_hSai.hdmatx);
00600
00601     /* Update the SAI audio Transfer DMA mod
e */
00602     BSP_AUDIO_hSai.hdmatx->Init.Mode = DMA_N
ORMAL;
00603
00604     /* Configure the DMA Stream with new Tra
nsfer DMA mode */
00605     HAL_DMA_Init(BSP_AUDIO_hSai.hdmatx);

00606 }
00607
00608     /****** Playback Buffer stereo/mono mo
de *****/
00609     if(AudioOutOption & BSP_AUDIO_OUT_STEREO
MODE)
00610     {
00611         /* Disable SAI peripheral to allow acces
s to SAI internal registers */
00612         __HAL_SAI_DISABLE(&BSP_AUDIO_hSai);
00613
00614         /* Update the SAI audio frame slot confi
guration */

```

```

00615     BSP_AUDIO_hSai.Init.MonoStereoMode = SAI
_STEREOMODE;
00616     HAL_SAI_Init(&BSP_AUDIO_hSai);
00617
00618     /* Enable SAI peripheral to generate MCLK */
00619     __HAL_SAI_ENABLE(&BSP_AUDIO_hSai);
00620 }
00621 else /* BSP_AUDIO_OUT_MONOMODE */
00622 {
00623     /* Disable SAI peripheral to allow access to SAI internal registers */
00624     __HAL_SAI_DISABLE(&BSP_AUDIO_hSai);
00625
00626     /* Update the SAI audio frame slot configuration */
00627     BSP_AUDIO_hSai.Init.MonoStereoMode = SAI_MONOMODE;
00628     HAL_SAI_Init(&BSP_AUDIO_hSai);
00629
00630     /* Enable SAI peripheral to generate MCLK */
00631     __HAL_SAI_ENABLE(&BSP_AUDIO_hSai);
00632 }
00633 }
00634
00635 /**
00636  * @brief register user callback functions
00637  * @param ErrorCallback: pointer to the error callback function
00638  * @param HalfTransferCallback: pointer to the half transfer callback function
00639  * @param TransferCompleteCallback: pointer to the transfer complete callback function
00640  * @retval None
00641  */

```

```

00642 void BSP_AUDIO_OUT_RegisterCallbacks(Audio_C
allbackTypeDef ErrorCallback,
00643                                     Audio_C
allbackTypeDef HalfTransferCallback,
00644                                     Audio_C
allbackTypeDef TransferCompleteCallback)
00645 {
00646     hAudioOut.CbError                = ErrorCallba
ck;
00647     hAudioOut.CbHalfTransfer          = HalfTransfe
rCallback;
00648     hAudioOut.CbTransferComplete = TransferCom
pleteCallback;
00649 }
00650
00651 /**
00652  * @brief Tx Transfer completed callbacks.
00653  * @param hsai: SAI handle
00654  * @retval None
00655  */
00656 void HAL_SAI_TxCpltCallback(SAI_HandleTypeDe
f *hsai)
00657 {
00658     /* Invoke the registered 'TransferComplete
' function (if any) */
00659     if (hAudioOut.CbTransferComplete != (Audio
_CallbackTypeDef)NULL)
00660     {
00661         hAudioOut.CbTransferComplete();
00662     }
00663 }
00664
00665 /**
00666  * @brief Tx Half Transfer completed callb
acks.
00667  * @param hsai: SAI handle
00668  * @retval None

```

```

00669     */
00670 void HAL_SAI_TxHalfCpltCallback(SAI_HandleType
peDef *hsai)
00671 {
00672     /* Invoke the registered 'HalfTransfer' ca
llback function (if any) */
00673     if (hAudioOut.CbHalfTransfer != (Audio_Cal
lbackTypeDef)NULL)
00674     {
00675         hAudioOut.CbHalfTransfer();
00676     }
00677 }
00678
00679 /**
00680  * @brief SAI error callbacks.
00681  * @param hsai: SAI handle
00682  * @retval None
00683  */
00684 void HAL_SAI_ErrorCallback(SAI_HandleTypeDef
*hsai)
00685 {
00686     /* Invoke the registered 'ErrorCallback' c
allback function (if any) */
00687     if (hAudioOut.CbError != (Audio_CallbackTy
peDef)NULL)
00688     {
00689         hAudioOut.CbError();
00690     }
00691 }
00692
00693 /**
00694  * @}
00695  */
00696
00697 /** @addtogroup STM32L476G_EVAL_AUDIO_Export
ed_Functions
00698  * @{

```

```

00699     */
00700
00701 /**
00702  * @brief  Initializes micropone related pe
ripherals.
00703  * @note   This function assumes that the S
AI input clock (through PLL_M)
00704  *         is already configured and ready
to be used.
00705  * @param  AudioFreq: Audio frequency to be
configured for the SAI peripheral.
00706  * @param  BitRes: Audio frequency to be co
nfigured for the SAI peripheral.
00707  * @param  ChnlNbr: Audio frequency to be c
onfigured for the SAI peripheral.
00708  * @retval BSP AUDIO status
00709  */
00710 uint8_t BSP_AUDIO_IN_Init(uint32_t AudioFreq
, uint32_t BitRes, uint32_t ChnlNbr)
00711 {
00712     /* Update the audio input context */
00713     hAudioIn.Frequency           = AudioFreq;
00714     hAudioIn.BitResolution       = BitRes;
00715     hAudioIn.ChannelNbr         = ChnlNbr;
00716     hAudioIn.CbError            = (Audio_Callb
ackTypeDef)NULL;
00717     hAudioIn.CbHalfTransfer      = (Audio_Callb
ackTypeDef)NULL;
00718     hAudioIn.CbTransferComplete = (Audio_Callb
ackTypeDef)NULL;
00719
00720     /* Configure the SAI PLL according to the
requested audio frequency */
00721     if (AUDIO_SAIPLLConfig(AudioFreq) != AUDIO
_OK)
00722     {
00723         return AUDIO_ERROR;

```

```

00724     }
00725
00726     /* Initializes the Digital Filter for Sigma-Delta Modulators interface */
00727     if(AUDIO_DFSDMx_Init(AudioFreq) != AUDIO_OK
)
00728     {
00729         return AUDIO_ERROR;
00730     }
00731
00732     return AUDIO_OK;
00733 }
00734
00735 /**
00736  * @brief De-Initializes microphone related peripherals.
00737  * @retval BSP AUDIO status
00738
00739  */
00740 uint8_t BSP_AUDIO_IN_DeInit(void)
00741 {
00742     /* De-initializes the Digital Filter for Sigma-Delta Modulators interface */
00743     if (AUDIO_DFSDMx_DeInit() != AUDIO_OK)
00744     {
00745         return AUDIO_ERROR;
00746     }
00747
00748     /* Reset the audio input context */
00749     memset(&hAudioIn, 0, sizeof(hAudioIn));
00750
00751     return AUDIO_OK;
00752 }
00753
00754 /**
00755  * @brief Starts audio recording.
00756  * @param pbuf: Main buffer pointer for the

```

```

e recorded data storing
00757  * @param size: Current size of the record
ed buffer
00758  * @note The Right channel is start at fi
rst with synchro on start of Left channel
00759  * @retval BSP AUDIO status
00760  */
00761 uint8_t BSP_AUDIO_IN_Record(uint16_t* pbuf,
uint32_t size)
00762 {
00763     hAudioIn.pRecBuf = pbuf;
00764     hAudioIn.RecSize = size;
00765
00766     /* Allocate hAudioIn.LeftRecBuff buffer */
00767     #if defined(BSP_AUDIO_USE_RTOS)
00768     hAudioIn.LeftRecBuff = (int32_t *)k_mallo
c(size * sizeof(int32_t));
00769     #else
00770     hAudioIn.LeftRecBuff = (int32_t *)malloc(
size * sizeof(int32_t));
00771     #endif
00772     if(hAudioIn.LeftRecBuff == NULL)
00773     {
00774         return AUDIO_ERROR;
00775     }
00776
00777     /* Call the Media layer start function for
left channel */
00778     if(HAL_DFSDM_FilterRegularStart_DMA(&BSP_A
UDIO_hDfsdmLeftFilter,
00779                                         (int32
_t*)hAudioIn.LeftRecBuff,
00780                                         (hAudi
oIn.RecSize/DEFAULT_AUDIO_IN_CHANNEL_NBR)) != HAL_
OK)
00781     {
00782         return AUDIO_ERROR;

```

```

00783     }
00784
00785     return AUDIO_OK;
00786 }
00787
00788 /**
00789  * @brief Updates the audio frequency.
00790  * @param AudioFreq: Audio frequency used
00791  * to record the audio stream.
00792  * @note This API should be called after
00793  * the BSP_AUDIO_IN_Init() to adjust the
00794  * audio frequency.
00795  * @retval BSP AUDIO status
00796  */
00797 uint8_t BSP_AUDIO_IN_SetFrequency(uint32_t A
00798 udioFreq)
00799 {
00800     /* Configure the SAI PLL according to the
00801     requested audio frequency */
00802     if (AUDIO_SAIPLLConfig(AudioFreq) != AUDIO
00803     _OK)
00804     {
00805         return AUDIO_ERROR;
00806     }
00807
00808     /* De-initializes the Digital Filter for S
00809     igma-Delta Modulators interface */
00810     if(AUDIO_DFSDMx_DeInit() != AUDIO_OK)
00811     {
00812         return AUDIO_ERROR;
00813     }
00814
00815     /* Initializes the Digital Filter for Sigm
00816     a-Delta Modulators interface */
00817     if(AUDIO_DFSDMx_Init(AudioFreq) != AUDIO_OK
00818     )
00819     {

```



```

00812     return AUDIO_ERROR;
00813 }
00814
00815     return AUDIO_OK;
00816 }
00817
00818 /**
00819  * @brief Regular conversion complete call
back.
00820  * @note In interrupt mode, user has to r
ead conversion value in this function
00821         using HAL_DFSDM_FilterGetRegular
Value.
00822  * @param hdfsdm_filter : DFSDM filter han
dle.
00823  * @retval None
00824  */
00825 void HAL_DFSDM_FilterRegConvCpltCallback(DFS
DM_Filter_HandleTypeDef *hdfsdm_filter)
00826 {
00827     uint32_t index;
00828     uint32_t recbufsize = (hAudioIn.RecSize/DE
FAULT_AUDIO_IN_CHANNEL_NBR);
00829
00830     for(index = (recbufsize/2); index < recbuf
size; index++)
00831     {
00832         hAudioIn.pRecBuf[index] = (uint16_t)(Sat
uraLH((hAudioIn.LeftRecBuff[index] >> 8), -32760,
32760));
00833     }
00834
00835     /* Invoke the registered 'TransferComplete
' function (if any) */
00836     if (hAudioIn.CbTransferComplete != (Audio_
CallbackTypeDef)NULL)
00837     {

```

```

00838     hAudioIn.CbTransferComplete();
00839 }
00840 }
00841
00842 /**
00843  * @brief Half regular conversion complete
00844  * @param hdfsdm_filter : DFSDM filter handle.
00845  * @retval None
00846  */
00847 void HAL_DFSDM_FilterRegConvHalfCpltCallback
(DFSDM_Filter_HandleTypeDef *hdfsdm_filter)
00848 {
00849     uint32_t index;
00850     uint32_t recbufsize = (hAudioIn.RecSize/DE
FAULT_AUDIO_IN_CHANNEL_NBR);
00851
00852
00853     for(index = 0; index < (recbufsize/2); index++)
00854     {
00855         hAudioIn.pRecBuf[index] = (uint16_t)(Satura
LH((hAudioIn.LeftRecBuff[index] >> 8), -32760,
32760));
00856     }
00857
00858     /* Invoke the registered 'HalfTransfer' ca
llback function (if any) */
00859     if (hAudioIn.CbHalfTransfer != (Audio_Call
backTypeDef)NULL)
00860     {
00861         hAudioIn.CbHalfTransfer();
00862     }
00863 }
00864
00865 /**

```

```

00866     * @brief Error callback.
00867     * @param hdfsdm_filter : DFSDM filter han
dle.
00868     * @retval None
00869     */
00870 void HAL_DFSDM_FilterErrorCallback(DFSDM_Fil
ter_HandleTypeDef *hdfsdm_filter)
00871 {
00872     /* Invoke the registered 'ErrorCallback' c
allback function (if any) */
00873     if (hAudioIn.CbError != (Audio_CallbackTyp
eDef)NULL)
00874     {
00875         hAudioIn.CbError();
00876     }
00877 }
00878
00879 /**
00880     * @brief Stops audio recording.
00881     * @retval BSP_AUDIO status
00882     */
00883 uint8_t BSP_AUDIO_IN_Stop(void)
00884 {
00885     /* Call the Media layer stop function for
left channel */
00886     if(HAL_DFSDM_FilterRegularStop_DMA(&BSP_AU
DIO_hDfsdmLeftFilter) != HAL_OK )
00887     {
00888         return AUDIO_ERROR;
00889     }
00890
00891     /* Free hAudioIn.LeftRecBuff buffer */
00892     #if defined(BSP_AUDIO_USE_RTOS)
00893     k_free((void *)hAudioIn.LeftRecBuff);
00894     #else
00895     free((void *)hAudioIn.LeftRecBuff);
00896     #endif

```

```

00897
00898     return AUDIO_OK;
00899 }
00900
00901 /**
00902  * @brief Pauses the audio file stream.
00903  * @retval BSP AUDIO status
00904  */
00905 uint8_t BSP_AUDIO_IN_Pause(void)
00906 {
00907     /* Call the Media layer stop function */
00908     if(HAL_DFSDM_FilterRegularStop_DMA(&BSP_AUDIO_hDfsdmLeftFilter) != HAL_OK)
00909     {
00910         return AUDIO_ERROR;
00911     }
00912
00913     return AUDIO_OK;
00914 }
00915
00916 /**
00917  * @brief Resumes the audio file stream.
00918  * @retval BSP AUDIO status
00919  */
00920 uint8_t BSP_AUDIO_IN_Resume(void)
00921 {
00922     /* Call the Media layer start function for
00923      left channel */
00923     if(HAL_DFSDM_FilterRegularStart_DMA(&BSP_AUDIO_hDfsdmLeftFilter,
00924                                         (int32_t*)hAudioIn.LeftRecBuff,
00925                                         (hAudioIn.RecSize/DEFAULT_AUDIO_IN_CHANNEL_NBR)) != HAL_OK)
00926     {
00927         return AUDIO_ERROR;

```

```

00928     }
00929
00930     return AUDIO_OK;
00931 }
00932
00933 /**
00934  * @brief register user callback functions
00935  * @param errorCallback: pointer to the error callback function
00936  * @param HalfTransferCallback: pointer to the half transfer callback function
00937  * @param TransferCompleteCallback: pointer to the transfer complete callback function
00938  * @retval None
00939  */
00940 void BSP_AUDIO_IN_RegisterCallbacks(Audio_CallbackTypeDef errorCallback,
00941                                     Audio_CallbackTypeDef HalfTransferCallback,
00942                                     Audio_CallbackTypeDef TransferCompleteCallback)
00943 {
00944     hAudioIn.CbError = errorCallback;
00945     hAudioIn.CbHalfTransfer = HalfTransferCallback;
00946     hAudioIn.CbTransferComplete = TransferCompleteCallback;
00947 }
00948 /**
00949  * @}
00950  */
00951
00952 /* private functions -----
-----*/
00953 /** @addtogroup STM32L476G_DISCOVERY_AUDIO_P

```

```

private_Functions
00954     * @{
00955     */
00956 /**
00957     * @brief Initializes the Audio Codec audio
00958     * @param AudioFreq: Audio frequency to be
00959     * @note The default SlotActive configura
00960     * and user can update this configu
00961     * @retval BSP_AUDIO status
00962     */
00963 static uint8_t AUDIO_SAIx_Init(uint32_t Audi
00964 {
00965     /* Disable SAI peripheral to allow access
00966     __HAL_SAI_DISABLE(&BSP_AUDIO_hSai);
00967
00968     /* Initialize the BSP_AUDIO_hSai Instance
00969     parameter */
00970     BSP_AUDIO_hSai.Instance = AUDIO_SAIx;
00971
00972     /* Configure SAI_Block_x
00973     LSBFirst: Disabled
00974     DataSize: 16 */
00975     BSP_AUDIO_hSai.Init.AudioMode          = SAI_M
00976     ODEMASTER_TX;
00977     BSP_AUDIO_hSai.Init.Synchro             = SAI_A
00978     SYNCHRONOUS;
00979     BSP_AUDIO_hSai.Init.SynchroExt          = SAI_S
00980     YNCEXT_DISABLE;
00981     BSP_AUDIO_hSai.Init.OutputDrive         = SAI_0
00982     UTPUTDRIVE_ENABLE;
00983     BSP_AUDIO_hSai.Init.NoDivider           = SAI_M

```

```

ASTERDIVIDER_ENABLE;
00979  BSP_AUDIO_hSai.Init.FIFOThreshold  = SAI_F
IF0THRESHOLD_1QF;
00980  BSP_AUDIO_hSai.Init.AudioFrequency = SAI_A
UDIO_FREQUENCY_MCKDIV;
00981  BSP_AUDIO_hSai.Init.Mckdiv        = SAICl
ockDivider(AudioFreq);
00982  BSP_AUDIO_hSai.Init.MonoStereoMode = SAI_S
TEREOMODE;
00983  BSP_AUDIO_hSai.Init.CompandingMode = SAI_N
OCOMPANDING;
00984  BSP_AUDIO_hSai.Init.TriState      = SAI_O
UTPUT_NOTRELEASED;
00985  BSP_AUDIO_hSai.Init.Protocol      = SAI_F
REE_PROTOCOL;
00986  BSP_AUDIO_hSai.Init.DataSize      = SAI_D
ATASIZE_16;
00987  BSP_AUDIO_hSai.Init.FirstBit      = SAI_F
IRSTBIT_MSB;
00988  BSP_AUDIO_hSai.Init.ClockStrobing = SAI_C
LOCKSTROBING_RISINGEDGE;
00989
00990  /* Configure SAI_Block_x Frame
00991  Frame Length: 32
00992  Frame active Length: 16
00993  FS Definition: Start frame + Channel Side
identification
00994  FS Polarity: FS active Low
00995  FS Offset: FS asserted one bit before the
first bit of slot 0 */
00996  BSP_AUDIO_hSai.FrameInit.FrameLength = 32;
00997  BSP_AUDIO_hSai.FrameInit.ActiveFrameLength
    = 16;
00998  BSP_AUDIO_hSai.FrameInit.FSDefinition = SA
I_FS_CHANNEL_IDENTIFICATION;
00999  BSP_AUDIO_hSai.FrameInit.FSPolarity = SAI_
FS_ACTIVE_LOW;

```

```

01000     BSP_AUDIO_hSai.FrameInit.FSOffset = SAI_FS
_BEFOREFIRSTBIT;
01001
01002     /* Configure SAI Block_x Slot
01003     Slot First Bit Offset: 0
01004     Slot Size : 16
01005     Slot Number: 2
01006     Slot Active: Slots 0 and 1 actives */
01007     BSP_AUDIO_hSai.SlotInit.FirstBitOffset = 0
;
01008     BSP_AUDIO_hSai.SlotInit.SlotSize = SAI_SLO
TSIZE_DATASIZE;
01009     BSP_AUDIO_hSai.SlotInit.SlotNumber = 2;
01010     BSP_AUDIO_hSai.SlotInit.SlotActive = SAI_S
LOTACTIVE_0 | SAI_SLOTACTIVE_1;
01011
01012     /* Initializes the SAI peripheral*/
01013     if (HAL_SAI_Init(&BSP_AUDIO_hSai) != HAL_O
K)
01014     {
01015         return AUDIO_ERROR;
01016     }
01017
01018     /* Enable SAI peripheral to generate MCLK
*/
01019     __HAL_SAI_ENABLE(&BSP_AUDIO_hSai);
01020
01021     return AUDIO_OK;
01022
01023 }
01024
01025 /**
01026  * @brief De-initializes the Audio Codec a
udio interface (SAI).
01027  * @retval BSP AUDIO status
01028  */
01029 static uint8_t AUDIO_SAIx_DeInit(void)

```



```

01030 {
01031     /* Disable the SAI audio block */
01032     __HAL_SAI_DISABLE(&BSP_AUDIO_hSai);
01033
01034     /* De-initializes the SAI peripheral */
01035     if (HAL_SAI_DeInit(&BSP_AUDIO_hSai) != HAL
_OK)
01036     {
01037         return AUDIO_ERROR;
01038     }
01039
01040     /* Disable SAIx PLL */
01041     if (AUDIO_SAIx_PLL_DISABLE() != AUDIO_OK)
01042     {
01043         return AUDIO_ERROR;
01044     }
01045
01046     return AUDIO_OK;
01047 }
01048
01049 /**
01050  * @brief SAI MSP Init
01051  * @param hsai : pointer to a SAI_HandleTy
01052  * @retval None
01053  */
01054 void HAL_SAI_MspInit(SAI_HandleTypeDef *hsai
)
01055 {
01056     GPIO_InitTypeDef GPIO_InitStruct;
01057
01058     /* Enable SAI clock */
01059     AUDIO_SAIx_CLK_ENABLE();
01060
01061     /* Enable GPIO clock */
01062     AUDIO_SAIx_MCK_SCK_SD_FS_ENABLE();
01063

```

```

01064  /* CODEC_SAI pins configuration: FS, SCK,
MCK and SD pins -----*/
01065  GPIO_InitStruct.Pin = AUDIO_SAIx_FS_PIN |
AUDIO_SAIx_SCK_PIN | AUDIO_SAIx_SD_PIN | AUDIO_SAI
x_MCK_PIN;
01066  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
01067  GPIO_InitStruct.Pull = GPIO_NOPULL;
01068  GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
01069  GPIO_InitStruct.Alternate = AUDIO_SAIx_MCK
_SCK_SD_FS_AF;
01070  HAL_GPIO_Init(AUDIO_SAIx_MCK_SCK_SD_FS_GPI
O_PORT, &GPIO_InitStruct);
01071
01072  /* Enable the DMA clock */
01073  AUDIO_SAIx_DMAX_CLK_ENABLE();
01074
01075  if(hsai->Instance == AUDIO_SAIx)
01076  {
01077      /* Configure the hDmaSai handle paramete
rs */
01078      hDmaSai.Init.Request                      = DMA_R
EQUEST_1;
01079      hDmaSai.Init.Direction                    = DMA_M
EMORY_TO_PERIPH;
01080      hDmaSai.Init.PeriphInc                    = DMA_P
INC_DISABLE;
01081      hDmaSai.Init.MemInc                      = DMA_M
INC_ENABLE;
01082      hDmaSai.Init.PeriphDataAlignment = AUDIO
_SAIx_DMAX_PERIPH_DATA_SIZE;
01083      hDmaSai.Init.MemDataAlignment            = AUDIO
_SAIx_DMAX_MEM_DATA_SIZE;
01084      hDmaSai.Init.Mode                        = DMA_N
ORMAL;
01085      hDmaSai.Init.Priority                    = DMA_P
RIORITY_HIGH;
01086

```

```

01087     hDmaSai.Instance = AUDIO_SAIx_DMAX_CHANN
EL;
01088
01089     /* Associate the DMA handle */
01090     __HAL_LINKDMA(hsai, hdmatx, hDmaSai);
01091
01092     /* Deinitialize the Stream for new trans
fer */
01093     HAL_DMA_DeInit(&hDmaSai);
01094
01095     /* Configure the DMA Stream */
01096     HAL_DMA_Init(&hDmaSai);
01097 }
01098
01099     /* SAI DMA IRQ Channel configuration */
01100     HAL_NVIC_SetPriority(AUDIO_SAIx_DMAX_IRQ,
AUDIO_OUT_IRQ_PREPRIO, 0);
01101     HAL_NVIC_EnableIRQ(AUDIO_SAIx_DMAX_IRQ);
01102 }
01103
01104 /**
01105  * @brief SAI MSP De-init
01106  * @param hsai : pointer to a SAI_HandleTy
peDef structure
01107  * @retval None
01108  */
01109 void HAL_SAI_MspDeInit(SAI_HandleTypeDef *hs
ai)
01110 {
01111     /* Disable SAI DMA Channel IRQ */
01112     HAL_NVIC_DisableIRQ(AUDIO_SAIx_DMAX_IRQ);
01113
01114     /* Reset the DMA Stream configuration*/
01115     HAL_DMA_DeInit(&hDmaSai);
01116
01117     /* Disable the DMA clock */
01118     AUDIO_SAIx_DMAX_CLK_DISABLE();

```

```

01119
01120     /* De-initialize FS, SCK, MCK and SD pins*/

01121     HAL_GPIO_DeInit(AUDIO_SAIx_MCK_SCK_SD_FS_G
PIO_PORT,
01122                     AUDIO_SAIx_FS_PIN | AUDIO_
SAIx_SCK_PIN | AUDIO_SAIx_SD_PIN | AUDIO_SAIx_MCK_
PIN);
01123
01124     /* Disable GPIO clock */
01125     AUDIO_SAIx_MCK_SCK_SD_FS_DISABLE();
01126
01127     /* Disable SAI clock */
01128     AUDIO_SAIx_CLK_DISABLE();
01129 }
01130
01131 /**
01132  * @brief Resets the audio codec. It resto
res the default configuration of the
01133  *         codec (this function shall be ca
lled before initializing the codec).
01134  * @retval None
01135  */
01136 static void AUDIO_CODEC_Reset(void)
01137 {
01138     /* Initialize the audio driver structure */

01139     hAudioOut.AudioDrv = &cs43l22_drv;
01140
01141     hAudioOut.AudioDrv->Reset(AUDIO_I2C_ADDRESS
);
01142 }
01143
01144 /**
01145  * @}
01146  */
01147

```

```

01148 /** @addtogroup STM32L476G_DISCOVERY_AUDIO_P
private_Functions
01149     * @{
01150     */
01151
01152 /**
01153     * @brief Initializes the Digital Filter f
or Sigma-Delta Modulators interface (DFSDM).
01154     * @param AudioFreq: Audio frequency to be
used to set correctly the DFSDM peripheral.
01155     * @retval BSP AUDIO status
01156     */
01157 static uint8_t AUDIO_DFSDMx_Init(uint32_t Au
dioFreq)
01158 {
01159     /*####CHANNEL 2####*/
01160     hAudioIn.hDfsdmLeftChannel.Init.OutputCloc
k.Activation = ENABLE;
01161     hAudioIn.hDfsdmLeftChannel.Init.OutputCloc
k.Selection = DFSDM_CHANNEL_OUTPUT_CLOCK_AUDIO;
01162     /* Set the DFSDM clock OUT audio frequency
configuration */
01163     hAudioIn.hDfsdmLeftChannel.Init.OutputCloc
k.Divider = DFSDMClockDivider(AudioFreq);
01164     hAudioIn.hDfsdmLeftChannel.Init.Input.Mult
iplexer = DFSDM_CHANNEL_EXTERNAL_INPUTS;
01165     hAudioIn.hDfsdmLeftChannel.Init.Input.Data
Packing = DFSDM_CHANNEL_STANDARD_MODE;
01166     hAudioIn.hDfsdmLeftChannel.Init.Input.Pins
= DFSDM_CHANNEL_SAME_CHANNEL_PINS;
01167     /* Request to sample stable data for LEFT
micro on Rising edge */
01168     hAudioIn.hDfsdmLeftChannel.Init.SerialInte
rface.Type = DFSDM_CHANNEL_SPI_RISING;
01169     hAudioIn.hDfsdmLeftChannel.Init.SerialInte
rface.SpiClock = DFSDM_CHANNEL_SPI_CLOCK_INTERNAL;
01170     hAudioIn.hDfsdmLeftChannel.Init.Awd.Filter

```

```

Order                = DFSDM_CHANNEL_SINC1_ORDER;
01171  hAudioIn.hDfsdmLeftChannel.Init.Awd.Oversa
mpling                = 10;
01172  hAudioIn.hDfsdmLeftChannel.Init.Offset
                = 0;
01173  hAudioIn.hDfsdmLeftChannel.Init.RightBitSh
ift                = DFSDMRightBitShift(AudioFreq);
01174
01175  hAudioIn.hDfsdmLeftChannel.Instance
                = DFSDM_Channel2;

01176
01177      /* Init the DFSDM Channel */
01178  if (HAL_DFSDM_ChannelInit(&hAudioIn.hDfsdm
LeftChannel) != HAL_OK)
01179  {
01180      return AUDIO_ERROR;
01181  }
01182
01183      /*#####FILTER 0#####*/
01184  BSP_AUDIO_hDfsdmLeftFilter.Init.RegularPar
am.Trigger          = DFSDM_FILTER_SW_TRIGGER;
01185  BSP_AUDIO_hDfsdmLeftFilter.Init.RegularPar
am.FastMode         = ENABLE;
01186  BSP_AUDIO_hDfsdmLeftFilter.Init.RegularPar
am.DmaMode          = ENABLE;
01187  BSP_AUDIO_hDfsdmLeftFilter.Init.InjectedPa
ram.Trigger         = DFSDM_FILTER_SW_TRIGGER;
01188  BSP_AUDIO_hDfsdmLeftFilter.Init.InjectedPa
ram.ScanMode        = DISABLE;
01189  BSP_AUDIO_hDfsdmLeftFilter.Init.InjectedPa
ram.DmaMode         = DISABLE;
01190  BSP_AUDIO_hDfsdmLeftFilter.Init.InjectedPa
ram.ExtTrigger      = DFSDM_FILTER_EXT_TRIG_TIM8_TR
GO;
01191  BSP_AUDIO_hDfsdmLeftFilter.Init.InjectedPa
ram.ExtTriggerEdge  = DFSDM_FILTER_EXT_TRIG_BOTH_ED
GES;

```

```

01192     BSP_AUDIO_hDfsdmLeftFilter.Init.FilterPara
m.SincOrder          = DFSDMFilterOrder(AudioFreq);
01193     /* Set the DFSDM Filters Oversampling to h
ave correct sample rate */
01194     BSP_AUDIO_hDfsdmLeftFilter.Init.FilterPara
m.Oversampling       = DFSDMOversampling(AudioFreq);
01195     BSP_AUDIO_hDfsdmLeftFilter.Init.FilterPara
m.IntOversampling    = 1;
01196
01197     BSP_AUDIO_hDfsdmLeftFilter.Instance
                    = AUDIO_DFSDMx_LEFT_FILTER;

01198
01199     /* Init the DFSDM Filter */
01200     if (HAL_DFSDM_FilterInit(&BSP_AUDIO_hDfsdm
LeftFilter) != HAL_OK)
01201     {
01202         return AUDIO_ERROR;
01203     }
01204
01205     /* Configure regular channel */
01206     if (HAL_DFSDM_FilterConfigRegChannel(&BSP_
AUDIO_hDfsdmLeftFilter,
01207                                         DFSDM_
CHANNEL_2,
01208                                         DFSDM_
CONTINUOUS_CONV_ON) != HAL_OK)
01209     {
01210         return AUDIO_ERROR;
01211     }
01212
01213     return AUDIO_OK;
01214 }
01215
01216 /**
01217  * @brief De-initializes the Digital Filte
r for Sigma-Delta Modulators interface (DFSDM).
01218  * @retval BSP AUDIO status

```

```

01219  */
01220 static uint8_t AUDIO_DFSDMx_DeInit(void)
01221 {
01222     /* De-initializes the DFSDM filters to all
01223     ow access to DFSDM internal registers */
01223     if (HAL_DFSDM_FilterDeInit(&BSP_AUDIO_hDfs
01224 dmLeftFilter) != HAL_OK)
01224     {
01225         return AUDIO_ERROR;
01226     }
01227
01228     /* De-initializes the DFSDM channels to al
01229 low access to DFSDM internal registers */
01229     if (HAL_DFSDM_ChannelDeInit(&hAudioIn.hDfs
01230 dmLeftChannel) != HAL_OK)
01230     {
01231         return AUDIO_ERROR;
01232     }
01233
01234     /* Disable DFSDM clock */
01235     AUDIO_DFSDMx_CLK_DISABLE();
01236
01237     /* Disable SAIx PLL */
01238     if (AUDIO_SAIx_PLL_DISABLE() != AUDIO_OK)
01239     {
01240         return AUDIO_ERROR;
01241     }
01242
01243     /* DFSDM reset */
01244     __HAL_RCC_DFSDM_FORCE_RESET();
01245     __HAL_RCC_DFSDM_RELEASE_RESET();
01246
01247     return AUDIO_OK;
01248 }
01249
01250 /**
01251  * @brief Initializes the DFSDM channel MS

```



```

P.
01252     * @param  hdfsdm_channel : DFSDM channel h
andle.
01253     * @retval None
01254     */
01255 void HAL_DFSDM_ChannelMspInit(DFSDM_Channel_
HandleTypeDef *hdfsdm_channel)
01256 {
01257     GPIO_InitTypeDef  GPIO_InitStruct;
01258
01259     /* Enable DFSDM clock */
01260     AUDIO_DFSDMx_CLK_ENABLE();
01261
01262     /* Enable GPIO clock */
01263     AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_CLK_ENA
BLE();
01264
01265     /* DFSDM pins configuration: DFSDM_CKOUT,
DMIC_DATIN pins -----*/
01266     GPIO_InitStruct.Pin = AUDIO_DFSDMx_CKOUT_P
IN | AUDIO_DFSDMx_DMIC_DATIN_PIN;
01267     GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
01268     GPIO_InitStruct.Pull = GPIO_NOPULL;
01269     GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
01270     GPIO_InitStruct.Alternate = AUDIO_DFSDMx_C
KOUT_DMIC_DATIN_AF;
01271     HAL_GPIO_Init(AUDIO_DFSDMx_CKOUT_DMIC_DATI
N_GPIO_PORT, &GPIO_InitStruct);
01272 }
01273
01274 /**
01275     * @brief  De-initializes the DFSDM channel
MSP.
01276     * @param  hdfsdm_channel : DFSDM channel h
andle.
01277     * @retval None
01278     */

```

```

01279 void HAL_DFSDM_ChannelMspDeInit(DFSDM_Channel_
01280     HandleTypeDef *hdfsdm_channel)
01281 {
01282     GPIO_InitTypeDef  GPIO_InitStructure;
01283     /* Enable GPIO clock */
01284     AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_CLK_ENA
01285     BLE();
01286     /* DFSDM pins configuration: DFSDM_CKOUT */

01287     GPIO_InitStructure.Pin = AUDIO_DFSDMx_CKOUT_P
01288     IN;
01289     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP
01290     ;
01291     GPIO_InitStructure.Pull = GPIO_NOPULL;
01292     GPIO_InitStructure.Speed = GPIO_SPEED_LOW;
01293     HAL_GPIO_Init(AUDIO_DFSDMx_CKOUT_DMIC_DATI
01294     N_GPIO_PORT, &GPIO_InitStructure);
01295     HAL_GPIO_WritePin(AUDIO_DFSDMx_CKOUT_DMIC_
01296     DATIN_GPIO_PORT, AUDIO_DFSDMx_CKOUT_PIN, GPIO_PIN_
01297     RESET);
01298
01299     /* De-initialize DMIC_DATIN pin */
01300     HAL_GPIO_DeInit(AUDIO_DFSDMx_CKOUT_DMIC_DA
01301     TIN_GPIO_PORT, AUDIO_DFSDMx_DMIC_DATIN_PIN);
01302 }
01303
01304 /**
01305  * @brief  Initializes the DFSDM filter MSP.
01306  *
01307  * @param  hdfsdm_filter : DFSDM filter han
01308  * dle.
01309  *
01310  * @retval None
01311  */
01312 void HAL_DFSDM_FilterMspInit(DFSDM_Filter_Ha

```

```

ndleTypeDef *hdfsdm_filter)
01305 {
01306     /* Enable DFSDM clock */
01307     AUDIO_DFSDMx_CLK_ENABLE();
01308
01309     /* Enable the DMA clock */
01310     AUDIO_DFSDMx_DMAX_CLK_ENABLE();
01311
01312     /* Configure the hAudioIn.hDmaDfsdmLeft ha
ndle parameters */
01313     hAudioIn.hDmaDfsdmLeft.Init.Request
        = DMA_REQUEST_0;
01314     hAudioIn.hDmaDfsdmLeft.Init.Direction
        = DMA_PERIPH_TO_MEMORY;
01315     hAudioIn.hDmaDfsdmLeft.Init.PeriphInc
        = DMA_PINC_DISABLE;
01316     hAudioIn.hDmaDfsdmLeft.Init.MemInc
        = DMA_MINC_ENABLE;
01317     hAudioIn.hDmaDfsdmLeft.Init.PeriphDataAlig
nment = AUDIO_DFSDMx_DMAX_PERIPH_DATA_SIZE;
01318     hAudioIn.hDmaDfsdmLeft.Init.MemDataAlignme
nt    = AUDIO_DFSDMx_DMAX_MEM_DATA_SIZE;
01319     hAudioIn.hDmaDfsdmLeft.Init.Mode
        = DMA_CIRCULAR;
01320     hAudioIn.hDmaDfsdmLeft.Init.Priority
        = DMA_PRIORITY_HIGH;
01321
01322     hAudioIn.hDmaDfsdmLeft.Instance
        = AUDIO_DFSDMx_DMAX_LEFT_CHANNEL;
01323
01324     /* Associate the DMA handle */
01325     __HAL_LINKDMA(hdfsdm_filter, hdmaReg, hAud
ioIn.hDmaDfsdmLeft);
01326
01327     /* Reset DMA handle state */
01328     __HAL_DMA_RESET_HANDLE_STATE(&hAudioIn.hDm
aDfsdmLeft);

```

```

01329
01330     /* Configure the DMA Channel */
01331     HAL_DMA_Init(&hAudioIn.hDmaDfsdmLeft);

01332
01333     /* DMA IRQ Channel configuration */
01334     HAL_NVIC_SetPriority(AUDIO_DFSDMx_DMax_LEFT_IRQ, AUDIO_OUT_IRQ_PREPRIO, 0);
01335     HAL_NVIC_EnableIRQ(AUDIO_DFSDMx_DMax_LEFT_IRQ);
01336 }
01337
01338 /**
01339  * @brief De-initializes the DFSDM filter
01340  * @param hdfsdm_filter : DFSDM filter handle.
01341  * @retval None
01342  */
01343 void HAL_DFSDM_FilterMspDeInit(DFSDM_Filter_HandleTypeDef *hdfsdm_filter)
01344 {
01345     /* Disable DMA Channel IRQ */
01346     HAL_NVIC_DisableIRQ(AUDIO_DFSDMx_DMax_LEFT_IRQ);
01347
01348     /* De-initialize the DMA Channel */
01349     HAL_DMA_DeInit(&hAudioIn.hDmaDfsdmLeft);

01350
01351     /* Disable the DMA clock */
01352     AUDIO_DFSDMx_DMax_CLK_DISABLE();
01353 }
01354
01355 /**
01356  * @brief Configures the SAI PLL clock according to the required audio frequency.

```

```

01357     * @param Frequency: Audio frequency.
01358     * @retval BSP AUDIO status
01359     * @note The SAI PLL input clock must be
configured in the user application.
01360     * The SAI PLL configuration done w
ithin this function assumes that
01361     * the SAI PLL input clock runs at
8 MHz.
01362     */
01363 static uint8_t AUDIO_SAIPLLConfig(uint32_t F
requency)
01364 {
01365     RCC_PeriphCLKInitTypeDef RCC_ExCLKInitStru
ct;
01366
01367     /* Retreive actual RCC configuration */
01368     HAL_RCCEx_GetPeriphCLKConfig(&RCC_ExCLKIni
tStruct);
01369
01370     if ( (Frequency == AUDIO_FREQUENCY_11K
)
01371         || (Frequency == AUDIO_FREQUENCY_22K
)
01372         || (Frequency == AUDIO_FREQUENCY_44K
) ) )
01373     {
01374         /* Configure PLLSAI prescalers */
01375         /* SAI clock config
01376         PLLSAI1_VCO= 8 Mhz * PLLSAI1N = 8 * 24 =
VCO_192M
01377         SAI_CK_x = PLLSAI1_VCO/PLLSAI1P = 192/17
= 11.294 Mhz */
01378         RCC_ExCLKInitStruct.PeriphClockSelection
= RCC_PERIPHCLK_SAI1;
01379         RCC_ExCLKInitStruct.PLLSAI1.PLLSAI1N
= 24;
01380         RCC_ExCLKInitStruct.PLLSAI1.PLLSAI1P

```

```

    = 17;
01381     RCC_ExCLKInitStruct.PLLSAI1.PLLSAI1Clock
Out = RCC_PLLSAI1_SAI1CLK;
01382     RCC_ExCLKInitStruct.Sai1ClockSelection
    = RCC_SAI1CLKSOURCE_PLLSAI1;
01383 }
01384     else /* AUDIO_FREQUENCY_8K, AUDIO_FREQUENC
Y_16K, AUDIO_FREQUENCY_48K, AUDIO_FREQUENCY_96K */
01385     {
01386         /* SAI clock config
01387         PLLSAI1_VCO= 8 Mhz * PLLSAI1N = 8 * 43 =
VCO_344M
01388         SAI_CK_x = PLLSAI1_VCO/PLLSAI1P = 344/7
= 49.142 Mhz */
01389         RCC_ExCLKInitStruct.PeriphClockSelection
    = RCC_PERIPHCLK_SAI1;
01390         RCC_ExCLKInitStruct.PLLSAI1.PLLSAI1N
    = 43;
01391         RCC_ExCLKInitStruct.PLLSAI1.PLLSAI1P
    = 7;
01392         RCC_ExCLKInitStruct.PLLSAI1.PLLSAI1Clock
Out = RCC_PLLSAI1_SAI1CLK;
01393         RCC_ExCLKInitStruct.Sai1ClockSelection
    = RCC_SAI1CLKSOURCE_PLLSAI1;
01394     }
01395
01396     if (HAL_RCCEx_PeriphCLKConfig(&RCC_ExCLKIn
itStruct) != HAL_OK)
01397     {
01398         return AUDIO_ERROR;
01399     }
01400
01401     return AUDIO_OK;
01402 }
01403
01404 /**
01405     * @}

```

```
01406    */
01407
01408  /*
01409    * @}
01410    */
01411
01412  /*
01413    * @}
01414    */
01415
01416  /*
01417    * @}
01418    */
01419
01420  /*
01421    * @}
01422    */
01423
01424  /***** (C) COPYRIGHT STMicroelectronics *****/
01425  *****END OF FILE*****/
```

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

Modules

## STM32L476G- DISCOVERY AUDIO

[STM32L476G-DISCOVERY](#)

This file includes the low layer driver for cs43l22 Audio Codec available on STM32L476G-Discovery board(MB1184). [More...](#)



## Modules

Private Types
Private Constants
Private Macros
Private Variables
Exported Variables
Private Functions
Exported Types
Exported Constants
Exported Macros
Exported Functions
STM32L476G_EVAL_AUDIO_Exported_Functions

## Detailed Description

This file includes the low layer driver for cs43l22 Audio Codec available on STM32L476G-Discovery board(MB1184).

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Modules](#)

## STM32L476G- DISCOVERY COMPASS

[STM32L476G-DISCOVERY](#)

## Modules

<b>Private Types</b>
<b>Private Constants</b>
<b>Private Macros</b>
<b>Private Variables</b>
<b>Exported Types</b>
<b>Exported Constants</b>
<b>Exported Macros</b>
<b>Exported Functions</b>
<b>Private Functions</b>

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Modules](#)

## STM32L476G- DISCOVERY GYROSCOPE

[STM32L476G-DISCOVERY](#)

## Modules

<b>Private Types</b>
<b>Private Constants</b>
<b>Private Macros</b>
<b>Private Variables</b>
<b>Private Functions</b>
<b>Exported Types</b>
<b>Exported Constants</b>
<b>Exported Macros</b>
<b>Exported Functions</b>

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Modules](#)

## STM32L476G- DISCOVERY IDD

[STM32L476G-DISCOVERY](#)

This file includes the Idd driver for STM32L476G-DISCOVERY board.  
[More...](#)

# Modules

Private Defines
Private Variables
Private Functions
Exported Functions
Exported Types
Exported Defines



## Detailed Description

This file includes the Idd driver for STM32L476G-DISCOVERY board.

It allows user to measure MCU Idd current on board, especially in different low power modes.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Data Structures](#)

## Exported Types

[STM32L476G-DISCOVERY QSPI](#)

## Data Structures

---

```
struct QSPI_Info
```

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_qspi.h

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery_qspi.h
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file contains the common d
00008      *             efines and functions prototypes for
00009      *             the stm32l476g_discovery_qspi.c
00010      *             driver.
00011      *             ****
00012      * @attention
00013      *
00014      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00015      * icroelectronics</center></h2>
00016      *
00017      * Redistribution and use in source and bin
00018      * ary forms, with or without modification,
00019      * are permitted provided that the followin
00020      * g conditions are met:
```

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

```

00035      *
00036      ****
00037      */
00038
00039 /* Define to prevent recursive inclusion ---
-----*/
00040 #ifndef __STM32L476G_DISCOVERY_QSPI_H
00041 #define __STM32L476G_DISCOVERY_QSPI_H
00042
00043 #ifdef __cplusplus
00044     extern "C" {
00045 #endif
00046
00047 /* Includes -----
-----*/
00048 #include "stm32l4xx_hal.h"
00049 #include "../Components/n25q128a/n25q128a.h"
00050
00051 /** @addtogroup BSP
00052     * @{
00053     */
00054
00055 /** @addtogroup STM32L476G_DISCOVERY
00056     * @{
00057     */
00058
00059 /** @addtogroup STM32L476G_DISCOVERY_QSPI
00060     * @{
00061     */
00062
00063 /* Exported constants -----
-----*/
00064 /** @defgroup STM32L476G_DISCOVERY_QSPI_Exported_Constants Exported Constants
00065     * @{
00066     */

```

```

00067 /* QSPI Error codes */
00068 #define QSPI_OK ((uint8_t)0x00)
00069 #define QSPI_ERROR ((uint8_t)0x01)
00070 #define QSPI_BUSY ((uint8_t)0x02)
00071 #define QSPI_NOT_SUPPORTED ((uint8_t)0x04)
00072 #define QSPI_SUSPENDED ((uint8_t)0x08)
00073
00074 /**
00075  * @}
00076  */
00077
00078 /* Exported types -----
-----*/
00079 /** @defgroup STM32L476G_DISCOVERY_QSPI_Exported_Types Exported Types
00080  * @{
00081  */
00082 /* QSPI Info */
00083 typedef struct {
00084     uint32_t FlashSize; /*!< Size of
the flash */
00085     uint32_t EraseSectorSize; /*!< Size of
sectors for the erase operation */
00086     uint32_t EraseSectorsNumber; /*!< Number o
f sectors for the erase operation */
00087     uint32_t ProgPageSize; /*!< Size of
pages for the program operation */
00088     uint32_t ProgPagesNumber; /*!< Number o
f pages for the program operation */
00089 } QSPI_Info;
00090
00091 /**
00092  * @}
00093  */
00094
00095 /* Exported functions -----
-----*/

```

```

00096 /** @defgroup STM32L476G_DISCOVERY_QSPI_Exported_Functions Exported Functions
00097     * @{
00098     */
00099 uint8_t BSP_QSPI_Init          (void);
00100 uint8_t BSP_QSPI_DeInit        (void);
00101 uint8_t BSP_QSPI_Read          (uint8_t* pData
, uint32_t ReadAddr, uint32_t Size);
00102 uint8_t BSP_QSPI_Write        (uint8_t* pData
, uint32_t WriteAddr, uint32_t Size);
00103 uint8_t BSP_QSPI_Erase_Block  (uint32_t Block
Address);
00104 uint8_t BSP_QSPI_Erase_Sector(uint32_t Sector);
00105 uint8_t BSP_QSPI_Erase_Chip   (void);
00106 uint8_t BSP_QSPI_GetStatus     (void);
00107 uint8_t BSP_QSPI_GetInfo       (QSPI_Info* pInfo);
00108 uint8_t BSP_QSPI_EnableMemoryMappedMode(void);
00109 uint8_t BSP_QSPI_SuspendErase(void);
00110 uint8_t BSP_QSPI_ResumeErase  (void);
00111
00112 /**
00113     * @}
00114     */
00115
00116 /**
00117     * @}
00118     */
00119
00120 /**
00121     * @}
00122     */
00123
00124 /**
00125     * @}

```



```
00126    */
00127
00128 #ifdef __cplusplus
00129 }
00130 #endif
00131
00132 #endif /* __STM32L476G_DISCOVERY_QSPI_H */
00133
00134 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE*****/
```

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by [doxygen](#) 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_qspi.c

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery_qspi.c
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file includes a standard d
river for the N25Q128A QSPI
00008      *          memory mounted on STM32L476G-Di
scovery board.
00009      @verbatim
00010      =====
00011      ##### How to use this d
river #####
00012      =====
00013      [...]
00014      (#) This driver is used to drive the N25Q
128A QSPI external
00015      memory mounted on STM32L476G-DISCO ev
```

```

evaluation board.
00016
00017      (#) This driver need a specific component
      driver (N25Q128A) to be included with.
00018
00019      (#) Initialization steps:
00020          (++) Initialize the QPSI external mem
ory using the BSP_QSPI_Init() function. This
00021          function includes the MSP layer
hardware resources initialization and the
00022          QSPI interface with the external
      memory.
00023
00024      (#) QSPI memory operations
00025          (++) QSPI memory can be accessed with
      read/write operations once it is
00026          initialized.
00027          Read/write operation can be perf
ormed with AHB access using the functions
00028          BSP_QSPI_Read()/BSP_QSPI_Write()
      .
00029          (++) The function BSP_QSPI_GetInfo()
returns the configuration of the QSPI memory.
00030          (see the QSPI memory data sheet)
00031          (++) Perform erase block operation us
ing the function BSP_QSPI_Erase_Block() and by
00032          specifying the block address. Yo
u can perform an erase operation of the whole
00033          chip by calling the function BSP
_QSPI_Erase_Chip().
00034          (++) The function BSP_QSPI_GetStatus(
) returns the current status of the QSPI memory.
00035          (see the QSPI memory data sheet)
00036      @endverbatim
00037      *****
      *****
00038      * @attention

```

00039       \*

00040       \*   <h2><center>&copy; COPYRIGHT(c) 2015 STM  
icroelectronics</center></h2>

00041       \*

00042       \*   Redistribution and use in source and bin  
ary forms, with or without modification,

00043       \*   are permitted provided that the followin  
g conditions are met:

00044       \*       1. Redistributions of source code must  
      retain the above copyright notice,

00045       \*       this list of conditions and the fol  
lowing disclaimer.

00046       \*       2. Redistributions in binary form must  
      reproduce the above copyright notice,

00047       \*       this list of conditions and the fol  
lowing disclaimer in the documentation

00048       \*       and/or other materials provided wit  
h the distribution.

00049       \*       3. Neither the name of STMicroelectron  
ics nor the names of its contributors

00050       \*       may be used to endorse or promote p  
roducts derived from this software

00051       \*       without specific prior written perm  
ission.

00052       \*

00053       \*   THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"

00054       \*   AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE

00055       \*   IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE

00056       \*   DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE

00057       \*   FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL

00058       \*   DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR

```

00059      * SERVICES; LOSS OF USE, DATA, OR PROFITS;
      OR BUSINESS INTERRUPTION) HOWEVER
00060      * CAUSED AND ON ANY THEORY OF LIABILITY, W
      HETHER IN CONTRACT, STRICT LIABILITY,
00061      * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI
      SE) ARISING IN ANY WAY OUT OF THE USE
00062      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
      POSSIBILITY OF SUCH DAMAGE.
00063      *
00064      *****
      *****
00065      */
00066
00067 /* Includes -----
      -----*/
00068 #include "stm32l476g_discovery_qspi.h"
00069
00070 /** @addtogroup BSP
00071     * @{
00072     */
00073
00074 /** @addtogroup STM32L476G_DISCOVERY
00075     * @{
00076     */
00077
00078 /** @defgroup STM32L476G_DISCOVERY_QSPI STM3
      2L476G-DISCOVERY QSPI
00079     * @{
00080     */
00081
00082 /* Private variables -----
      -----*/
00083
00084 /** @defgroup STM32L476G_DISCOVERY_QSPI_Priv
      ate_Variables Private Variables
00085     * @{
00086     */

```

```

00087 QSPI_HandleTypeDef QSPIHandle;
00088
00089 /**
00090  * @}
00091  */
00092
00093
00094 /* Private functions -----
-----*/
00095
00096 /** @defgroup STM32L476G_DISCOVERY_QSPI_Priv
ate_Functions Private Functions
00097  * @{
00098  */
00099 static void      QSPI_MspInit          (void
);
00100 static void      QSPI_MspDeInit        (void
);
00101 static uint8_t   QSPI_ResetMemory      (QSPI
_HandleTypeDef *hqspi);
00102 static uint8_t   QSPI_DummyCyclesCfg   (QSPI
_HandleTypeDef *hqspi);
00103 static uint8_t   QSPI_WriteEnable      (QSPI
_HandleTypeDef *hqspi);
00104 static uint8_t   QSPI_AutoPollingMemReady(QSPI
_HandleTypeDef *hqspi, uint32_t Timeout);
00105
00106 /**
00107  * @}
00108  */
00109
00110 /* Exported functions -----
-----*/
00111
00112 /** @addtogroup STM32L476G_DISCOVERY_QSPI_Ex
ported_Functions
00113  * @{

```

```

00114    */
00115
00116 /**
00117  * @brief Initializes the QSPI interface.
00118  * @retval QSPI memory status
00119  */
00120 uint8_t BSP_QSPI_Init(void)
00121 {
00122     QSPIHandle.Instance = QUADSPI;
00123
00124     /* Call the DeInit function to reset the d
river */
00125     if (HAL_QSPI_DeInit(&QSPIHandle) != HAL_OK
)
00126     {
00127         return QSPI_ERROR;
00128     }
00129
00130     /* System level initialization */
00131     QSPI_MspInit();
00132
00133     /* QSPI initialization */
00134     QSPIHandle.Init.ClockPrescaler      = 0; /*
Clock = Fhb = 80 MHz */
00135     QSPIHandle.Init.FifoThreshold       = 4;
00136     QSPIHandle.Init.SampleShifting     = QSPI_
SAMPLE_SHIFTING_NONE;
00137     QSPIHandle.Init.FlashSize           = POSIT
ION_VAL(N25Q128A_FLASH_SIZE) - 1;
00138     QSPIHandle.Init.ChipSelectHighTime = QSPI_
CS_HIGH_TIME_1_CYCLE;
00139     QSPIHandle.Init.ClockMode          = QSPI_
CLOCK_MODE_0;
00140
00141     if (HAL_QSPI_Init(&QSPIHandle) != HAL_OK)
00142     {
00143         return QSPI_ERROR;

```

```

00144     }
00145
00146     /* QSPI memory reset */
00147     if (QSPI_ResetMemory(&QSPISHandle) != QSPI_
00148 OK)
00149     {
00149         return QSPI_NOT_SUPPORTED;
00150     }
00151
00152     /* Configuration of the dummy cycles on QS
00153 PI memory side */
00153     if (QSPI_DummyCyclesCfg(&QSPISHandle) != QS
00154 PI_OK)
00154     {
00155         return QSPI_NOT_SUPPORTED;
00156     }
00157
00158     return QSPI_OK;
00159 }
00160
00161 /**
00162  * @brief De-Initializes the QSPI interfac
00163 e.
00164  * @retval QSPI memory status
00165 */
00165 uint8_t BSP_QSPI_DeInit(void)
00166 {
00167     QSPISHandle.Instance = QUADSPI;
00168
00169     /* Call the DeInit function to reset the d
00170 river */
00170     if (HAL_QSPI_DeInit(&QSPISHandle) != HAL_OK
00171 )
00171     {
00172         return QSPI_ERROR;
00173     }
00174

```



```

00175     /* System level De-initialization */
00176     QSPI_MspDeInit();
00177
00178     return QSPI_OK;
00179 }
00180
00181 /**
00182  * @brief Reads an amount of data from the
00183  *        QSPI memory.
00184  * @param pData: Pointer to data to be read
00185  * @param ReadAddr: Read start address
00186  * @param Size: Size of data to read
00187  * @retval QSPI memory status
00188 */
00188 uint8_t BSP_QSPI_Read(uint8_t* pData, uint32_t ReadAddr, uint32_t Size)
00189 {
00190     QSPI_CommandTypeDef sCommand;
00191
00192     /* Initialize the read command */
00193     sCommand.InstructionMode = QSPI_INSTRUCTION_1_LINE;
00194     sCommand.Instruction = QUAD_INOUT_FAST_READ_CMD;
00195     sCommand.AddressMode = QSPI_ADDRESS_4_LINES;
00196     sCommand.AddressSize = QSPI_ADDRESS_24_BITS;
00197     sCommand.Address = ReadAddr;
00198     sCommand.AlternateByteMode = QSPI_ALTERNATE_BYTES_NONE;
00199     sCommand.DataMode = QSPI_DATA_4_LINES;
00200     sCommand.DummyCycles = N25Q128A_DUMMY_CYCLES_READ_QUAD;
00201     sCommand.NbData = Size;

```

```

00202     sCommand.DdrMode                = QSPI_DDR_MODE
_DISABLE;
00203     sCommand.DdrHoldHalfCycle         = QSPI_DDR_HHC_
ANALOG_DELAY;
00204     sCommand.SIOOMode                 = QSPI_SIOO_INS
T_EVERY_CMD;
00205
00206     /* Configure the command */
00207     if (HAL_QSPI_Command(&QSPISHandle, &sComman
d, HAL_QSPI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00208     {
00209         return QSPI_ERROR;
00210     }
00211
00212     /* Reception of the data */
00213     if (HAL_QSPI_Receive(&QSPISHandle, pData, H
AL_QSPI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00214     {
00215         return QSPI_ERROR;
00216     }
00217
00218     return QSPI_OK;
00219 }
00220
00221 /**
00222  * @brief Writes an amount of data to the
QSPI memory.
00223  * @param pData: Pointer to data to be wri
tten
00224  * @param WriteAddr: Write start address
00225  * @param Size: Size of data to write
00226  * @retval QSPI memory status
00227  */
00228 uint8_t BSP_QSPI_Write(uint8_t* pData, uint3
2_t WriteAddr, uint32_t Size)
00229 {
00230     QSPI_CommandTypeDef sCommand;

```

```

00231     uint32_t end_addr, current_size, current_a
ddr;
00232
00233     /* Calculation of the size between the wri
te address and the end of the page */
00234     current_addr = 0;
00235
00236     while (current_addr <= WriteAddr)
00237     {
00238         current_addr += N25Q128A_PAGE_SIZE;
00239     }
00240     current_size = current_addr - WriteAddr;
00241
00242     /* Check if the size of the data is less t
han the remaining place in the page */
00243     if (current_size > Size)
00244     {
00245         current_size = Size;
00246     }
00247
00248     /* Initialize the adress variables */
00249     current_addr = WriteAddr;
00250     end_addr = WriteAddr + Size;
00251
00252     /* Initialize the program command */
00253     sCommand.InstructionMode    = QSPI_INSTRUCT
ION_1_LINE;
00254     sCommand.Instruction        = EXT_QUAD_IN_F
AST_PROG_CMD;
00255     sCommand.AddressMode        = QSPI_ADDRESS_
4_LINES;
00256     sCommand.AddressSize        = QSPI_ADDRESS_
24_BITS;
00257     sCommand.AlternateByteMode  = QSPI_ALTERNAT
E_BYTES_NONE;
00258     sCommand.DataMode           = QSPI_DATA_4_L
INES;

```

```

00259     sCommand.DummyCycles          = 0;
00260     sCommand.DdrMode                 = QSPI_DDR_MODE
_DISABLE;
00261     sCommand.DdrHoldHalfCycle        = QSPI_DDR_HHC_
ANALOG_DELAY;
00262     sCommand.SIOOMode                = QSPI_SIOO_INS
T_EVERY_CMD;
00263
00264     /* Perform the write page by page */
00265     do
00266     {
00267         sCommand.Address = current_addr;
00268         sCommand.NbData  = current_size;
00269
00270         /* Enable write operations */
00271         if (QSPI_WriteEnable(&QSPIHandle) != QSP
I_OK)
00272         {
00273             return QSPI_ERROR;
00274         }
00275
00276         /* Configure the command */
00277         if (HAL_QSPI_Command(&QSPIHandle, &sComm
and, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00278         {
00279             return QSPI_ERROR;
00280         }
00281
00282         /* Transmission of the data */
00283         if (HAL_QSPI_Transmit(&QSPIHandle, pData
, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00284         {
00285             return QSPI_ERROR;
00286         }
00287
00288         /* Configure automatic polling mode to w
ait for end of program */

```

```

00289         if (QSPI_AutoPollingMemReady(&QSPIHandle
, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != QSPI_OK)
00290     {
00291         return QSPI_ERROR;
00292     }
00293
00294     /* Update the address and size variables
for next page programming */
00295     current_addr += current_size;
00296     pData += current_size;
00297     current_size = ((current_addr + N25Q128A
_PAGE_SIZE) > end_addr) ? (end_addr - current_addr
) : N25Q128A_PAGE_SIZE;
00298 } while (current_addr < end_addr);
00299
00300 return QSPI_OK;
00301 }
00302
00303 /**
00304  * @brief Erases the specified block of th
e QSPI memory.
00305  * @param BlockAddress: Block address to e
rase
00306  * @retval QSPI memory status
00307  */
00308 uint8_t BSP_QSPI_Erase_Block(uint32_t BlockA
ddress)
00309 {
00310     QSPI_CommandTypeDef sCommand;
00311
00312     /* Initialize the erase command */
00313     sCommand.InstructionMode = QSPI_INSTRUCT
ION_1_LINE;
00314     sCommand.Instruction = SUBSECTOR_ERASE_CMD;
00315     sCommand.AddressMode = QSPI_ADDRESS_
1_LINE;

```

```

00316     sCommand.AddressSize          = QSPI_ADDRESS_
24_BITS;
00317     sCommand.Address                = BlockAddress;
00318     sCommand.AlternateByteMode      = QSPI_ALTERNAT
E_BYTES_NONE;
00319     sCommand.DataMode               = QSPI_DATA_NON
E;
00320     sCommand.DummyCycles            = 0;
00321     sCommand.DdrMode                = QSPI_DDR_MODE
_DISABLE;
00322     sCommand.DdrHoldHalfCycle      = QSPI_DDR_HHC_
ANALOG_DELAY;
00323     sCommand.SIOOMode              = QSPI_SIOO_INS
T_EVERY_CMD;
00324
00325     /* Enable write operations */
00326     if (QSPI_WriteEnable(&QSPIDHandle) != QSPI_
OK)
00327     {
00328         return QSPI_ERROR;
00329     }
00330
00331     /* Send the command */
00332     if (HAL_QSPI_Command(&QSPIDHandle, &sComman
d, HAL_QSPI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00333     {
00334         return QSPI_ERROR;
00335     }
00336
00337     /* Configure automatic polling mode to wai
t for end of erase */
00338     if (QSPI_AutoPollingMemReady(&QSPIDHandle,
N25Q128A_SUBSECTOR_ERASE_MAX_TIME) != QSPI_OK)
00339     {
00340         return QSPI_ERROR;
00341     }
00342

```

```

00343     return QSPI_OK;
00344 }
00345
00346 /**
00347  * @brief Erases the specified sector of the QSPI memory.
00348  * @param Sector: Sector address to erase (0 to 255)
00349  * @retval QSPI memory status
00350  * @note This function is non blocking meaning that sector erase
00351  *        operation is started but not completed when the function
00352  *        returns. Application has to call BSP_QSPI_GetStatus()
00353  *        to know when the device is available again (i.e. erase operation
00354  *        completed).
00355  */
00356 uint8_t BSP_QSPI_Erase_Sector(uint32_t Sector)
00357 {
00358     QSPI_CommandTypeDef sCommand;
00359
00360     if (Sector >= (uint32_t)(N25Q128A_FLASH_SIZE/N25Q128A_SECTOR_SIZE))
00361     {
00362         return QSPI_ERROR;
00363     }
00364
00365     /* Initialize the erase command */
00366     sCommand.InstructionMode = QSPI_INSTRUCTION_1_LINE;
00367     sCommand.Instruction = SECTOR_ERASE_CMD;
00368     sCommand.AddressMode = QSPI_ADDRESS_1_LINE;

```

```

00369     sCommand.AddressSize          = QSPI_ADDRESS_
24_BITS;
00370     sCommand.Address                = (Sector * N25
Q128A_SECTOR_SIZE);
00371     sCommand.AlternateByteMode      = QSPI_ALTERNAT
E_BYTES_NONE;
00372     sCommand.DataMode               = QSPI_DATA_NON
E;
00373     sCommand.DummyCycles            = 0;
00374     sCommand.DdrMode                = QSPI_DDR_MODE
_DISABLE;
00375     sCommand.DdrHoldHalfCycle       = QSPI_DDR_HHC_
ANALOG_DELAY;
00376     sCommand.SIOOMode               = QSPI_SIOO_INS
T_EVERY_CMD;
00377
00378     /* Enable write operations */
00379     if (QSPI_WriteEnable(&QSPIHandle) != QSPI_
OK)
00380     {
00381         return QSPI_ERROR;
00382     }
00383
00384     /* Send the command */
00385     if (HAL_QSPI_Command(&QSPIHandle, &sComman
d, HAL_QSPI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00386     {
00387         return QSPI_ERROR;
00388     }
00389
00390     return QSPI_OK;
00391 }
00392
00393 /**
00394  * @brief Erases the entire QSPI memory.
00395  * @retval QSPI memory status
00396  */

```



```

00397 uint8_t BSP_QSPI_Erase_Chip(void)
00398 {
00399     QSPI_CommandTypeDef sCommand;
00400
00401     /* Initialize the erase command */
00402     sCommand.InstructionMode = QSPI_INSTRUCT
ION_1_LINE;
00403     sCommand.Instruction      = BULK_ERASE_CM
D;
00404     sCommand.AddressMode      = QSPI_ADDRESS_
NONE;
00405     sCommand.AlternateByteMode = QSPI_ALTERNAT
E_BYTES_NONE;
00406     sCommand.DataMode         = QSPI_DATA_NON
E;
00407     sCommand.DummyCycles      = 0;
00408     sCommand.DdrMode          = QSPI_DDR_MODE
_DISABLE;
00409     sCommand.DdrHoldHalfCycle = QSPI_DDR_HHC_
ANALOG_DELAY;
00410     sCommand.SIOOMode         = QSPI_SIOO_INS
T_EVERY_CMD;
00411
00412     /* Enable write operations */
00413     if (QSPI_WriteEnable(&QSPIHandle) != QSPI_
OK)
00414     {
00415         return QSPI_ERROR;
00416     }
00417
00418     /* Send the command */
00419     if (HAL_QSPI_Command(&QSPIHandle, &sComman
d, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00420     {
00421         return QSPI_ERROR;
00422     }
00423

```

```

00424  /* Configure automatic polling mode to wait for end of erase */
00425  if (QSPI_AutoPollingMemReady(&QSPIHandle,
N25Q128A_BULK_ERASE_MAX_TIME) != QSPI_OK)
00426  {
00427      return QSPI_ERROR;
00428  }
00429
00430  return QSPI_OK;
00431 }
00432
00433 /**
00434  * @brief Reads current status of the QSPI
memory.
00435  * @retval QSPI memory status
00436  */
00437 uint8_t BSP_QSPI_GetStatus(void)
00438 {
00439     QSPI_CommandTypeDef sCommand;
00440     uint8_t reg;
00441
00442     /* Initialize the read flag status register command */
00443     sCommand.InstructionMode = QSPI_INSTRUCTION_1_LINE;
00444     sCommand.Instruction = READ_FLAG_STATUS_REG_CMD;
00445     sCommand.AddressMode = QSPI_ADDRESS_NONE;
00446     sCommand.AlternateByteMode = QSPI_ALTERNATE_BYTES_NONE;
00447     sCommand.DataMode = QSPI_DATA_1_LINE;
00448     sCommand.DummyCycles = 0;
00449     sCommand.NbData = 1;
00450     sCommand.DdrMode = QSPI_DDR_MODE_DISABLE;

```

```
00451     sCommand.DdrHoldHalfCycle   = QSPI_DDR_HHC_
ANALOG_DELAY;
00452     sCommand.SI00Mode              = QSPI_SI00_INS
T_EVERY_CMD;
00453
00454     /* Configure the command */
00455     if (HAL_QSPI_Command(&QSPIHandle, &sComman
d, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00456     {
00457         return QSPI_ERROR;
00458     }
00459
00460     /* Reception of the data */
00461     if (HAL_QSPI_Receive(&QSPIHandle, &reg, HA
L_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00462     {
00463         return QSPI_ERROR;
00464     }
00465
00466     /* Check the value of the register */
00467     if ((reg & (N25Q128A_FSR_PRERR | N25Q128A_
FSR_VPPERR | N25Q128A_FSR_PGERR | N25Q128A_FSR_ERE
RR)) != 0)
00468     {
00469         return QSPI_ERROR;
00470     }
00471     else if ((reg & (N25Q128A_FSR_PGSUS | N25Q
128A_FSR_ERSUS)) != 0)
00472     {
00473         return QSPI_SUSPENDED;
00474     }
00475     else if ((reg & N25Q128A_FSR_READY) != 0)
00476     {
00477         return QSPI_OK;
00478     }
00479     else
00480     {
```

```

00481     return QSPI_BUSY;
00482 }
00483 }
00484
00485 /**
00486  * @brief Return the configuration of the
00487  * @param pInfo: pointer on the configurat
00488  * @retval QSPI memory status
00489  */
00490 uint8_t BSP_QSPI_GetInfo(QSPI_Info* pInfo)
00491 {
00492     /* Configure the structure with the memory
00493     configuration */
00494     pInfo->FlashSize          = N25Q128A_FLASH
00495     _SIZE;
00496     pInfo->EraseSectorSize    = N25Q128A_SUBSE
00497     CTOR_SIZE;
00498     pInfo->EraseSectorsNumber = (N25Q128A_FLAS
00499     H_SIZE/N25Q128A_SUBSECTOR_SIZE);
00500     pInfo->ProgPageSize       = N25Q128A_PAGE_
00501     SIZE;
00502     pInfo->ProgPagesNumber    = (N25Q128A_FLAS
00503     H_SIZE/N25Q128A_PAGE_SIZE);
00504     return QSPI_OK;
00505 }
00506
00507 /**
00508  * @brief Configure the QSPI in memory-map
00509  * @retval QSPI memory status
00510  */
00511 uint8_t BSP_QSPI_EnableMemoryMappedMode(void
00512 )
00513 {

```

```

00508     QSPI_CommandTypeDef      sCommand;
00509     QSPI_MemoryMappedTypeDef    sMemMappedCfg;
00510
00511     /* Configure the command for the read instruction */
00512     sCommand.InstructionMode     = QSPI_INSTRUCTION_1_LINE;
00513     sCommand.Instruction         = QUAD_INOUT_FAST_READ_CMD;
00514     sCommand.AddressMode         = QSPI_ADDRESS_4_LINES;
00515     sCommand.AddressSize         = QSPI_ADDRESS_24_BITS;
00516     sCommand.AlternateByteMode   = QSPI_ALTERNATE_BYTES_NONE;
00517     sCommand.DataMode            = QSPI_DATA_4_LINES;
00518     sCommand.DummyCycles         = N25Q128A_DUMMY_CYCLES_READ_QUAD;
00519     sCommand.DdrMode             = QSPI_DDR_MODE_DISABLE;
00520     sCommand.DdrHoldHalfCycle    = QSPI_DDR_HHC_ANALOG_DELAY;
00521     sCommand.SIOOMode           = QSPI_SIOO_INST_EVERY_CMD;
00522
00523     /* Configure the memory mapped mode */
00524     sMemMappedCfg.TimeoutActivation = QSPI_TIMEOUT_COUNTER_ENABLE;
00525     sMemMappedCfg.TimeoutPeriod    = 4; /* 50 ns (4 periods of a 80 MHz clock) */
00526
00527     if (HAL_QSPI_MemoryMapped(&QSPIHandle, &sCommand, &sMemMappedCfg) != HAL_OK)
00528     {
00529         return QSPI_ERROR;
00530     }

```

```

00531
00532     return QSPI_OK;
00533 }
00534
00535 /**
00536  * @brief This function suspends an ongoing erase command.
00537  * @retval QSPI memory status
00538  */
00539 uint8_t BSP_QSPI_SuspendErase(void)
00540 {
00541     QSPI_CommandTypeDef sCommand;
00542
00543     /* Check whether the device is busy (erase
00544        operation is
00545        in progress).
00546        */
00547     if (BSP_QSPI_GetStatus() == QSPI_BUSY)
00548     {
00549         /* Initialize the erase command */
00550         sCommand.InstructionMode = QSPI_INSTRUCTION_1_LINE;
00551         sCommand.Instruction = PROG_ERASE_SUSPEND_CMD;
00552         sCommand.AddressMode = QSPI_ADDRESS_NONE;
00553         sCommand.AlternateByteMode = QSPI_ALTERNATE_BYTES_NONE;
00554         sCommand.DataMode = QSPI_DATA_NONE;
00555         sCommand.DummyCycles = 0;
00556         sCommand.DdrMode = QSPI_DDR_MODE_DISABLE;
00557         sCommand.DdrHoldHalfCycle = QSPI_DDR_HHC_ANALOG_DELAY;
00558         sCommand.SIOOMode = QSPI_SIOO_INST_EVERY_CMD;

```

```

00558
00559     /* Enable write operations */
00560     if (QSPI_WriteEnable(&QSPIHandle) != QSP
I_OK)
00561     {
00562         return QSPI_ERROR;
00563     }
00564
00565     /* Send the command */
00566     if (HAL_QSPI_Command(&QSPIHandle, &sComm
and, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00567     {
00568         return QSPI_ERROR;
00569     }
00570
00571     if (BSP_QSPI_GetStatus() == QSPI_SUSPEND
ED)
00572     {
00573         return QSPI_OK;
00574     }
00575
00576     return QSPI_ERROR;
00577 }
00578
00579 return QSPI_OK;
00580 }
00581
00582 /**
00583  * @brief This function resumes a paused e
rase command.
00584  * @retval QSPI memory status
00585  */
00586 uint8_t BSP_QSPI_ResumeErase(void)
00587 {
00588     QSPI_CommandTypeDef sCommand;
00589
00590     /* Check whether the device is in suspende

```

```

d state */
00591     if (BSP_QSPI_GetStatus() == QSPI_SUSPENDED
)
00592     {
00593         /* Initialize the erase command */
00594         sCommand.InstructionMode    = QSPI_INSTRU
CTION_1_LINE;
00595         sCommand.Instruction        = PROG_ERASE_
RESUME_CMD;
00596         sCommand.AddressMode       = QSPI_ADDRES
S_NONE;
00597         sCommand.AlternateByteMode = QSPI_ALTERN
ATE_BYTES_NONE;
00598         sCommand.DataMode          = QSPI_DATA_N
ONE;
00599         sCommand.DummyCycles       = 0;
00600         sCommand.DdrMode           = QSPI_DDR_MO
DE_DISABLE;
00601         sCommand.DdrHoldHalfCycle  = QSPI_DDR_HH
C_ANALOG_DELAY;
00602         sCommand.SIOOMode          = QSPI_SIOO_I
NST_EVERY_CMD;
00603
00604         /* Enable write operations */
00605         if (QSPI_WriteEnable(&QSPISHandle) != QSP
I_OK)
00606         {
00607             return QSPI_ERROR;
00608         }
00609
00610         /* Send the command */
00611         if (HAL_QSPI_Command(&QSPISHandle, &sComm
and, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00612         {
00613             return QSPI_ERROR;
00614         }
00615

```



```

00616      /*
00617      When this command is executed, the statu
00618      s register write in progress bit is set to 1, and
00619      the flag status register program erase c
00620      ontroller bit is set to 0. This command is ignored
00621      if the device is not in a suspended stat
00622      e.
00623      */
00624      if (BSP_QSPI_GetStatus() == QSPI_BUSY)
00625      {
00626          return QSPI_OK;
00627      }
00628      return QSPI_ERROR;
00629  }
00630  return QSPI_OK;
00631  }
00632
00633  /**
00634   * @}
00635   */
00636
00637  /** @addtogroup STM32L476G_DISCOVERY_QSPI_Pr
00638  ivate_Functions
00639   * @{
00640   */
00641  /**
00642   * @brief Initializes the QSPI MSP.
00643   * @retval None
00644   */
00645  static void QSPI_MspInit(void)
00646  {
00647      GPIO_InitTypeDef GPIO_InitStruct;
00648

```

```

00649  /* Enable the QuadSPI memory interface clock */
00650  __HAL_RCC_QSPI_CLK_ENABLE();
00651
00652  /* Reset the QuadSPI memory interface */
00653  __HAL_RCC_QSPI_FORCE_RESET();
00654  __HAL_RCC_QSPI_RELEASE_RESET();
00655
00656  /* Enable GPIO clocks */
00657  __HAL_RCC_GPIOE_CLK_ENABLE();
00658
00659  /* QSPI CS GPIO pin configuration */
00660  GPIO_InitStruct.Pin      = GPIO_PIN_11;
00661  GPIO_InitStruct.Mode     = GPIO_MODE_AF_PP;
00662  GPIO_InitStruct.Pull     = GPIO_PULLUP;
00663  GPIO_InitStruct.Speed    = GPIO_SPEED_HIGH;
00664  GPIO_InitStruct.Alternate = GPIO_AF10_QSPI;
00665  HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);
00666
00667  /* QSPI CLK, D0, D1, D2 and D3 GPIO pins configuration */
00668  GPIO_InitStruct.Pin      = (GPIO_PIN_10 |
    GPIO_PIN_12 | GPIO_PIN_13 | GPIO_PIN_14 | GPIO_PIN_15);
00669  GPIO_InitStruct.Pull     = GPIO_NOPULL;
00670  HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);
00671 }
00672
00673 /**
00674  * @brief De-Initializes the QSPI MSP.
00675  * @retval None
00676  */
00677 static void QSPI_MspDeInit(void)
00678 {

```

```

00679     GPIO_InitTypeDef GPIO_InitStructure;
00680
00681     /* QSPI CLK, CS, PE10 - PE15 GPIO pins de-
configuration */
00682
00683     __HAL_RCC_GPIOE_CLK_ENABLE();
00684     HAL_GPIO_DeInit(GPIOE, (GPIO_PIN_12 |
GPIO_PIN_13 | GPIO_PIN_14 | GPIO_PIN_15));
00685     /* Set GPIOE pin 11 in pull up mode (optim
um default setting) */
00686     GPIO_InitStructure.Mode          = GPIO_MODE_INPU
T;
00687     GPIO_InitStructure.Pin           = GPIO_PIN_11;
00688     GPIO_InitStructure.Pull          = GPIO_NOPULL;
00689     GPIO_InitStructure.Speed          = GPIO_SPEED_LOW
;
00690     HAL_GPIO_Init(GPIOE, &GPIO_InitStructure);
00691
00692     /* Set GPIOE pin 10 in no pull, low state
(optimum default setting) */
00693     GPIO_InitStructure.Mode          = GPIO_MODE_OUTP
UT_PP ;
00694     GPIO_InitStructure.Pull          = GPIO_NOPULL;
00695     GPIO_InitStructure.Pin           = (GPIO_PIN_10);
00696     HAL_GPIO_Init(GPIOE, &GPIO_InitStructure);
00697     HAL_GPIO_WritePin(GPIOE, GPIO_PIN_10, GPIO
_PIN_RESET);
00698
00699     /* Reset the QuadSPI memory interface */
00700     __HAL_RCC_QSPI_FORCE_RESET();
00701     __HAL_RCC_QSPI_RELEASE_RESET();
00702
00703     /* Disable the QuadSPI memory interface cl
ock */
00704     __HAL_RCC_QSPI_CLK_DISABLE();
00705 }
00706

```

```

00707 /**
00708  * @brief This function reset the QSPI mem
00709  * @param hqspi: QSPI handle
00710  * @retval None
00711  */
00712 static uint8_t QSPI_ResetMemory(QSPI_HandleT
ypeDef *hqspi)
00713 {
00714     QSPI_CommandTypeDef sCommand;
00715
00716     /* Initialize the reset enable command */
00717     sCommand.InstructionMode = QSPI_INSTRUCT
ION_1_LINE;
00718     sCommand.Instruction      = RESET_ENABLE_
CMD;
00719     sCommand.AddressMode      = QSPI_ADDRESS_
NONE;
00720     sCommand.AlternateByteMode = QSPI_ALTERNAT
E_BYTES_NONE;
00721     sCommand.DataMode          = QSPI_DATA_NON
E;
00722     sCommand.DummyCycles      = 0;
00723     sCommand.DdrMode          = QSPI_DDR_MODE
_DISABLE;
00724     sCommand.DdrHoldHalfCycle = QSPI_DDR_HHC_
ANALOG_DELAY;
00725     sCommand.SIOOMode         = QSPI_SIOO_INS
T_EVERY_CMD;
00726
00727     /* Send the command */
00728     if (HAL_QSPI_Command(&QSPIHandle, &sComman
d, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00729     {
00730         return QSPI_ERROR;
00731     }
00732

```

```

00733     /* Send the reset memory command */
00734     sCommand.Instruction = RESET_MEMORY_CMD;
00735     if (HAL_QSPI_Command(&QSPIHandle, &sCommand, HAL_QSPI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00736     {
00737         return QSPI_ERROR;
00738     }
00739
00740     /* Configure automatic polling mode to wait the memory is ready */
00741     if (QSPI_AutoPollingMemReady(&QSPIHandle, HAL_QSPI_TIMEOUT_DEFAULT_VALUE) != QSPI_OK)
00742     {
00743         return QSPI_ERROR;
00744     }
00745
00746     return QSPI_OK;
00747 }
00748
00749 /**
00750  * @brief This function configure the dummy cycles on memory side.
00751  * @param hqspi: QSPI handle
00752  * @retval None
00753  */
00754 static uint8_t QSPI_DummyCyclesCfg(QSPI_HandleTypeDefTypeDef *hqspi)
00755 {
00756     QSPI_CommandTypeDef sCommand;
00757     uint8_t reg;
00758
00759     /* Initialize the read volatile configuration register command */
00760     sCommand.InstructionMode = QSPI_INSTRUCTION_1_LINE;
00761     sCommand.Instruction = READ_VOL_CFG_REG_CMD;

```

```

00762     sCommand.AddressMode           = QSPI_ADDRESS_
NONE;
00763     sCommand.AlternateByteMode = QSPI_ALTERNAT
E_BYTES_NONE;
00764     sCommand.DataMode                 = QSPI_DATA_1_L
INE;
00765     sCommand.DummyCycles               = 0;
00766     sCommand.NbData                    = 1;
00767     sCommand.DdrMode                   = QSPI_DDR_MODE
_DISABLE;
00768     sCommand.DdrHoldHalfCycle         = QSPI_DDR_HHC_
ANALOG_DELAY;
00769     sCommand.SIOOMode                  = QSPI_SIOO_INS
T_EVERY_CMD;
00770
00771     /* Configure the command */
00772     if (HAL_QSPI_Command(&QSPIHandle, &sComman
d, HAL_QSPI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00773     {
00774         return QSPI_ERROR;
00775     }
00776
00777     /* Reception of the data */
00778     if (HAL_QSPI_Receive(&QSPIHandle, &reg, HA
L_QSPI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00779     {
00780         return QSPI_ERROR;
00781     }
00782
00783     /* Enable write operations */
00784     if (QSPI_WriteEnable(&QSPIHandle) != QSPI_
OK)
00785     {
00786         return QSPI_ERROR;
00787     }
00788
00789     /* Update volatile configuration register

```

```

(with new dummy cycles) */
00790     sCommand.Instruction = WRITE_VOL_CFG_REG_C
MD;
00791     MODIFY_REG(reg, N25Q128A_VCR_NB_DUMMY, (N2
5Q128A_DUMMY_CYCLES_READ_QUAD << POSITION_VAL(N25Q
128A_VCR_NB_DUMMY)));
00792
00793     /* Configure the write volatile configurat
ion register command */
00794     if (HAL_QSPI_Command(&QSPIHandle, &sComman
d, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00795     {
00796         return QSPI_ERROR;
00797     }
00798
00799     /* Transmission of the data */
00800     if (HAL_QSPI_Transmit(&QSPIHandle, &reg, H
AL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00801     {
00802         return QSPI_ERROR;
00803     }
00804
00805     return QSPI_OK;
00806 }
00807
00808 /**
00809  * @brief This function send a Write Enabl
e and wait it is effective.
00810  * @param hqspi: QSPI handle
00811  * @retval None
00812  */
00813 static uint8_t QSPI_WriteEnable(QSPI_HandleT
ypeDef *hqspi)
00814 {
00815     QSPI_CommandTypeDef      sCommand;
00816     QSPI_AutoPollingTypeDef  sConfig;
00817

```

```

00818  /* Enable write operations */
00819  sCommand.InstructionMode = QSPI_INSTRUCT
ION_1_LINE;
00820  sCommand.Instruction      = WRITE_ENABLE_
CMD;
00821  sCommand.AddressMode     = QSPI_ADDRESS_
NONE;
00822  sCommand.AlternateByteMode = QSPI_ALTERNAT
E_BYTES_NONE;
00823  sCommand.DataMode        = QSPI_DATA_NON
E;
00824  sCommand.DummyCycles     = 0;
00825  sCommand.DdrMode          = QSPI_DDR_MODE
_DISABLE;
00826  sCommand.DdrHoldHalfCycle = QSPI_DDR_HHC_
ANALOG_DELAY;
00827  sCommand.SIOOMode        = QSPI_SIOO_INS
T_EVERY_CMD;
00828
00829  if (HAL_QSPI_Command(&QSPISHandle, &sComman
d, HAL_QSPI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00830  {
00831      return QSPI_ERROR;
00832  }
00833
00834  /* Configure automatic polling mode to wai
t for write enabling */
00835  sConfig.Match              = N25Q128A_SR_WREN
;
00836  sConfig.Mask               = N25Q128A_SR_WREN
;
00837  sConfig.MatchMode          = QSPI_MATCH_MODE_
AND;
00838  sConfig.StatusBytesSize    = 1;
00839  sConfig.Interval           = 0x10;
00840  sConfig.AutomaticStop      = QSPI_AUTOMATIC_S
TOP_ENABLE;

```



```

00841
00842     sCommand.Instruction      = READ_STATUS_REG_
CMD;
00843     sCommand.DataMode          = QSPI_DATA_1_LINE
;
00844
00845     if (HAL_QSPI_AutoPolling(&QSPIHandle, &sCo
mmand, &sConfig, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) !
= HAL_OK)
00846     {
00847         return QSPI_ERROR;
00848     }
00849
00850     return QSPI_OK;
00851 }
00852
00853 /**
00854  * @brief This function read the SR of the
memory and wait the EOP.
00855  * @param hqspi: QSPI handle
00856  * @param Timeout: Timeout for auto-polling

00857  * @retval None
00858  */
00859 static uint8_t QSPI_AutoPollingMemReady(QSPI
_HandleTypeDef *hqspi, uint32_t Timeout)
00860 {
00861     QSPI_CommandTypeDef      sCommand;
00862     QSPI_AutoPollingTypeDef sConfig;
00863
00864     /* Configure automatic polling mode to wai
t for memory ready */
00865     sCommand.InstructionMode  = QSPI_INSTRUCT
ION_1_LINE;
00866     sCommand.Instruction      = READ_STATUS_R
EG_CMD;
00867     sCommand.AddressMode      = QSPI_ADDRESS_

```

```

NONE;
00868     sCommand.AlternateByteMode = QSPI_ALTERNAT
E_BYTES_NONE;
00869     sCommand.DataMode           = QSPI_DATA_1_L
INE;
00870     sCommand.DummyCycles         = 0;
00871     sCommand.DdrMode             = QSPI_DDR_MODE
_DISABLE;
00872     sCommand.DdrHoldHalfCycle    = QSPI_DDR_HHC_
ANALOG_DELAY;
00873     sCommand.SIOOMode           = QSPI_SIOO_INS
T_EVERY_CMD;
00874
00875     sConfig.Match                 = 0;
00876     sConfig.Mask                  = N25Q128A_SR_WIP;
00877     sConfig.MatchMode             = QSPI_MATCH_MODE_
AND;
00878     sConfig.StatusBytesSize       = 1;
00879     sConfig.Interval              = 0x10;
00880     sConfig.AutomaticStop         = QSPI_AUTOMATIC_S
TOP_ENABLE;
00881
00882     if (HAL_QSPI_AutoPolling(&QSPIHandle, &sCo
mmand, &sConfig, Timeout) != HAL_OK)
00883     {
00884         return QSPI_ERROR;
00885     }
00886
00887     return QSPI_OK;
00888 }
00889
00890 /**
00891  * @}
00892  */
00893
00894 /**
00895  * @}

```

```
00896    */
00897
00898  /**
00899    * @}
00900    */
00901
00902  /**
00903    * @}
00904    */
00905
00906  /******* (C) COPYRIGHT STMicroelectronics *****END OF FILE*****/
00907
```

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#) | [Variables](#)

## Exported Variables

[STM32L476G-DISCOVERY Common](#)

## Defines

#define	__SPI_DIRECTION_2LINES(__HANDLE__)
#define	__SPI_DIRECTION_2LINES_RXONLY(__HANDLE__)
#define	__SPI_DIRECTION_1LINE_TX(__HANDLE__)
#define	__SPI_DIRECTION_1LINE_RX(__HANDLE__)

## Variables

GPIO_TypeDef *	<b>LED_PORT [LEDn]</b> LED variables.
const uint16_t	<b>LED_PIN [LEDn]</b>
GPIO_TypeDef *	<b>JOY_PORT [JOYn]</b> JOYSTICK variables.
const uint16_t	<b>JOY_PIN [JOYn]</b>
const uint8_t	<b>JOY_IRQn [JOYn]</b>
uint32_t	<b>I2c1Timeout =</b> <b>DISCOVERY_I2C2_TIMEOUT_MAX</b> BUS variables.
uint32_t	<b>I2c2Timeout =</b> <b>DISCOVERY_I2C2_TIMEOUT_MAX</b>
static I2C_HandleTypeDef	<b>I2c1Handle</b>
static I2C_HandleTypeDef	<b>I2c2Handle</b>
uint32_t	<b>SpixTimeout = SPIx_TIMEOUT_MAX</b>
static SPI_HandleTypeDef	<b>SpiHandle</b>

## Define Documentation

```
#define __SPI_DIRECTION_1LINE_RX ( __HANDLE__ )
```

**Value:**

```
do {\
    CLEAR_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RXONLY | SPI_CR1_BIDIMODE | SPI_CR1_BIDIOE);\
    SET_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_BIDIMODE);\
} while(0);
```

Definition at line 164 of file stm32l476g\_discovery.c.

Referenced by **ACCELERO\_IO\_Read()**, and **MAGNETO\_IO\_Read()**.

```
#define __SPI_DIRECTION_1LINE_TX ( __HANDLE__ )
```

**Value:**

[illegible]

Definition at line 159 of file stm32l476g\_discovery.c.

Referenced by [ACCELERIO\\_IO\\_Read\(\)](#), [ACCELERIO\\_IO\\_Write\(\)](#), [MAGNETO\\_IO\\_Read\(\)](#), and [MAGNETO\\_IO\\_Write\(\)](#).

**#define** \_\_SPI\_DIRECTION\_2LINES( \_\_HANDLE\_\_ )

**Value:**

```
do{\
    CLEAR_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RXONLY | SPI_CR1_BIDIMODE | SPI_CR1_BIDIOE);\
}while(0);
```

Definition at line **150** of file **stm32l476g\_discovery.c**.

Referenced by **GYRO\_IO\_Read()**, and **GYRO\_IO\_Write()**.

**#define** \_\_SPI\_DIRECTION\_2LINES\_RXONLY( \_\_HANDLE\_\_ )

**Value:**

```
do{\
    CLEAR_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RXONLY | SPI_CR1_BIDIMODE | SPI_CR1_BIDIOE);\
    SET_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RXONLY);\
}while(0);
```

Definition at line **154** of file **stm32l476g\_discovery.c**.



## Variable Documentation

**I2C\_HandleTypeDef I2c1Handle** [static]

Definition at line **143** of file **stm32l476g\_discovery.c**.

Referenced by **I2C1\_DeInit()**, **I2C1\_Error()**, **I2C1\_Init()**, **I2C1\_ReadBuffer()**, and **I2C1\_WriteBuffer()**.

**uint32\_t I2c1Timeout = DISCOVERY\_I2C2\_TIMEOUT\_MAX**

BUS variables.

Definition at line **141** of file **stm32l476g\_discovery.c**.

Referenced by **I2C1\_ReadBuffer()**, and **I2C1\_WriteBuffer()**.

**I2C\_HandleTypeDef I2c2Handle** [static]

Definition at line **144** of file **stm32l476g\_discovery.c**.

Referenced by **I2C2\_DeInit()**, **I2C2\_Error()**, **I2C2\_Init()**, **I2C2\_ReadBuffer()**, **I2C2\_ReadData()**, **I2C2\_WriteBuffer()**, and **I2C2\_WriteData()**.

**uint32\_t I2c2Timeout = DISCOVERY\_I2C2\_TIMEOUT\_MAX**

Definition at line **142** of file **stm32l476g\_discovery.c**.

Referenced by **I2C2\_ReadBuffer()**, **I2C2\_ReadData()**, **I2C2\_WriteBuffer()**, and **I2C2\_WriteData()**.

**const uint8\_t JOY\_IRQn[JOYn]**

Initial value:

```
{SEL_JOY_EXTI_IRQn,
                                LEFT_JOY_EXTI_I
RQn,
                                RIGHT_JOY_EXTI_
IRQn,
                                DOWN_JOY_EXTI_I
RQn,
                                UP_JOY_EXTI_IRQn
}
```

Definition at line **131** of file **stm32l476g\_discovery.c**.

Referenced by **BSP\_JOY\_Init()**.

**const uint16\_t JOY\_PIN[JOYn]**

Initial value:

```
{SEL_JOY_PIN,
                                LEFT_JOY_PIN,
                                RIGHT_JOY_PIN,
                                DOWN_JOY_PIN,
                                UP_JOY_PIN}
```

Definition at line **125** of file **stm32l476g\_discovery.c**.

Referenced by **BSP\_JOY\_DeInit()**, **BSP\_JOY\_GetState()**, and **BSP\_JOY\_Init()**.

**GPIO\_TypeDef\* JOY\_PORT[JOYn]**

Initial value:

```
{SEL_JOY_GPIO_PORT,
```

ORT,	DOWN_JOY_GPIO_P
ORT,	LEFT_JOY_GPIO_P
PORT,	RIGHT_JOY_GPIO_
}	UP_JOY_GPIO_PORT

JOYSTICK variables.

Definition at line **119** of file **stm32l476g\_discovery.c**.

Referenced by **BSP\_JOY\_DeInit()**, **BSP\_JOY\_GetState()**, and **BSP\_JOY\_Init()**.

**const uint16\_t LED\_PIN[LEDn]**

Initial value:

{LED4_PIN,	LED5_PIN}
------------	-----------

Definition at line **105** of file **stm32l476g\_discovery.c**.

Referenced by **BSP\_LED\_DeInit()**, **BSP\_LED\_Init()**, **BSP\_LED\_Off()**, **BSP\_LED\_On()**, and **BSP\_LED\_Toggle()**.

**GPIO\_TypeDef\* LED\_PORT[LEDn]**

Initial value:

{LED4_GPIO_PORT,	LED5_GPIO_PORT}
------------------	-----------------

LED variables.

Definition at line **102** of file **stm32l476g\_discovery.c**.

Referenced by **BSP\_LED\_DeInit()**, **BSP\_LED\_Init()**, **BSP\_LED\_Off()**, **BSP\_LED\_On()**, and **BSP\_LED\_Toggle()**.

**SPI\_HandleTypeDef SpiHandle** [static]

Definition at line **171** of file **stm32l476g\_discovery.c**.

Referenced by **ACCELERO\_IO\_Read()**, **ACCELERO\_IO\_Write()**, **GYRO\_IO\_Read()**, **GYRO\_IO\_Write()**, **MAGNETO\_IO\_Read()**, **MAGNETO\_IO\_Write()**, **SPIx\_DeInit()**, **SPIx\_Init()**, **SPIx\_Read()**, **SPIx\_Write()**, and **SPIx\_WriteRead()**.

**uint32\_t SpixTimeout = SPIx\_TIMEOUT\_MAX**

Definition at line **170** of file **stm32l476g\_discovery.c**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## Private Defines

[STM32L476G-DISCOVERY Common](#)

# Defines

#define	__STM32L476G_DISCOVERY_BSP_VERSION_MAIN	(0x00)
	STM32L476G DISCOVERY BSP Driver version number	
	\$VERSION\$.	
#define	__STM32L476G_DISCOVERY_BSP_VERSION_SUB1	(0x00)
#define	__STM32L476G_DISCOVERY_BSP_VERSION_SUB2	(0x00)
#define	__STM32L476G_DISCOVERY_BSP_VERSION_RC	(0x01)
#define	__STM32L476G_DISCOVERY_BSP_VERSION	

## Define Documentation

**#define** `__STM32L476G_DISCOVERY_BSP_VERSION`

**Value:**

```
((__STM32L476G_DISCOVERY_BSP_VERSION_MAIN << 24) \
    | (__STM32L476G_DISCOVERY_BSP_VERSION_SUB1 <<
16) \
    | (__STM32L476G_DISCOVERY_BSP_VERSION_SUB2 <<
8 ) \
    | (__STM32L476G_DISCOVERY_BSP_VERSION_RC))
```

Definition at line **76** of file `stm32l476g_discovery.c`.

Referenced by `BSP_GetVersion()`.

**#define** `__STM32L476G_DISCOVERY_BSP_VERSION_MAIN` (0x00

STM32L476G DISCOVERY BSP Driver version number \$VERSION\$.

[31:24] main version

Definition at line **72** of file `stm32l476g_discovery.c`.

**#define** `__STM32L476G_DISCOVERY_BSP_VERSION_RC` (0x01)

[7:0] release candidate

Definition at line **75** of file `stm32l476g_discovery.c`.

**#define** `__STM32L476G_DISCOVERY_BSP_VERSION_SUB1` (0x00

[23:16] sub1 version

Definition at line **73** of file **stm32l476g\_discovery.c**.

**#define \_\_STM32L476G\_DISCOVERY\_BSP\_VERSION\_SUB2** (0x00

[15:8] sub2 version

Definition at line **74** of file **stm32l476g\_discovery.c**.



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## BUS Constants

[Exported Constants](#)

## Defines

#define	<b>DISCOVERY_SPIx</b>	SPI2
#define	<b>DISCOVERY_SPIx_CLOCK_ENABLE()</b>	__HAL_RCC_SPI2_
#define	<b>DISCOVERY_SPIx_CLOCK_DISABLE()</b>	__HAL_RCC_SPI2_
#define	<b>DISCOVERY_SPIx_GPIO_PORT</b>	GPIOA /* GPIOA */
#define	<b>DISCOVERY_SPIx_AF</b>	GPIO_AF5_SPI2
#define	<b>DISCOVERY_SPIx_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GI
#define	<b>DISCOVERY_SPIx_GPIO_CLK_DISABLE()</b>	__HAL_RCC_G
#define	<b>DISCOVERY_SPIx_GPIO_FORCE_RESET()</b>	__HAL_RCC_
#define	<b>DISCOVERY_SPIx_GPIO_RELEASE_RESET()</b>	__HAL_RCC
#define	<b>DISCOVERY_SPIx_SCK_PIN</b>	GPIO_PIN_1 /* PD.01 */
#define	<b>DISCOVERY_SPIx_MISO_PIN</b>	GPIO_PIN_3 /* PD.03 */
#define	<b>DISCOVERY_SPIx_MOSI_PIN</b>	GPIO_PIN_4 /* PD.04 */
#define	<b>SPIx_TIMEOUT_MAX</b>	((uint32_t)0x1000)
#define	<b>READWRITE_CMD</b>	((uint8_t)0x80)
#define	<b>MULTIPLEBYTE_CMD</b>	((uint8_t)0x40)
#define	<b>DUMMY_BYTE</b>	((uint8_t)0x00)
#define	<b>DISCOVERY_I2C1_SCL_GPIO_PORT</b>	GPIOB
#define	<b>DISCOVERY_I2C1_SDA_GPIO_PORT</b>	GPIOB
#define	<b>DISCOVERY_I2C1_SCL_PIN</b>	GPIO_PIN_6
#define	<b>DISCOVERY_I2C1_SDA_PIN</b>	GPIO_PIN_7
#define	<b>DISCOVERY_I2C1_SCL_SDA_AF</b>	GPIO_AF4_I2C1
#define	<b>DISCOVERY_I2C1</b>	I2C1
#define	<b>DISCOVERY_I2C1_CLK_ENABLE()</b>	__HAL_RCC_I2C1_CLI
#define	<b>DISCOVERY_I2C1_CLK_DISABLE()</b>	__HAL_RCC_I2C1_CL
#define	<b>DISCOVERY_I2C1_SDA_GPIO_CLK_ENABLE()</b>	__HAL_RC
#define	<b>DISCOVERY_I2C1_SCL_GPIO_CLK_ENABLE()</b>	__HAL_RC
#define	<b>DISCOVERY_I2C1_SDA_GPIO_CLK_DISABLE()</b>	__HAL_R
#define	<b>DISCOVERY_I2C1_SCL_GPIO_CLK_DISABLE()</b>	__HAL_R
#define	<b>DISCOVERY_I2C1_FORCE_RESET()</b>	__HAL_RCC_I2C1_F
#define	<b>DISCOVERY_I2C1_RELEASE_RESET()</b>	__HAL_RCC_I2C1.
#define	<b>DISCOVERY_I2C1_EV_IRQn</b>	I2C1_EV_IRQn

```

#define DISCOVERY_I2C1_EV_IRQHandler I2C1_EV_IRQHandler
#define DISCOVERY_I2C1_ER_IRQn I2C1_ER_IRQn
#define DISCOVERY_I2C1_ER_IRQHandler I2C1_ER_IRQHandler
#define AUDIO_I2C_ADDRESS ((uint16_t) 0x94)
#define DISCOVERY_I2C1_TIMEOUT_MAX 3000
#define DISCOVERY_I2C2_SCL_PIN GPIO_PIN_10
#define DISCOVERY_I2C2_SCL_GPIO_PORT GPIOB
#define DISCOVERY_I2C2_SDA_PIN GPIO_PIN_11
#define DISCOVERY_I2C2_SDA_GPIO_PORT GPIOB
#define DISCOVERY_I2C2_SCL_SDA_AF GPIO_AF4_I2C2
#define DISCOVERY_I2C2 I2C2
#define DISCOVERY_I2C2_CLK_ENABLE() __HAL_RCC_I2C2_CLK_ENABLE()
#define DISCOVERY_I2C2_CLK_DISABLE() __HAL_RCC_I2C2_CLK_DISABLE()
#define DISCOVERY_I2C2_SDA_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define DISCOVERY_I2C2_SCL_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define DISCOVERY_I2C2_SDA_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define DISCOVERY_I2C2_SCL_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define DISCOVERY_I2C2_FORCE_RESET() __HAL_RCC_I2C2_FORCE_RESET()
#define DISCOVERY_I2C2_RELEASE_RESET() __HAL_RCC_I2C2_RELEASE_RESET()
#define DISCOVERY_I2C2_EV_IRQn I2C2_EV_IRQn
#define DISCOVERY_I2C2_ER_IRQn I2C2_ER_IRQn
#define IDD_I2C_ADDRESS ((uint16_t) 0x84)
#define DISCOVERY_I2C2_TIMEOUT_MAX 3000
#define ACCELERO_CS_LOW() HAL_GPIO_WritePin(ACCELERO_CS_PIN, GPIO_PIN_RESET)
Accelerometer Chip Select macro definition.
#define ACCELERO_CS_HIGH() HAL_GPIO_WritePin(ACCELERO_CS_PIN, GPIO_PIN_SET)
#define ACCELERO_CS_GPIO_PORT GPIOE /* GPIOE */
Accelerometer SPI Interface pins.
#define ACCELERO_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOE_CLK_ENABLE()
#define ACCELERO_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPIOE_CLK_DISABLE()
#define ACCELERO_CS_PIN GPIO_PIN_0 /* PE.00 */
#define ACCELERO_XLINT_GPIO_PORT GPIOE /* GPIOE */

```

Accelerometer Interrupt pins.

#define **ACCELERO\_XLINT\_GPIO\_CLK\_ENABLE()** \_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE()

#define **ACCELERO\_XLINT\_GPIO\_CLK\_DISABLE()** \_\_HAL\_RCC\_GPIOA\_CLK\_DISABLE()

#define **ACCELERO\_XLINT\_PIN** GPIO\_PIN\_1 /\* PE.01 \*/

#define **ACCELERO\_XLINT\_EXTI\_IRQn** EXTI1\_IRQn

#define **MAGNETO\_CS\_LOW()** HAL\_GPIO\_WritePin(**MAGNETO\_CS\_GPIO\_PORT**,  
**MAGNETO\_CS\_PIN**, GPIO\_PIN\_RESET)

Magnetometer Chip Select macro definition.

#define **MAGNETO\_CS\_HIGH()** HAL\_GPIO\_WritePin(**MAGNETO\_CS\_GPIO\_PORT**,  
**MAGNETO\_CS\_PIN**, GPIO\_PIN\_SET)

#define **MAGNETO\_CS\_GPIO\_PORT** GPIOC /\* GPIOC \*/

Magnetometer SPI Interface pins.

#define **MAGNETO\_CS\_GPIO\_CLK\_ENABLE()** \_\_HAL\_RCC\_GPIOC\_CLK\_ENABLE()

#define **MAGNETO\_CS\_GPIO\_CLK\_DISABLE()** \_\_HAL\_RCC\_GPIOC\_CLK\_DISABLE()

#define **MAGNETO\_CS\_PIN** GPIO\_PIN\_0 /\* PC.00 \*/

#define **MAGNETO\_INT\_GPIO\_PORT** GPIOC /\* GPIOC \*/

Magnetometer Interrupt pins.

#define **MAGNETO\_INT\_GPIO\_CLK\_ENABLE()** \_\_HAL\_RCC\_GPIOC\_CLK\_ENABLE()

#define **MAGNETO\_INT\_GPIO\_CLK\_DISABLE()** \_\_HAL\_RCC\_GPIOC\_CLK\_DISABLE()

#define **MAGNETO\_INT1\_PIN** GPIO\_PIN\_1 /\* PC.01 \*/

#define **MAGNETO\_INT1\_EXTI\_IRQn** EXTI1\_IRQn

#define **MAGNETO\_DRDY\_GPIO\_PORT** GPIOC /\* GPIOC \*/

#define **MAGNETO\_DRDY\_GPIO\_CLK\_ENABLE()** \_\_HAL\_RCC\_GPIOC\_CLK\_ENABLE()

#define **MAGNETO\_DRDY\_GPIO\_CLK\_DISABLE()** \_\_HAL\_RCC\_GPIOC\_CLK\_DISABLE()

#define **MAGNETO\_DRDY\_PIN** GPIO\_PIN\_2 /\* PC.01 \*/

#define **CODEC\_AUDIO\_POWER\_OFF()** HAL\_GPIO\_WritePin(**AUDIO\_RESET\_GPIO\_PORT**,  
**AUDIO\_RESET\_PIN**, GPIO\_PIN\_RESET)

Audio codec chip reset definition.

#define **CODEC\_AUDIO\_POWER\_ON()** HAL\_GPIO\_WritePin(**AUDIO\_RESET\_GPIO\_PORT**,  
**AUDIO\_RESET\_PIN**, GPIO\_PIN\_SET)

#define **AUDIO\_RESET\_GPIO\_CLK\_ENABLE()** \_\_HAL\_RCC\_GPIOE\_CLK\_ENABLE()

#define **AUDIO\_RESET\_GPIO\_CLK\_DISABLE()** \_\_HAL\_RCC\_GPIOE\_CLK\_DISABLE()

#define **AUDIO\_RESET\_PIN** GPIO\_PIN\_3

#define **AUDIO\_RESET\_GPIO** GPIOE

```

#define GYRO_CS_LOW() HAL_GPIO_WritePin(GYRO_CS_GPIO_PORT,  
GPIO_PIN_RESET)  
Gyroscope Chip Select macro definition.

#define GYRO_CS_HIGH() HAL_GPIO_WritePin(GYRO_CS_GPIO_PORT,  
GPIO_PIN_SET)

#define GYRO_CS_GPIO_PORT GPIOD /* GPIOD */  
Gyroscope SPI Interface pins.

#define GYRO_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define GYRO_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPIOD_CLK_DISABLE()
#define GYRO_CS_PIN GPIO_PIN_7 /* PD.07 */
#define GYRO_INT1_GPIO_PORT GPIOD /* GPIOD */  
Gyroscope Interrupt pins.

#define GYRO_INT1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define GYRO_INT1_GPIO_CLK_DISABLE() __HAL_RCC_GPIOD_CLK_DISABLE()
#define GYRO_INT1_PIN GPIO_PIN_2 /* PD.02 */
#define GYRO_INT1_EXTI_IRQn EXTI2_IRQn
#define GYRO_INT2_GPIO_PORT GPIOB /* GPIOB */
#define GYRO_INT2_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define GYRO_INT2_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define GYRO_INT2_PIN GPIO_PIN_8 /* PB.08 */
#define GYRO_INT2_EXTI_IRQn EXTI9_5_IRQn
#define IDD_INT_GPIO_PORT GPIOC /* GPIOC */  
Idd current measurement interface pins.

#define IDD_INT_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC_CLK_ENABLE()
#define IDD_INT_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC_CLK_DISABLE()
#define IDD_INT_PIN GPIO_PIN_13 /* PC.13 */
#define IDD_INT_EXTI_IRQn EXTI15_10_IRQn
#define IDD_WAKEUP_GPIO_PORT GPIOA /* GPIOA */
#define IDD_WAKEUP_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()
#define IDD_WAKEUP_GPIO_CLK_DISABLE() __HAL_RCC_GPIOA_CLK_DISABLE()
#define IDD_WAKEUP_PIN GPIO_PIN_4 /* PA.04 */

```

## Define Documentation

**#define ACCELERO\_CS\_GPIO\_CLK\_DISABLE ( ) \_\_HAL\_RCC\_G**

Definition at line **406** of file **stm32l476g\_discovery.h**.

**#define ACCELERO\_CS\_GPIO\_CLK\_ENABLE ( ) \_\_HAL\_RCC\_GF**

Definition at line **405** of file **stm32l476g\_discovery.h**.

Referenced by **ACCELERO\_IO\_DeInit()**, **ACCELERO\_IO\_Init()**, and **GYRO\_IO\_Init()**.

**#define ACCELERO\_CS\_GPIO\_PORT GPIOE /\* GPIOE \*/**

Accelerometer SPI Interface pins.

Definition at line **404** of file **stm32l476g\_discovery.h**.

Referenced by **ACCELERO\_IO\_DeInit()**, **ACCELERO\_IO\_Init()**, and **GYRO\_IO\_Init()**.

**#define ACCELERO\_CS\_HIGH ( ) HAL\_GPIO\_WritePin(ACCELER**

Definition at line **399** of file **stm32l476g\_discovery.h**.

Referenced by **ACCELERO\_IO\_DeInit()**, **ACCELERO\_IO\_Init()**, **ACCELERO\_IO\_Read()**, **ACCELERO\_IO\_Write()**, and **GYRO\_IO\_Init()**.

**#define ACCELERO\_CS\_LOW ( ) HAL\_GPIO\_WritePin(ACCELER**

Accelerometer Chip Select macro definition.

Definition at line **398** of file **stm32l476g\_discovery.h**.

Referenced by **ACCELERO\_IO\_Read()**, and **ACCELERO\_IO\_Write()**.

```
#define ACCELERO_CS_PIN GPIO_PIN_0 /* PE.00 */
```

Definition at line **407** of file **stm32l476g\_discovery.h**.

Referenced by **ACCELERO\_IO\_DeInit()**, **ACCELERO\_IO\_Init()**, and **GYRO\_IO\_Init()**.

```
#define ACCELERO_XLINT_EXTI_IRQn EXTI1_IRQn
```

Definition at line **416** of file **stm32l476g\_discovery.h**.

```
#define ACCELERO_XLINT_GPIO_CLK_DISABLE ( ) __HAL_RCC
```

Definition at line **414** of file **stm32l476g\_discovery.h**.

```
#define ACCELERO_XLINT_GPIO_CLK_ENABLE ( ) __HAL_RCC_
```

Definition at line **413** of file **stm32l476g\_discovery.h**.

```
#define ACCELERO_XLINT_GPIO_PORT GPIOE /* GPIOE */
```

Accelerometer Interrupt pins.

Definition at line **412** of file **stm32l476g\_discovery.h**.

```
#define ACCELERO_XLINT_PIN GPIO_PIN_1 /* PE.01 */
```

Definition at line **415** of file **stm32l476g\_discovery.h**.

```
#define AUDIO_I2C_ADDRESS ((uint16_t) 0x94)
```

Definition at line **335** of file **stm32l476g\_discovery.h**.

Referenced by **AUDIO\_CODEC\_Reset()**, **BSP\_AUDIO\_OUT\_Init()**, **BSP\_AUDIO\_OUT\_Pause()**, **BSP\_AUDIO\_OUT\_Play()**, **BSP\_AUDIO\_OUT\_Resume()**, **BSP\_AUDIO\_OUT\_SetMute()**, **BSP\_AUDIO\_OUT\_SetOutputMode()**, **BSP\_AUDIO\_OUT\_SetVolume()**, and **BSP\_AUDIO\_OUT\_Stop()**.

```
#define AUDIO_RESET_GPIO GPIOE
```

Definition at line **461** of file **stm32l476g\_discovery.h**.

Referenced by **AUDIO\_IO\_Init()**.

```
#define AUDIO_RESET_GPIO_CLK_DISABLE ( ) __HAL_RCC_GP
```

Definition at line **459** of file **stm32l476g\_discovery.h**.

```
#define AUDIO_RESET_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPI
```

Definition at line **458** of file **stm32l476g\_discovery.h**.

Referenced by **AUDIO\_IO\_Init()**.

```
#define AUDIO_RESET_PIN GPIO_PIN_3
```



Definition at line **460** of file **stm32l476g\_discovery.h**.

Referenced by **AUDIO\_IO\_Init()**.

**#define CODEC\_AUDIO\_POWER\_OFF ( ) HAL\_GPIO\_WritePin(AL**

Audio codec chip reset definition.

Definition at line **454** of file **stm32l476g\_discovery.h**.

Referenced by **AUDIO\_IO\_Init()**.

**#define CODEC\_AUDIO\_POWER\_ON ( ) HAL\_GPIO\_WritePin(AUI**

Definition at line **455** of file **stm32l476g\_discovery.h**.

Referenced by **AUDIO\_IO\_Init()**.

**#define DISCOVERY\_I2C1 I2C1**

Definition at line **302** of file **stm32l476g\_discovery.h**.

Referenced by **I2C1\_Init()**, **I2C1\_MspDeInit()**, and **I2C1\_MspInit()**.

**#define DISCOVERY\_I2C1\_CLK\_DISABLE ( ) \_\_HAL\_RCC\_I2C1\_C**

Definition at line **304** of file **stm32l476g\_discovery.h**.

Referenced by **I2C1\_MspDeInit()**.

**#define DISCOVERY\_I2C1\_CLK\_ENABLE ( ) \_\_HAL\_RCC\_I2C1\_C**

Definition at line **303** of file **stm32l476g\_discovery.h**.

Referenced by [I2C1\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C1\_ER\_IRQHandler** I2C1\_ER\_IRQHandler

Definition at line **316** of file [stm32l476g\\_discovery.h](#).

**#define DISCOVERY\_I2C1\_ER\_IRQn** I2C1\_ER\_IRQn

Definition at line **315** of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C1\\_MspDeInit\(\)](#), and [I2C1\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C1\_EV\_IRQHandler** I2C1\_EV\_IRQHandler

Definition at line **314** of file [stm32l476g\\_discovery.h](#).

**#define DISCOVERY\_I2C1\_EV\_IRQn** I2C1\_EV\_IRQn

Definition at line **313** of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C1\\_MspDeInit\(\)](#), and [I2C1\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C1\_FORCE\_RESET ( )** \_\_HAL\_RCC\_I2C1\_

Definition at line **309** of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C1\\_MspDeInit\(\)](#), and [I2C1\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C1\_RELEASE\_RESET ( )** \_\_HAL\_RCC\_I2C1\_

Definition at line **310** of file [stm32l476g\\_discovery.h](#).

Referenced by **I2C1\_MspDeInit()**, and **I2C1\_MspInit()**.

**#define DISCOVERY\_I2C1\_SCL\_GPIO\_CLK\_DISABLE ( ) \_\_HAL\_**

Definition at line **308** of file **stm32l476g\_discovery.h**.

**#define DISCOVERY\_I2C1\_SCL\_GPIO\_CLK\_ENABLE ( ) \_\_HAL\_F**

Definition at line **306** of file **stm32l476g\_discovery.h**.

Referenced by **I2C1\_MspDeInit()**, and **I2C1\_MspInit()**.

**#define DISCOVERY\_I2C1\_SCL\_GPIO\_PORT GPIOB**

Definition at line **290** of file **stm32l476g\_discovery.h**.

Referenced by **I2C1\_MspDeInit()**, and **I2C1\_MspInit()**.

**#define DISCOVERY\_I2C1\_SCL\_PIN GPIO\_PIN\_6**

Definition at line **293** of file **stm32l476g\_discovery.h**.

Referenced by **I2C1\_MspDeInit()**, and **I2C1\_MspInit()**.

**#define DISCOVERY\_I2C1\_SCL\_SDA\_AF GPIO\_AF4\_I2C1**

Definition at line **299** of file **stm32l476g\_discovery.h**.

Referenced by **I2C1\_MspInit()**.

**#define DISCOVERY\_I2C1\_SDA\_GPIO\_CLK\_DISABLE ( ) \_\_HAL\_**

Definition at line **307** of file [stm32l476g\\_discovery.h](#).

**#define DISCOVERY\_I2C1\_SDA\_GPIO\_CLK\_ENABLE ( ) \_\_HAL\_I**

Definition at line **305** of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C1\\_MspDeInit\(\)](#), and [I2C1\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C1\_SDA\_GPIO\_PORT GPIOB**

Definition at line **291** of file [stm32l476g\\_discovery.h](#).

**#define DISCOVERY\_I2C1\_SDA\_PIN GPIO\_PIN\_7**

Definition at line **294** of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C1\\_MspDeInit\(\)](#), and [I2C1\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C1\_TIMEOUT\_MAX 3000**

Definition at line **342** of file [stm32l476g\\_discovery.h](#).

**#define DISCOVERY\_I2C2 I2C2**

Definition at line **355** of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C2\\_Init\(\)](#), [I2C2\\_MspDeInit\(\)](#), and [I2C2\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C2\_CLK\_DISABLE ( ) \_\_HAL\_RCC\_I2C2\_C**

Definition at line **357** of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C2\\_MspDeInit\(\)](#).

**#define DISCOVERY\_I2C2\_CLK\_ENABLE ( ) \_\_HAL\_RCC\_I2C2\_C**

Definition at line [356](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C2\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C2\_ER\_IRQn I2C2\_ER\_IRQn**

Definition at line [367](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C2\\_MspDeInit\(\)](#), and [I2C2\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C2\_EV\_IRQn I2C2\_EV\_IRQn**

Definition at line [366](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C2\\_MspDeInit\(\)](#), and [I2C2\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C2\_FORCE\_RESET ( ) \_\_HAL\_RCC\_I2C2\_**

Definition at line [362](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C2\\_MspDeInit\(\)](#), and [I2C2\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C2\_RELEASE\_RESET ( ) \_\_HAL\_RCC\_I2C**

Definition at line [363](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [I2C2\\_MspDeInit\(\)](#), and [I2C2\\_MspInit\(\)](#).

**#define DISCOVERY\_I2C2\_SCL\_GPIO\_CLK\_DISABLE ( ) \_\_HAL\_**

Definition at line **361** of file **stm32l476g\_discovery.h**.

**#define DISCOVERY\_I2C2\_SCL\_GPIO\_CLK\_ENABLE ( ) \_\_HAL\_F**

Definition at line **359** of file **stm32l476g\_discovery.h**.

Referenced by **I2C2\_MspDeInit()**, and **I2C2\_MspInit()**.

**#define DISCOVERY\_I2C2\_SCL\_GPIO\_PORT GPIOB**

Definition at line **350** of file **stm32l476g\_discovery.h**.

Referenced by **I2C2\_MspDeInit()**, and **I2C2\_MspInit()**.

**#define DISCOVERY\_I2C2\_SCL\_PIN GPIO\_PIN\_10**

Definition at line **349** of file **stm32l476g\_discovery.h**.

Referenced by **I2C2\_MspDeInit()**, and **I2C2\_MspInit()**.

**#define DISCOVERY\_I2C2\_SCL\_SDA\_AF GPIO\_AF4\_I2C2**

Definition at line **353** of file **stm32l476g\_discovery.h**.

Referenced by **I2C2\_MspInit()**.

**#define DISCOVERY\_I2C2\_SDA\_GPIO\_CLK\_DISABLE ( ) \_\_HAL\_**

Definition at line **360** of file **stm32l476g\_discovery.h**.

**#define DISCOVERY\_I2C2\_SDA\_GPIO\_CLK\_ENABLE ( ) \_\_HAL\_I**

Definition at line **358** of file **stm32l476g\_discovery.h**.

Referenced by **I2C2\_MspDeInit()**, and **I2C2\_MspInit()**.

**#define DISCOVERY\_I2C2\_SDA\_GPIO\_PORT GPIOB**

Definition at line **352** of file **stm32l476g\_discovery.h**.

**#define DISCOVERY\_I2C2\_SDA\_PIN GPIO\_PIN\_11**

Definition at line **351** of file **stm32l476g\_discovery.h**.

Referenced by **I2C2\_MspDeInit()**, and **I2C2\_MspInit()**.

**#define DISCOVERY\_I2C2\_TIMEOUT\_MAX 3000**

Definition at line **391** of file **stm32l476g\_discovery.h**.

**#define DISCOVERY\_SPIx SPI2**

Definition at line **257** of file **stm32l476g\_discovery.h**.

Referenced by **SPIx\_Init()**.

**#define DISCOVERY\_SPIx\_AF GPIO\_AF5\_SPI2**

Definition at line **261** of file **stm32l476g\_discovery.h**.

Referenced by **SPIx\_MspInit()**.

**#define DISCOVERY\_SPIx\_CLOCK\_DISABLE ( ) \_\_HAL\_RCC\_SPI**

Definition at line **259** of file **stm32l476g\_discovery.h**.

Referenced by **SPIx\_MspDeInit()**.

**#define DISCOVERY\_SPIx\_CLOCK\_ENABLE ( ) \_\_HAL\_RCC\_SPI**

Definition at line **258** of file **stm32l476g\_discovery.h**.

Referenced by **SPIx\_MspInit()**.

**#define DISCOVERY\_SPIx\_GPIO\_CLK\_DISABLE ( ) \_\_HAL\_RCC\_**

Definition at line **263** of file **stm32l476g\_discovery.h**.

**#define DISCOVERY\_SPIx\_GPIO\_CLK\_ENABLE ( ) \_\_HAL\_RCC\_ (**

Definition at line **262** of file **stm32l476g\_discovery.h**.

Referenced by **SPIx\_MspDeInit()**, and **SPIx\_MspInit()**.

**#define DISCOVERY\_SPIx\_GPIO\_FORCE\_RESET ( ) \_\_HAL\_RCC**

Definition at line **264** of file **stm32l476g\_discovery.h**.

Referenced by **SPIx\_MspDeInit()**.

**#define DISCOVERY\_SPIx\_GPIO\_PORT GPIOID /\* GPIOID \*/**

Definition at line **260** of file **stm32l476g\_discovery.h**.



Referenced by [SPIx\\_MspDeInit\(\)](#), and [SPIx\\_MspInit\(\)](#).

```
#define DISCOVERY_SPIx_GPIO_RELEASE_RESET ( ) __HAL_RCC_GPIOA_CLK_ENABLE()
```

Definition at line [265](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [SPIx\\_MspDeInit\(\)](#).

```
#define DISCOVERY_SPIx_MISO_PIN GPIO_PIN_3 /* PD.03 */
```

Definition at line [267](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [SPIx\\_MspDeInit\(\)](#), and [SPIx\\_MspInit\(\)](#).

```
#define DISCOVERY_SPIx_MOSI_PIN GPIO_PIN_4 /* PD.04 */
```

Definition at line [268](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [SPIx\\_MspDeInit\(\)](#), and [SPIx\\_MspInit\(\)](#).

```
#define DISCOVERY_SPIx_SCK_PIN GPIO_PIN_1 /* PD.01*/
```

Definition at line [266](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [SPIx\\_MspDeInit\(\)](#), and [SPIx\\_MspInit\(\)](#).

```
#define DUMMY_BYTE ((uint8_t)0x00)
```

Definition at line [281](#) of file [stm32l476g\\_discovery.h](#).

```
#define GYRO_CS_GPIO_CLK_DISABLE ( ) __HAL_RCC_GPIOC_CLK_DISABLE()
```

Definition at line **475** of file **stm32l476g\_discovery.h**.

```
#define GYRO_CS_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIOD_
```

Definition at line **474** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define GYRO_CS_GPIO_PORT GPIOD /* GPIOD */
```

Gyroscope SPI Interface pins.

Definition at line **473** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define GYRO_CS_HIGH ( ) HAL_GPIO_WritePin(GYRO_CS_GPIOC
```

Definition at line **468** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, **GYRO\_IO\_Init()**, **GYRO\_IO\_Read()**, and **GYRO\_IO\_Write()**.

```
#define GYRO_CS_LOW ( ) HAL_GPIO_WritePin(GYRO_CS_GPIOC
```

Gyroscope Chip Select macro definition.

Definition at line **467** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_Read()**, and **GYRO\_IO\_Write()**.

```
#define GYRO_CS_PIN GPIO_PIN_7 /* PD.07 */
```

Definition at line **476** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define GYRO_INT1_EXTI_IRQn  EXTI2_IRQn
```

Definition at line **486** of file **stm32l476g\_discovery.h**.

```
#define GYRO_INT1_GPIO_CLK_DISABLE ( )  __HAL_RCC_GPIOI
```

Definition at line **484** of file **stm32l476g\_discovery.h**.

```
#define GYRO_INT1_GPIO_CLK_ENABLE ( )  __HAL_RCC_GPIOD
```

Definition at line **483** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define GYRO_INT1_GPIO_PORT  GPIOD /* GPIOD */
```

Gyroscope Interrupt pins.

Definition at line **482** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define GYRO_INT1_PIN  GPIO_PIN_2 /* PD.02 */
```

Definition at line **485** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define GYRO_INT2_EXTI_IRQn  EXTI9_5_IRQn
```

Definition at line **491** of file **stm32l476g\_discovery.h**.

```
#define GYRO_INT2_GPIO_CLK_DISABLE ( )  __HAL_RCC_GPIOE
```

Definition at line **489** of file **stm32l476g\_discovery.h**.

```
#define GYRO_INT2_GPIO_CLK_ENABLE ( )  __HAL_RCC_GPIOE
```

Definition at line **488** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define GYRO_INT2_GPIO_PORT  GPIOB /* GPIOB */
```

Definition at line **487** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define GYRO_INT2_PIN  GPIO_PIN_8 /* PB.08 */
```

Definition at line **490** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_DeInit()**, and **GYRO\_IO\_Init()**.

```
#define IDD_I2C_ADDRESS  ((uint16_t) 0x84)
```

Definition at line **384** of file **stm32l476g\_discovery.h**.

Referenced by **BSP\_IDD\_ClearIT()**, **BSP\_IDD\_Config()**,  
**BSP\_IDD\_DeInit()**, **BSP\_IDD\_DisableIT()**, **BSP\_IDD\_EnableIT()**,

**BSP\_IDD\_ErrorClearIT()**, **BSP\_IDD\_ErrorDisableIT()**,  
**BSP\_IDD\_ErrorEnableIT()**, **BSP\_IDD\_ErrorGetCode()**,  
**BSP\_IDD\_ErrorGetITStatus()**, **BSP\_IDD\_GetITStatus()**,  
**BSP\_IDD\_GetValue()**, **BSP\_IDD\_Init()**, **BSP\_IDD\_LowPower()**,  
**BSP\_IDD\_Reset()**, **BSP\_IDD\_StartMeasure()**, and  
**BSP\_IDD\_WakeUp()**.

**#define IDD\_INT\_EXTI\_IRQn** EXTI15\_10\_IRQn

Definition at line **513** of file **stm32l476g\_discovery.h**.

Referenced by **MFX\_IO\_DeInit()**, and **MFX\_IO\_ITConfig()**.

**#define IDD\_INT\_GPIO\_CLK\_DISABLE ( )** \_\_HAL\_RCC\_GPIOC\_C

Definition at line **511** of file **stm32l476g\_discovery.h**.

**#define IDD\_INT\_GPIO\_CLK\_ENABLE ( )** \_\_HAL\_RCC\_GPIOC\_C

Definition at line **510** of file **stm32l476g\_discovery.h**.

Referenced by **MFX\_IO\_DeInit()**, and **MFX\_IO\_ITConfig()**.

**#define IDD\_INT\_GPIO\_PORT** GPIOC /\* GPIOC \*/

Idd current measurement interface pins.

Definition at line **509** of file **stm32l476g\_discovery.h**.

Referenced by **MFX\_IO\_DeInit()**, and **MFX\_IO\_ITConfig()**.

**#define IDD\_INT\_PIN** GPIO\_PIN\_13 /\* PC.13 \*/

Definition at line **512** of file **stm32l476g\_discovery.h**.

Referenced by **MFX\_IO\_DeInit()**, and **MFX\_IO\_ITConfig()**.

```
#define IDD_WAKEUP_GPIO_CLK_DISABLE ( ) __HAL_RCC_GPIO
```

Definition at line **517** of file **stm32l476g\_discovery.h**.

```
#define IDD_WAKEUP_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIO
```

Definition at line **516** of file **stm32l476g\_discovery.h**.

Referenced by **MFX\_IO\_DeInit()**, and **MFX\_IO\_EnableWakeupPin()**.

```
#define IDD_WAKEUP_GPIO_PORT GPIOA /* GPIOA */
```

Definition at line **515** of file **stm32l476g\_discovery.h**.

Referenced by **MFX\_IO\_DeInit()**, **MFX\_IO\_EnableWakeupPin()**, and **MFX\_IO\_Wakeup()**.

```
#define IDD_WAKEUP_PIN GPIO_PIN_4 /* PA.04 */
```

Definition at line **518** of file **stm32l476g\_discovery.h**.

Referenced by **MFX\_IO\_DeInit()**, **MFX\_IO\_EnableWakeupPin()**, and **MFX\_IO\_Wakeup()**.

```
#define MAGNETO_CS_GPIO_CLK_DISABLE ( ) __HAL_RCC_GP
```

Definition at line **430** of file **stm32l476g\_discovery.h**.

```
#define MAGNETO_CS_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPI
```

Definition at line **429** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_Init()**, **MAGNETO\_IO\_DeInit()**, and **MAGNETO\_IO\_Init()**.

```
#define MAGNETO_CS_GPIO_PORT GPIOC /* GPIOC */
```

Magnetometer SPI Interface pins.

Definition at line **428** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_Init()**, **MAGNETO\_IO\_DeInit()**, and **MAGNETO\_IO\_Init()**.

```
#define MAGNETO_CS_HIGH ( ) HAL_GPIO_WritePin(MAGNETO_
```

Definition at line **423** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_Init()**, **MAGNETO\_IO\_DeInit()**, **MAGNETO\_IO\_Init()**, **MAGNETO\_IO\_Read()**, and **MAGNETO\_IO\_Write()**.

```
#define MAGNETO_CS_LOW ( ) HAL_GPIO_WritePin(MAGNETO_
```

Magnetometer Chip Select macro definition.

Definition at line **422** of file **stm32l476g\_discovery.h**.

Referenced by **MAGNETO\_IO\_Read()**, and **MAGNETO\_IO\_Write()**.

```
#define MAGNETO_CS_PIN GPIO_PIN_0 /* PC.00 */
```

Definition at line **431** of file **stm32l476g\_discovery.h**.

Referenced by **GYRO\_IO\_Init()**, **MAGNETO\_IO\_DeInit()**, and **MAGNETO\_IO\_Init()**.

```
#define MAGNETO_DRDY_GPIO_CLK_DISABLE ( ) __HAL_RCC_
```

Definition at line **445** of file **stm32l476g\_discovery.h**.

```
#define MAGNETO_DRDY_GPIO_CLK_ENABLE ( ) __HAL_RCC_ (
```

Definition at line **444** of file **stm32l476g\_discovery.h**.

```
#define MAGNETO_DRDY_GPIO_PORT GPIOC /* GPIOC */
```

Definition at line **443** of file **stm32l476g\_discovery.h**.

```
#define MAGNETO_DRDY_PIN GPIO_PIN_2 /* PC.01 */
```

Definition at line **446** of file **stm32l476g\_discovery.h**.

Referenced by **MAGNETO\_IO\_DeInit()**.

```
#define MAGNETO_INT1_EXTI_IRQn EXTI1_IRQn
```

Definition at line **441** of file **stm32l476g\_discovery.h**.

```
#define MAGNETO_INT1_PIN GPIO_PIN_1 /* PC.01 */
```

Definition at line **440** of file **stm32l476g\_discovery.h**.



Referenced by [MAGNETO\\_IO\\_DeInit\(\)](#).

```
#define MAGNETO_INT_GPIO_CLK_DISABLE ( ) __HAL_RCC_GP
```

Definition at line [439](#) of file [stm32l476g\\_discovery.h](#).

```
#define MAGNETO_INT_GPIO_CLK_ENABLE ( ) __HAL_RCC_GP
```

Definition at line [438](#) of file [stm32l476g\\_discovery.h](#).

```
#define MAGNETO_INT_GPIO_PORT GPIOC /* GPIOC */
```

Magnetometer Interrupt pins.

Definition at line [437](#) of file [stm32l476g\\_discovery.h](#).

```
#define MULTIPLEBYTE_CMD ((uint8_t)0x40)
```

Definition at line [279](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [GYRO\\_IO\\_Read\(\)](#), and [GYRO\\_IO\\_Write\(\)](#).

```
#define READWRITE_CMD ((uint8_t)0x80)
```

Definition at line [277](#) of file [stm32l476g\\_discovery.h](#).

Referenced by [GYRO\\_IO\\_Read\(\)](#).

```
#define SPIx_TIMEOUT_MAX ((uint32_t)0x1000)
```

Definition at line [275](#) of file [stm32l476g\\_discovery.h](#).

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Enumerations](#)

## Exported Constants

[STM32L476G-DISCOVERY ACCELEROMETER](#)

## Enumerations

```
enum ACCELERO_StatusTypeDef { ACCELERO_OK = 0,  
                               ACCELERO_ERROR = 1, ACCELERO_TIMEOUT = 2 }
```

## Enumeration Type Documentation

enum **ACCELERO\_StatusTypeDef**

**Enumerator:**

*ACCELERO\_OK*

*ACCELERO\_ERROR*

*ACCELERO\_TIMEOUT*

Definition at line **75** of file **stm32l476g\_discovery\_accelerometer.h**.

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Bus Operations Functions

[STM32L476G-DISCOVERY Common](#)

## Functions

static void	<b>SPIx_Init</b> (void) SPIx Bus initialization.
static void	<b>SPIx_Msplnit</b> (SPI_HandleTypeDef *hspi) SPI MSP Init.
static void	<b>SPIx_DeInit</b> (void) SPIx Bus Deinitialization.
static void	<b>SPIx_MspDeInit</b> (void) SPI MSP DeInit.
static uint8_t	<b>SPIx_WriteRead</b> (uint8_t Byte) Sends a Byte through the SPI interface and return the Byte received from the SPI bus.
static void	<b>SPIx_Write</b> (uint8_t Byte) Sends a Byte through the SPI interface.
static uint8_t	<b>SPIx_Read</b> (void) Receives a Byte from the SPI bus.
static void	<b>I2C1_Init</b> (void) Discovery I2C1 Bus initialization.
static void	<b>I2C1_Msplnit</b> (I2C_HandleTypeDef *hi2c) Discovery I2C1 MSP Initialization.
static void	<b>I2C1_DeInit</b> (void) Discovery I2C1 Bus Deitailization.
static void	<b>I2C1_MspDeInit</b> (I2C_HandleTypeDef *hi2c) Discovery I2C1 MSP Deinitialization.
static HAL_StatusTypeDef	<b>I2C1_WriteBuffer</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length) Write a value in a register of the device through BUS.

static HAL_StatusTypeDef	<b>I2C1_ReadBuffer</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length) Reads multiple data on the BUS.
static void	<b>I2C1_Error</b> (void) Discovery I2C1 error treatment function.
static void	<b>I2C2_Init</b> (void) Discovery I2C2 Bus initialization.
static void	<b>I2C2_MspltInit</b> (I2C_HandleTypeDef *hi2c) Discovery I2C2 MSP Initialization.
static void	<b>I2C2_DeInit</b> (void) Discovery I2C2 Bus Deinitialization.
static void	<b>I2C2_MspDeInit</b> (I2C_HandleTypeDef *hi2c) Discovery I2C2 MSP DeInitialization.
static void	<b>I2C2_WriteData</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t Value) Write a value in a register of the device through BUS.
static HAL_StatusTypeDef	<b>I2C2_WriteBuffer</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length) Write a value in a register of the device through BUS.
static uint8_t	<b>I2C2_ReadData</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize) Read a register of the device through BUS.
static HAL_StatusTypeDef	<b>I2C2_ReadBuffer</b> (uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length) Reads multiple data on the BUS.
static void	<b>I2C2_Error</b> (void) Discovery I2C2 error treatment function.



void **ACCELERO\_IO\_Init** (void)  
Configures  
COMPASS/ACCELEROMETER io  
interface.

void **ACCELERO\_IO\_DeInit** (void)  
De-Configures  
COMPASS/ACCELEROMETER io  
interface.

void **ACCELERO\_IO\_ITConfig** (void)  
Configures COMPASS / ACCELERO  
click IT.

void **ACCELERO\_IO\_Write** (uint8\_t  
RegisterAddr, uint8\_t Value)  
Writes one byte to the COMPASS /  
ACCELEROMETER.

uint8\_t **ACCELERO\_IO\_Read** (uint8\_t  
RegisterAddr)  
Reads a block of data from the  
COMPASS / ACCELEROMETER.

void **MAGNETO\_IO\_Init** (void)  
Configures COMPASS/MAGNETO SPI  
interface.

void **MAGNETO\_IO\_DeInit** (void)  
de-Configures COMPASS/MAGNETO  
SPI interface.

void **MAGNETO\_IO\_Write** (uint8\_t  
RegisterAddr, uint8\_t Value)  
Writes one byte to the  
COMPASS/MAGNETO.

uint8\_t **MAGNETO\_IO\_Read** (uint8\_t  
RegisterAddr)  
Reads a block of data from the  
COMPASS/MAGNETO.

void **GYRO\_IO\_Init** (void)  
Configures the GYRO SPI interface.

	void	<b>GYRO_IO_DeInit</b> (void)	de-Configures GYRO SPI interface.
	void	<b>GYRO_IO_Write</b> (uint8_t *pBuffer, uint8_t WriteAddr, uint16_t NumByteToWrite)	Writes one byte to the GYRO.
	void	<b>GYRO_IO_Read</b> (uint8_t *pBuffer, uint8_t ReadAddr, uint16_t NumByteToRead)	Reads a block of data from the GYROSCOPE.
	void	<b>MFX_IO_Init</b> (void)	Initializes MFX low level.
	void	<b>MFX_IO_DeInit</b> (void)	Deinitializes MFX low level.
	void	<b>MFX_IO_ITConfig</b> (void)	Configures MFX low level interrupt.
	void	<b>MFX_IO_EnableWakeupPin</b> (void)	Configures MFX wke up pin.
	void	<b>MFX_IO_Wakeup</b> (void)	Wakeup MFX.
	void	<b>MFX_IO_Write</b> (uint16_t Addr, uint8_t Reg, uint8_t Value)	MFX writes single data.
uint8_t		<b>MFX_IO_Read</b> (uint16_t Addr, uint8_t Reg)	MFX reads single data.
uint16_t		<b>MFX_IO_ReadMultiple</b> (uint16_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length)	MFX reads multiple data.
	void	<b>MFX_IO_WriteMultiple</b> (uint16_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length)	MFX writes multiple data.

void **MFX\_IO\_Delay** (uint32\_t Delay)  
MFX delay.

void **AUDIO\_IO\_Init** (void)  
Initializes Audio low level.

void **AUDIO\_IO\_DeInit** (void)  
Deinitializes Audio low level.

void **AUDIO\_IO\_Write** (uint8\_t Addr, uint8\_t  
Reg, uint8\_t Value)  
Writes a single data.

uint8\_t **AUDIO\_IO\_Read** (uint8\_t Addr, uint8\_t  
Reg)  
Reads a single data.

void **AUDIO\_IO\_Delay** (uint32\_t Delay)  
AUDIO Codec delay.

## Function Documentation

**void ACCELERO\_IO\_DeInit ( void )**

De-Configures COMPASS/ACCELEROMETER io interface.

**Return values:**

**None**

Definition at line **1188** of file **stm32l476g\_discovery.c**.

References **ACCELERO\_CS\_GPIO\_CLK\_ENABLE**, **ACCELERO\_CS\_GPIO\_PORT**, **ACCELERO\_CS\_HIGH**, **ACCELERO\_CS\_PIN**, and **SPIx\_DeInit()**.

Referenced by **BSP\_COMPASS\_DeInit()**.

**void ACCELERO\_IO\_Init ( void )**

Configures COMPASS/ACCELEROMETER io interface.

**Return values:**

**None**

Definition at line **1166** of file **stm32l476g\_discovery.c**.

References **ACCELERO\_CS\_GPIO\_CLK\_ENABLE**, **ACCELERO\_CS\_GPIO\_PORT**, **ACCELERO\_CS\_HIGH**, **ACCELERO\_CS\_PIN**, and **SPIx\_Init()**.

**void ACCELERO\_IO\_ITConfig ( void )**

Configures COMPASS / ACCELERO click IT.

**Return values:**

**None**

Definition at line **1211** of file **stm32l476g\_discovery.c**.

**uint8\_t ACCELERO\_IO\_Read ( uint8\_t RegisterAddr )**

Reads a block of data from the COMPASS / ACCELEROMETER.

**Parameters:**

**RegisterAddr** : specifies the COMPASS / ACCELEROMETER internal address register to read from

**Return values:**

**ACCELEROMETER** register value

Definition at line **1236** of file **stm32l476g\_discovery.c**.

References **\_\_SPI\_DIRECTION\_1LINE\_RX**, **\_\_SPI\_DIRECTION\_1LINE\_TX**, **ACCELERO\_CS\_HIGH**, **ACCELERO\_CS\_LOW**, **SpiHandle**, **SPIx\_Read()**, and **SPIx\_Write()**.

**void ACCELERO\_IO\_Write ( uint8\_t RegisterAddr,  
uint8\_t Value  
)**

Writes one byte to the COMPASS / ACCELEROMETER.

**Parameters:**

**RegisterAddr** specifies the COMPASS / ACCELEROMETER register to be written.

**Value** : Data to be written

**Return values:**

**None**

Definition at line **1221** of file **stm32l476g\_discovery.c**.

References **\_\_SPI\_DIRECTION\_1LINE\_TX**, **ACCELERO\_CS\_HIGH**, **ACCELERO\_CS\_LOW**, **SpiHandle**, and **SPIx\_Write()**.

**void AUDIO\_IO\_DeInit ( void )**

Deinitializes Audio low level.

**Return values:**

**None**

Definition at line **1692** of file **stm32l476g\_discovery.c**.

References **I2C1\_DeInit()**.

**void AUDIO\_IO\_Delay ( uint32\_t Delay )**

AUDIO Codec delay.

**Parameters:**

**Delay,:** Delay in ms

**Return values:**

**None**

Definition at line **1748** of file **stm32l476g\_discovery.c**.

**void AUDIO\_IO\_Init ( void )**

Initializes Audio low level.

**Return values:**

**None**

Definition at line **1658** of file **stm32l476g\_discovery.c**.

References **AUDIO\_RESET\_GPIO**,  
**AUDIO\_RESET\_GPIO\_CLK\_ENABLE**, **AUDIO\_RESET\_PIN**,  
**CODEC\_AUDIO\_POWER\_OFF**, **CODEC\_AUDIO\_POWER\_ON**, and  
**I2C1\_Init()**.

```
uint8_t AUDIO_IO_Read ( uint8_t Addr,  
                        uint8_t Reg  
                        )
```

Reads a single data.

**Parameters:**

**Addr,:** I2C address

**Reg,:** Reg address

**Return values:**

**Data** to be read

Definition at line **1734** of file **stm32l476g\_discovery.c**.

References **I2C1\_ReadBuffer()**.

```
void AUDIO_IO_Write ( uint8_t Addr,  
                     uint8_t Reg,  
                     uint8_t Value  
                     )
```

Writes a single data.

**Parameters:**

**Addr,:** I2C address

**Reg,:** Reg address

**Value,:** Data to be written

**Return values:**

**None**

Definition at line **1723** of file **stm32l476g\_discovery.c**.

References **I2C1\_WriteBuffer()**.

**void GYRO\_IO\_DeInit ( void )**

de-Configures GYRO SPI interface.

**Return values:**

**None**

Definition at line **1398** of file **stm32l476g\_discovery.c**.

References **GYRO\_CS\_GPIO\_CLK\_ENABLE**,  
**GYRO\_CS\_GPIO\_PORT**, **GYRO\_CS\_HIGH**, **GYRO\_CS\_PIN**,  
**GYRO\_INT1\_GPIO\_CLK\_ENABLE**, **GYRO\_INT1\_GPIO\_PORT**,  
**GYRO\_INT1\_PIN**, **GYRO\_INT2\_GPIO\_CLK\_ENABLE**,  
**GYRO\_INT2\_GPIO\_PORT**, **GYRO\_INT2\_PIN**, and **SPIx\_DeInit()**.

Referenced by **BSP\_GYRO\_DeInit()**.

**void GYRO\_IO\_Init ( void )**

Configures the GYRO SPI interface.

**Return values:**

**None**



Definition at line **1334** of file **stm32l476g\_discovery.c**.

References **ACCELERO\_CS\_GPIO\_CLK\_ENABLE**,  
**ACCELERO\_CS\_GPIO\_PORT**, **ACCELERO\_CS\_HIGH**,  
**ACCELERO\_CS\_PIN**, **GYRO\_CS\_GPIO\_CLK\_ENABLE**,  
**GYRO\_CS\_GPIO\_PORT**, **GYRO\_CS\_HIGH**, **GYRO\_CS\_PIN**,  
**GYRO\_INT1\_GPIO\_CLK\_ENABLE**, **GYRO\_INT1\_GPIO\_PORT**,  
**GYRO\_INT1\_PIN**, **GYRO\_INT2\_GPIO\_CLK\_ENABLE**,  
**GYRO\_INT2\_GPIO\_PORT**, **GYRO\_INT2\_PIN**,  
**MAGNETO\_CS\_GPIO\_CLK\_ENABLE**,  
**MAGNETO\_CS\_GPIO\_PORT**, **MAGNETO\_CS\_HIGH**,  
**MAGNETO\_CS\_PIN**, and **SPIx\_Init()**.

```
void GYRO_IO_Read ( uint8_t * pBuffer,  
                    uint8_t  ReadAddr,  
                    uint16_t NumByteToRead  
                    )
```

Reads a block of data from the GYROSCOPE.

**Parameters:**

<b>pBuffer</b>	: pointer to the buffer that receives the data read from the GYROSCOPE.
<b>ReadAddr</b>	: GYROSCOPE's internal address to read from.
<b>NumByteToRead</b>	: number of bytes to read from the GYROSCOPE.

**Return values:**

**None**

Definition at line **1467** of file **stm32l476g\_discovery.c**.

References **\_\_SPI\_DIRECTION\_2LINES**, **GYRO\_CS\_HIGH**,  
**GYRO\_CS\_LOW**, **MULTIPLEBYTE\_CMD**, **READWRITE\_CMD**,

**SpiHandle**, and **SPIx\_WriteRead()**.

```
void GYRO_IO_Write ( uint8_t * pBuffer,  
                    uint8_t  WriteAddr,  
                    uint16_t NumByteToWrite  
                    )
```

Writes one byte to the GYRO.

**Parameters:**

**pBuffer** : pointer to the buffer containing the data to be written to the GYRO.  
**WriteAddr** : GYRO's internal address to write to.  
**NumByteToWrite,:** Number of bytes to write.

**Return values:**

**None**

Definition at line **1431** of file **stm32l476g\_discovery.c**.

References **\_\_SPI\_DIRECTION\_2LINES**, **GYRO\_CS\_HIGH**, **GYRO\_CS\_LOW**, **MULTIPLEBYTE\_CMD**, **SpiHandle**, and **SPIx\_WriteRead()**.

```
static void I2C1_DeInit ( void ) [static]
```

Discovery I2C1 Bus Deinitialization.

**Return values:**

**None**

Definition at line **825** of file **stm32l476g\_discovery.c**.

References **I2C1\_MspDeInit()**, and **I2c1Handle**.

Referenced by [AUDIO\\_IO\\_DeInit\(\)](#).

**static void [I2C1\\_Error](#) ( void ) [static]**

Discovery I2C1 error treatment function.

**Return values:**

**None**

Definition at line [921](#) of file [stm32l476g\\_discovery.c](#).

References [I2C1\\_Init\(\)](#), and [I2c1Handle](#).

Referenced by [I2C1\\_ReadBuffer\(\)](#), and [I2C1\\_WriteBuffer\(\)](#).

**static void [I2C1\\_Init](#) ( void ) [static]**

Discovery I2C1 Bus initialization.

**Return values:**

**None**

Definition at line [750](#) of file [stm32l476g\\_discovery.c](#).

References [DISCOVERY\\_I2C1](#), [I2C1\\_MspInit\(\)](#), and [I2c1Handle](#).

Referenced by [AUDIO\\_IO\\_Init\(\)](#), and [I2C1\\_Error\(\)](#).

**static void [I2C1\\_MspDeInit](#) ( I2C\_HandleTypeDef \* [hi2c](#) ) [static]**

Discovery I2C1 MSP Deinitialization.

**Parameters:**

**[hi2c](#),:** I2C handle

**Return values:**

**None**

Definition at line **840** of file `stm32l476g_discovery.c`.

References `DISCOVERY_I2C1`, `DISCOVERY_I2C1_CLK_DISABLE`, `DISCOVERY_I2C1_ER_IRQn`, `DISCOVERY_I2C1_EV_IRQn`, `DISCOVERY_I2C1_FORCE_RESET`, `DISCOVERY_I2C1_RELEASE_RESET`, `DISCOVERY_I2C1_SCL_GPIO_CLK_ENABLE`, `DISCOVERY_I2C1_SCL_GPIO_PORT`, `DISCOVERY_I2C1_SCL_PIN`, `DISCOVERY_I2C1_SDA_GPIO_CLK_ENABLE`, and `DISCOVERY_I2C1_SDA_PIN`.

Referenced by `I2C1_DeInit()`.

**static void** `I2C1_Mspltinit` (`I2C_HandleTypeDef` \* **hi2c**) [`static`]

Discovery I2C1 MSP Initialization.

**Parameters:**

**hi2c**,: I2C handle

**Return values:**

**None**

Definition at line **774** of file `stm32l476g_discovery.c`.

References `DISCOVERY_I2C1`, `DISCOVERY_I2C1_CLK_ENABLE`, `DISCOVERY_I2C1_ER_IRQn`, `DISCOVERY_I2C1_EV_IRQn`, `DISCOVERY_I2C1_FORCE_RESET`, `DISCOVERY_I2C1_RELEASE_RESET`, `DISCOVERY_I2C1_SCL_GPIO_CLK_ENABLE`, `DISCOVERY_I2C1_SCL_GPIO_PORT`, `DISCOVERY_I2C1_SCL_PIN`, `DISCOVERY_I2C1_SCL_SDA_AF`,

**DISCOVERY\_I2C1\_SDA\_GPIO\_CLK\_ENABLE**, and **DISCOVERY\_I2C1\_SDA\_PIN**.

Referenced by **I2C1\_Init()**.

```
static HAL_StatusTypeDef I2C1_ReadBuffer ( uint16_t Addr,
                                           uint16_t Reg,
                                           uint16_t RegSize,
                                           uint8_t * pBuffer,
                                           uint16_t Length
                                           ) [static]
```

Reads multiple data on the BUS.

### Parameters:

**Addr,:** I2C Address  
**Reg,:** Reg Address  
**RegSize** : The target register size (can be 8BIT or 16BIT)  
**pBuffer,:** pointer to read data buffer  
**Length,:** length of the data

### Return values:

**0** if no problems to read multiple data

Definition at line 902 of file stm32l476g\_discovery.c.

References [I2C1\\_Error\(\)](#), [I2c1Handle](#), and [I2c1Timeout](#).

Referenced by **AUDIO\_IO\_Read()**.

[illegible]

**uint16\_t Length**  
**) [static]**

Write a value in a register of the device through BUS.

**Parameters:**

**Addr,:** Device address on BUS Bus.  
**Reg,:** The target register address to write  
**RegSize,:** The target register size (can be 8BIT or 16BIT)  
**pBuffer,:** The target register value to be written  
**Length,:** buffer size to be written

**Return values:**

**None**

Definition at line **878** of file **stm32l476g\_discovery.c**.

References **I2C1\_Error()**, **I2c1Handle**, and **I2c1Timeout**.

Referenced by **AUDIO\_IO\_Write()**.

**static void I2C2\_DeInit ( void ) [static]**

Discovery I2C2 Bus Deinitialization.

**Return values:**

**None**

Definition at line **1005** of file **stm32l476g\_discovery.c**.

References **I2C2\_MspDeInit()**, and **I2c2Handle**.

Referenced by **MFX\_IO\_DeInit()**.

**static void I2C2\_Error ( void ) [static]**

Discovery I2C2 error treatment function.

**Return values:**

**None**

Definition at line **1146** of file **stm32l476g\_discovery.c**.

References **I2C2\_Init()**, and **I2c2Handle**.

Referenced by **I2C2\_ReadBuffer()**, **I2C2\_ReadData()**, **I2C2\_WriteBuffer()**, and **I2C2\_WriteData()**.

**static void I2C2\_Init ( void ) [static]**

Discovery I2C2 Bus initialization.

**Return values:**

**None**

Definition at line **934** of file **stm32l476g\_discovery.c**.

References **DISCOVERY\_I2C2**, **I2C2\_MspInit()**, and **I2c2Handle**.

Referenced by **I2C2\_Error()**, and **MFX\_IO\_Init()**.

**static void I2C2\_MspDeInit ( I2C\_HandleTypeDef \* hi2c ) [static]**

Discovery I2C2 MSP DeInitialization.

**Parameters:**

**hi2c,:** I2C2 handle

**Return values:**

**None**

Definition at line **1020** of file **stm32l476g\_discovery.c**.

References **DISCOVERY\_I2C2**, **DISCOVERY\_I2C2\_CLK\_DISABLE**, **DISCOVERY\_I2C2\_ER\_IRQn**, **DISCOVERY\_I2C2\_EV\_IRQn**, **DISCOVERY\_I2C2\_FORCE\_RESET**, **DISCOVERY\_I2C2\_RELEASE\_RESET**, **DISCOVERY\_I2C2\_SCL\_GPIO\_CLK\_ENABLE**, **DISCOVERY\_I2C2\_SCL\_GPIO\_PORT**, **DISCOVERY\_I2C2\_SCL\_PIN**, **DISCOVERY\_I2C2\_SDA\_GPIO\_CLK\_ENABLE**, and **DISCOVERY\_I2C2\_SDA\_PIN**.

Referenced by **I2C2\_DeInit()**.

**static void I2C2\_MspInit ( I2C\_HandleTypeDef \* **hi2c** ) [static]**

Discovery I2C2 MSP Initialization.

**Parameters:**

**hi2c,:** I2C2 handle

**Return values:**

**None**

Definition at line **958** of file **stm32l476g\_discovery.c**.

References **DISCOVERY\_I2C2**, **DISCOVERY\_I2C2\_CLK\_ENABLE**, **DISCOVERY\_I2C2\_ER\_IRQn**, **DISCOVERY\_I2C2\_EV\_IRQn**, **DISCOVERY\_I2C2\_FORCE\_RESET**, **DISCOVERY\_I2C2\_RELEASE\_RESET**, **DISCOVERY\_I2C2\_SCL\_GPIO\_CLK\_ENABLE**, **DISCOVERY\_I2C2\_SCL\_GPIO\_PORT**, **DISCOVERY\_I2C2\_SCL\_PIN**, **DISCOVERY\_I2C2\_SCL\_SDA\_AF**, **DISCOVERY\_I2C2\_SDA\_GPIO\_CLK\_ENABLE**, and **DISCOVERY\_I2C2\_SDA\_PIN**.

Referenced by **I2C2\_Init()**.



```
static HAL_StatusTypeDef I2C2_ReadBuffer ( uint16_t Addr,
                                           uint16_t Reg,
                                           uint16_t RegSize,
                                           uint8_t * pBuffer,
                                           uint16_t Length
                                           ) [static]
```

Reads multiple data on the BUS.

**Parameters:**

**Addr,:** I2C Address

**Reg,:** Reg Address

**RegSize** : The target register size (can be 8BIT or 16BIT)

**pBuffer,:** pointer to read data buffer

**Length,:** length of the data

**Return values:**

**0** if no problems to read multiple data

Definition at line [1126](#) of file [stm32l476g\\_discovery.c](#).

References [I2C2\\_Error\(\)](#), [I2c2Handle](#), and [I2c2Timeout](#).

Referenced by [MFX\\_IO\\_ReadMultiple\(\)](#).

```
static uint8_t I2C2_ReadData ( uint16_t Addr,
                               uint16_t Reg,
                               uint16_t RegSize
                               ) [static]
```

Read a register of the device through BUS.

**Parameters:**

**Addr,:** Device address on BUS  
**Reg,:** The target register address to read  
**RegSize,:** The target register size (can be 8BIT or 16BIT)

**Return values:**

**read** register value

Definition at line **1100** of file **stm32l476g\_discovery.c**.

References **I2C2\_Error()**, **I2c2Handle**, and **I2c2Timeout**.

Referenced by **MFX\_IO\_Read()**.

```
static HAL_StatusTypeDef I2C2_WriteBuffer ( uint16_t Addr,  
                                             uint16_t Reg,  
                                             uint16_t RegSize,  
                                             uint8_t * pBuffer,  
                                             uint16_t Length  
                                             ) [static]
```

Write a value in a register of the device through BUS.

**Parameters:**

**Addr,:** Device address on BUS Bus.  
**Reg,:** The target register address to write  
**RegSize,:** The target register size (can be 8BIT or 16BIT)  
**pBuffer,:** The target register value to be written  
**Length,:** buffer size to be written

**Return values:**

**None**

Definition at line **1077** of file **stm32l476g\_discovery.c**.

References [I2C2\\_Error\(\)](#), [I2c2Handle](#), and [I2c2Timeout](#).

Referenced by [MFX\\_IO\\_WriteMultiple\(\)](#).

```
static void I2C2\_WriteData ( uint16_t Addr,  
                             uint16_t Reg,  
                             uint16_t RegSize,  
                             uint8_t  Value  
                             )           [static]
```

Write a value in a register of the device through BUS.

**Parameters:**

**Addr,:** Device address on BUS Bus.  
**Reg,:** The target register address to write  
**RegSize,:** The target register size (can be 8BIT or 16BIT)  
**Value,:** The target register value to be written

**Return values:**

**None**

Definition at line [1054](#) of file [stm32l476g\\_discovery.c](#).

References [I2C2\\_Error\(\)](#), [I2c2Handle](#), and [I2c2Timeout](#).

Referenced by [MFX\\_IO\\_Write\(\)](#).

```
void MAGNETO\_IO\_DeInit ( void )
```

de-Configures COMPASS/MAGNETO SPI interface.

**Return values:**

**None**

Definition at line **1275** of file **stm32l476g\_discovery.c**.

References **MAGNETO\_CS\_GPIO\_CLK\_ENABLE**,  
**MAGNETO\_CS\_GPIO\_PORT**, **MAGNETO\_CS\_HIGH**,  
**MAGNETO\_CS\_PIN**, **MAGNETO\_DRDY\_PIN**,  
**MAGNETO\_INT1\_PIN**, and **SPIx\_DeInit()**.

Referenced by **BSP\_COMPASS\_DeInit()**.

**void MAGNETO\_IO\_Init ( void )**

Configures COMPASS/MAGNETO SPI interface.

**Return values:**

**None**

Definition at line **1253** of file **stm32l476g\_discovery.c**.

References **MAGNETO\_CS\_GPIO\_CLK\_ENABLE**,  
**MAGNETO\_CS\_GPIO\_PORT**, **MAGNETO\_CS\_HIGH**,  
**MAGNETO\_CS\_PIN**, and **SPIx\_Init()**.

**uint8\_t MAGNETO\_IO\_Read ( uint8\_t RegisterAddr )**

Reads a block of data from the COMPASS/MAGNETO.

**Parameters:**

**RegisterAddr** : specifies the COMPASS/MAGNETO internal  
address register to read from

**Return values:**

**ACCELEROMETER** register value

Definition at line **1318** of file **stm32l476g\_discovery.c**.

References **\_\_SPI\_DIRECTION\_1LINE\_RX**,

`__SPI_DIRECTION_1LINE_TX`, `MAGNETO_CS_HIGH`, `MAGNETO_CS_LOW`, `SpiHandle`, `SPIx_Read()`, and `SPIx_Write()`.

```
void MAGNETO_IO_Write ( uint8_t RegisterAddr,  
                        uint8_t Value  
                        )
```

Writes one byte to the COMPASS/MAGNETO.

**Parameters:**

**RegisterAddr** specifies the COMPASS/MAGNETO register to be written.

**Value** : Data to be written

**Return values:**

**None**

Definition at line **1303** of file `stm32l476g_discovery.c`.

References `__SPI_DIRECTION_1LINE_TX`, `MAGNETO_CS_HIGH`, `MAGNETO_CS_LOW`, `SpiHandle`, and `SPIx_Write()`.

```
void MFX_IO_DeInit ( void )
```

Deinitializes MFX low level.

**Return values:**

**None**

Definition at line **1512** of file `stm32l476g_discovery.c`.

References `I2C2_DeInit()`, `IDD_INT_EXTI_IRQn`, `IDD_INT_GPIO_CLK_ENABLE`, `IDD_INT_GPIO_PORT`, `IDD_INT_PIN`, `IDD_WAKEUP_GPIO_CLK_ENABLE`, `IDD_WAKEUP_GPIO_PORT`, and `IDD_WAKEUP_PIN`.

**void MFX\_IO\_Delay ( uint32\_t Delay )**

MFX delay.

**Parameters:**

**Delay,:** Delay in ms

**Return values:**

**None**

Definition at line **1647** of file **stm32l476g\_discovery.c**.

**void MFX\_IO\_EnableWakeupPin ( void )**

Configures MFX wke up pin.

**Return values:**

**None**

Definition at line **1562** of file **stm32l476g\_discovery.c**.

References **IDD\_WAKEUP\_GPIO\_CLK\_ENABLE**,  
**IDD\_WAKEUP\_GPIO\_PORT**, and **IDD\_WAKEUP\_PIN**.

**void MFX\_IO\_Init ( void )**

Initializes MFX low level.

**Return values:**

**None**

Definition at line **1503** of file **stm32l476g\_discovery.c**.

References **I2C2\_Init()**.

**void MFX\_IO\_ITConfig ( void )**

Configures MFX low level interrupt.

**Return values:**

**None**

Definition at line **1539** of file **stm32l476g\_discovery.c**.

References **IDD\_INT\_EXTI\_IRQn**, **IDD\_INT\_GPIO\_CLK\_ENABLE**, **IDD\_INT\_GPIO\_PORT**, and **IDD\_INT\_PIN**.

**uint8\_t MFX\_IO\_Read ( uint16\_t Addr,  
uint8\_t Reg  
)**

MFX reads single data.

**Parameters:**

**Addr,:** I2C address

**Reg,:** Register address

**Return values:**

**Read** data

Definition at line **1611** of file **stm32l476g\_discovery.c**.

References **I2C2\_ReadData()**.

**uint16\_t MFX\_IO\_ReadMultiple ( uint16\_t Addr,  
uint8\_t Reg,  
uint8\_t \* Buffer,**

**uint16\_t Length**  
)

MFX reads multiple data.

**Parameters:**

**Addr,:** I2C address  
**Reg,:** Register address  
**Buffer,:** Pointer to data buffer  
**Length,:** Length of the data

**Return values:**

**Number** of read data

Definition at line **1624** of file **stm32l476g\_discovery.c**.

References **I2C2\_ReadBuffer()**.

**void MFX\_IO\_Wakeup ( void )**

Wakeup MFX.

**Return values:**

**None**

Definition at line **1581** of file **stm32l476g\_discovery.c**.

References **IDD\_WAKEUP\_GPIO\_PORT**, and **IDD\_WAKEUP\_PIN**.

**void MFX\_IO\_Write ( uint16\_t Addr,**  
                  **uint8\_t Reg,**  
                  **uint8\_t Value**  
                  **)**



MFX writes single data.

**Parameters:**

**Addr,:** I2C address

**Reg,:** Register address

**Value,:** Data to be written

**Return values:**

**None**

Definition at line **1600** of file **stm32l476g\_discovery.c**.

References **I2C2\_WriteData()**.

```
void MFX_IO_WriteMultiple ( uint16_t Addr,  
                             uint8_t  Reg,  
                             uint8_t * Buffer,  
                             uint16_t Length  
                             )
```

MFX writes multiple data.

**Parameters:**

**Addr,:** I2C address

**Reg,:** Register address

**Buffer,:** Pointer to data buffer

**Length,:** Length of the data

**Return values:**

**None**

Definition at line **1637** of file **stm32l476g\_discovery.c**.

References **I2C2\_WriteBuffer()**.

**void SPIx\_DeInit ( void ) [static]**

SPIx Bus Deinitialization.

**Return values:**

**None**

Definition at line **627** of file **stm32l476g\_discovery.c**.

References **SpiHandle**, and **SPIx\_MspDeInit()**.

Referenced by **ACCELERO\_IO\_DeInit()**, **GYRO\_IO\_DeInit()**, and **MAGNETO\_IO\_DeInit()**.

**static void SPIx\_Init ( void ) [static]**

SPIx Bus initialization.

**Return values:**

**None**

Definition at line **571** of file **stm32l476g\_discovery.c**.

References **DISCOVERY\_SPIx**, **SpiHandle**, and **SPIx\_MspInit()**.

Referenced by **ACCELERO\_IO\_Init()**, **GYRO\_IO\_Init()**, and **MAGNETO\_IO\_Init()**.

**static void SPIx\_MspDeInit ( void ) [static]**

SPI MSP DeInit.

**Return values:**

**None**

Definition at line **641** of file **stm32l476g\_discovery.c**.

References **DISCOVERY\_SPIx\_CLOCK\_DISABLE**,  
**DISCOVERY\_SPIx\_GPIO\_CLK\_ENABLE**,  
**DISCOVERY\_SPIx\_GPIO\_FORCE\_RESET**,  
**DISCOVERY\_SPIx\_GPIO\_PORT**,  
**DISCOVERY\_SPIx\_GPIO\_RELEASE\_RESET**,  
**DISCOVERY\_SPIx\_MISO\_PIN**, **DISCOVERY\_SPIx\_MOSI\_PIN**, and  
**DISCOVERY\_SPIx\_SCK\_PIN**.

Referenced by **SPIx\_DeInit()**.

**static void SPIx\_Mspltinit ( SPI\_HandleTypeDef \* **hspt** ) [static]**

SPI MSP Init.

**Parameters:**

**hspt**,: SPI handle

**Return values:**

**None**

Definition at line **604** of file **stm32l476g\_discovery.c**.

References **DISCOVERY\_SPIx\_AF**,  
**DISCOVERY\_SPIx\_CLOCK\_ENABLE**,  
**DISCOVERY\_SPIx\_GPIO\_CLK\_ENABLE**,  
**DISCOVERY\_SPIx\_GPIO\_PORT**, **DISCOVERY\_SPIx\_MISO\_PIN**,  
**DISCOVERY\_SPIx\_MOSI\_PIN**, and **DISCOVERY\_SPIx\_SCK\_PIN**.

Referenced by **SPIx\_Init()**.

**static uint8\_t SPIx\_Read ( void ) [static]**

Receives a Byte from the SPI bus.

**Return values:**

**The** received byte value

Definition at line **716** of file **stm32l476g\_discovery.c**.

References **SpiHandle**.

Referenced by **ACCELERO\_IO\_Read()**, and **MAGNETO\_IO\_Read()**.

```
static void SPIx_Write ( uint8_t Byte ) [static]
```

Sends a Byte through the SPI interface.

**Parameters:**

**Byte** : Byte to send.

**Return values:**

**none**.

Definition at line **692** of file **stm32l476g\_discovery.c**.

References **SpiHandle**.

Referenced by **ACCELERO\_IO\_Read()**, **ACCELERO\_IO\_Write()**, **MAGNETO\_IO\_Read()**, and **MAGNETO\_IO\_Write()**.

```
static uint8_t SPIx_WriteRead ( uint8_t Byte ) [static]
```

Sends a Byte through the SPI interface and return the Byte received from the SPI bus.

**Parameters:**

**Byte** : Byte send.

**Return values:**

**none.**

Definition at line **662** of file **stm32l476g\_discovery.c**.

References **SpiHandle**.

Referenced by **GYRO\_IO\_Read()**, and **GYRO\_IO\_Write()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Exported Functions

[STM32L476G-DISCOVERY COMPASS](#)

## Functions

	void <b>ACCELERO_IO_DeInit</b> (void) De-Configures COMPASS/ACCELEROMETER io interface.
	void <b>MAGNETO_IO_DeInit</b> (void) de-Configures COMPASS/MAGNETO SPI interface.
<b>COMPASS_StatusTypeDef</b>	<b>BSP_COMPASS_Init</b> (void) Initialize the COMPASS.
	void <b>BSP_COMPASS_DeInit</b> (void) DeInitialize the COMPASS.
	void <b>BSP_COMPASS_LowPower</b> (void) Put the COMPASS in low power mode.
	void <b>BSP_COMPASS_AccGetXYZ</b> (int16_t *pDataXYZ) Get XYZ acceleration values.
	void <b>BSP_COMPASS_MagGetXYZ</b> (int16_t *pDataXYZ) Get XYZ magnetometer values.

## Function Documentation

**void ACCELERO\_IO\_DeInit ( void )**

De-Configures COMPASS/ACCELEROMETER io interface.

**Return values:**

**None**

Definition at line **1188** of file **stm32l476g\_discovery.c**.

References **ACCELERO\_CS\_GPIO\_CLK\_ENABLE**, **ACCELERO\_CS\_GPIO\_PORT**, **ACCELERO\_CS\_HIGH**, **ACCELERO\_CS\_PIN**, and **SPIx\_DeInit()**.

Referenced by **BSP\_COMPASS\_DeInit()**.

**void BSP\_COMPASS\_AccGetXYZ ( int16\_t \* pDataXYZ )**

Get XYZ acceleration values.

**Parameters:**

**pDataXYZ** Pointer on 3 angular accelerations table with  
pDataXYZ[0] = X axis, pDataXYZ[1] = Y axis,  
pDataXYZ[2] = Z axis

**Return values:**

**None**

Definition at line **219** of file **stm32l476g\_discovery\_compass.c**.

References **AccelerometerDrv**.

**void BSP\_COMPASS\_DeInit ( void )**



DeInitialize the COMPASS.

**Return values:**

**None.**

Definition at line **188** of file **stm32l476g\_discovery\_compass.c**.

References **ACCELERO\_IO\_DeInit()**, and **MAGNETO\_IO\_DeInit()**.

### **COMPASS\_StatusTypeDef BSP\_COMPASS\_Init ( void )**

Initialize the COMPASS.

**Return values:**

**COMPASS\_OK** or COMPASS\_ERROR

Definition at line **112** of file **stm32l476g\_discovery\_compass.c**.

References **AccelerometerDrv**, **COMPASS\_ERROR**, **COMPASS\_OK**, and **MagnetoDrv**.

### **void BSP\_COMPASS\_LowPower ( void )**

Put the COMPASS in low power mode.

**Return values:**

**None**

Definition at line **199** of file **stm32l476g\_discovery\_compass.c**.

References **AccelerometerDrv**, and **MagnetoDrv**.

### **void BSP\_COMPASS\_MagGetXYZ ( int16\_t \* pDataXYZ )**

Get XYZ magnetometer values.

**Parameters:**

**pDataXYZ** Pointer on 3 magnetometer values table with  
pDataXYZ[0] = X axis, pDataXYZ[1] = Y axis,  
pDataXYZ[2] = Z axis

**Return values:**

**None**

Definition at line **233** of file **stm32l476g\_discovery\_compass.c**.

References **MagnetoDrv**.

**void MAGNETO\_IO\_DeInit ( void )**

de-Configures COMPASS/MAGNETO SPI interface.

**Return values:**

**None**

Definition at line **1275** of file **stm32l476g\_discovery.c**.

References **MAGNETO\_CS\_GPIO\_CLK\_ENABLE**,  
**MAGNETO\_CS\_GPIO\_PORT**, **MAGNETO\_CS\_HIGH**,  
**MAGNETO\_CS\_PIN**, **MAGNETO\_DRDY\_PIN**,  
**MAGNETO\_INT1\_PIN**, and **SPIx\_DeInit()**.

Referenced by **BSP\_COMPASS\_DeInit()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Variables](#)

## Private Variables

[STM32L476G-DISCOVERY ACCELEROMETER](#)

## Variables

---

```
static ACCELERO_DrvTypeDef * AccelerometerDrv
```

---

## Variable Documentation

**ACCELERO\_DrvTypeDef\* AccelerometerDrv** [static]

Definition at line **82** of file **stm32l476g\_discovery\_accelerometer.c**.

Referenced by **BSP\_ACCELERO\_GetXYZ()**,  
**BSP\_ACCELERO\_Init()**, and **BSP\_ACCELERO\_Reset()**.

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Variables](#)

## Private Variables

[STM32L476G-DISCOVERY COMPASS](#)

## Variables

static ACCELERO_DrvTypeDef *	<b>AccelerometerDrv</b>
------------------------------	-------------------------

static MAGNETO_DrvTypeDef *	<b>MagnetoDrv</b>
-----------------------------	-------------------

## Variable Documentation

**ACCELERO\_DrvTypeDef\* AccelerometerDrv** [static]

Definition at line **86** of file **stm32l476g\_discovery\_compass.c**.

Referenced by **BSP\_COMPASS\_AccGetXYZ()**,  
**BSP\_COMPASS\_Init()**, and **BSP\_COMPASS\_LowPower()**.

**MAGNETO\_DrvTypeDef\* MagnetoDrv** [static]

Definition at line **87** of file **stm32l476g\_discovery\_compass.c**.

Referenced by **BSP\_COMPASS\_Init()**,  
**BSP\_COMPASS\_LowPower()**, and **BSP\_COMPASS\_MagGetXYZ()**.



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## Private Constants

[STM32L476G-DISCOVERY GLASS LCD](#)

## Defines

#define	ASCII_CHAR_0	0x30	/* 0 */
---------	--------------	------	---------

#define	ASCII_CHAR_AT_SYMBOL	0x40	/* @ */
---------	----------------------	------	---------

#define	ASCII_CHAR_LEFT_OPEN_BRACKET	0x5B	/* [ */
---------	------------------------------	------	---------

#define	ASCII_CHAR_APOSTROPHE	0x60	/* ` */
---------	-----------------------	------	---------

#define	ASCII_CHAR_LEFT_OPEN_BRACE	0x7B	/* ( */
---------	----------------------------	------	---------

## Define Documentation

**#define ASCII\_CHAR\_0** 0x30 /\* 0 \*/

Definition at line **61** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **Convert()**.

**#define ASCII\_CHAR\_APOSTROPHE** 0x60 /\* ` \*/

Definition at line **64** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **Convert()**.

**#define ASCII\_CHAR\_AT\_SYMBOL** 0x40 /\* @ \*/

Definition at line **62** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **Convert()**.

**#define ASCII\_CHAR\_LEFT\_OPEN\_BRACE** 0x7B /\* ( \*/

Definition at line **65** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **Convert()**.

**#define ASCII\_CHAR\_LEFT\_OPEN\_BRACKET** 0x5B /\* [ \*/

Definition at line **63** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **Convert()**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Private Functions

[STM32L476G-DISCOVERY AUDIO](#)

## Functions

static void	<b>AUDIO_CODEC_Reset</b> (void) Resets the audio codec.
static uint8_t	<b>AUDIO_SAIx_Init</b> (uint32_t AudioFreq) Initializes the Audio Codec audio interface (SAI).
static uint8_t	<b>AUDIO_SAIx_DeInit</b> (void) De-initializes the Audio Codec audio interface (SAI).
static uint8_t	<b>AUDIO_DFSDMx_Init</b> (uint32_t AudioFreq) Initializes the Digital Filter for Sigma-Delta Modulators interface (DFSDM).
static uint8_t	<b>AUDIO_DFSDMx_DeInit</b> (void) De-initializes the Digital Filter for Sigma-Delta Modulators interface (DFSDM).
static uint8_t	<b>AUDIO_SAIPLLConfig</b> (uint32_t Frequency) Configures the SAI PLL clock according to the required audio frequency.
void	<b>HAL_SAI_MspInit</b> (SAI_HandleTypeDef *hsai) SAI MSP Init.
void	<b>HAL_SAI_MspDeInit</b> (SAI_HandleTypeDef *hsai) SAI MSP De-init.
void	<b>HAL_DFSDM_ChannelMspInit</b> (DFSDM_Channel_HandleTypeDef *hdfsdm_channel) Initializes the DFSDM channel MSP.
void	<b>HAL_DFSDM_ChannelMspDeInit</b> (DFSDM_Channel_HandleTypeDef *hdfsdm_channel) De-initializes the DFSDM channel MSP.
void	<b>HAL_DFSDM_FilterMspInit</b> (DFSDM_Filter_HandleTypeDef *hdfsdm_filter) Initializes the DFSDM filter MSP.
void	<b>HAL_DFSDM_FilterMspDeInit</b> (DFSDM_Filter_HandleTypeDef *hdfsdm_filter) De-initializes the DFSDM filter MSP.



## Function Documentation

**static void** [AUDIO\\_CODEC\\_Reset](#) (**void** ) [**static**]

Resets the audio codec.

It restores the default configuration of the codec (this function shall be called before initializing the codec).

### Return values:

**None**

Definition at line [1136](#) of file [stm32l476g\\_discovery\\_audio.c](#).

References [AUDIO\\_I2C\\_ADDRESS](#), and [AUDIO\\_OUT\\_TypeDef::AudioDrv](#).

Referenced by [BSP\\_AUDIO\\_OUT\\_Init\(\)](#).

**static uint8\_t** [AUDIO\\_DFSDMx\\_DeInit](#) (**void** ) [**static**]

De-initializes the Digital Filter for Sigma-Delta Modulators interface (DFSDM).

### Return values:

**BSP** AUDIO status

Definition at line [1220](#) of file [stm32l476g\\_discovery\\_audio.c](#).

References [AUDIO\\_DFSDMx\\_CLK\\_DISABLE](#), [AUDIO\\_ERROR](#), [AUDIO\\_OK](#), [AUDIO\\_SAIx\\_PLL\\_DISABLE](#), [BSP\\_AUDIO\\_hDfsdmLeftFilter](#), and [AUDIO\\_IN\\_TypeDef::hDfsdmLeftChannel](#).

Referenced by [BSP\\_AUDIO\\_IN\\_DeInit\(\)](#), and [BSP\\_AUDIO\\_IN\\_SetFrequency\(\)](#).



```
static uint8_t AUDIO_DFSDMx_Init ( uint32_t AudioFreq ) [static]
```

Initializes the Digital Filter for Sigma-Delta Modulators interface (DFSDM).

**Parameters:**

**AudioFreq,:** Audio frequency to be used to set correctly the DFSDM peripheral.

**Return values:**

**BSP** AUDIO status

Definition at line **1157** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_LEFT\_FILTER**, **AUDIO\_ERROR**, **AUDIO\_OK**, **BSP\_AUDIO\_hDfsdmLeftFilter**, **DFSDMClockDivider**, **DFSDMFilterOrder**, **DFSDMOverSampling**, **DFSDMRightBitShift**, and **AUDIO\_IN\_TypeDef::hDfsdmLeftChannel**.

Referenced by **BSP\_AUDIO\_IN\_Init()**, and **BSP\_AUDIO\_IN\_SetFrequency()**.

```
static uint8_t AUDIO_SAIPLLConfig ( uint32_t Frequency ) [static]
```

Configures the SAI PLL clock according to the required audio frequency.

**Parameters:**

**Frequency,:** Audio frequency.

**Return values:**

**BSP** AUDIO status

**Note:**

The SAI PLL input clock must be configured in the user application. The SAI PLL configuration done within this function assumes that the SAI PLL input clock runs at 8 MHz.

Definition at line **1363** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, and **AUDIO\_OK**.

Referenced by **BSP\_AUDIO\_IN\_Init()**, **BSP\_AUDIO\_IN\_SetFrequency()**, **BSP\_AUDIO\_OUT\_Init()**, and **BSP\_AUDIO\_OUT\_SetFrequency()**.

**static uint8\_t AUDIO\_SAIx\_DeInit ( void ) [static]**

De-initializes the Audio Codec audio interface (SAI).

**Return values:**

**BSP** AUDIO status

Definition at line **1029** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**, **AUDIO\_SAIx\_PLL\_DISABLE**, and **BSP\_AUDIO\_hSai**.

Referenced by **BSP\_AUDIO\_OUT\_DeInit()**.

**static uint8\_t AUDIO\_SAIx\_Init ( uint32\_t AudioFreq ) [static]**

Initializes the Audio Codec audio interface (SAI).

**Parameters:**

**AudioFreq,:** Audio frequency to be configured for the SAI peripheral.

**Note:**

The default SlotActive configuration is set to

CODEC\_AUDIOFRAME\_SLOT\_0123 and user can update this configuration using

**Return values:**

**BSP** AUDIO status

Definition at line **963** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**, **AUDIO\_SAIx**, **BSP\_AUDIO\_hSai**, and **SAIClockDivider**.

Referenced by **BSP\_AUDIO\_OUT\_Init()**.

**void HAL\_DFSDM\_ChannelMspDeInit ( DFSDM\_Channel\_HandleTy**

De-initializes the DFSDM channel MSP.

**Parameters:**

**hdfsdm\_channel** : DFSDM channel handle.

**Return values:**

**None**

Definition at line **1279** of file **stm32l476g\_discovery\_audio.c**.

References

**AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_CLK\_ENABLE**,  
**AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_PORT**,  
**AUDIO\_DFSDMx\_CKOUT\_PIN**, and  
**AUDIO\_DFSDMx\_DMIC\_DATIN\_PIN**.

**void HAL\_DFSDM\_ChannelMspInit ( DFSDM\_Channel\_HandleTypeI**

Initializes the DFSDM channel MSP.

**Parameters:**

**hdfsdm\_channel** : DFSDM channel handle.

**Return values:**

**None**

Definition at line **1255** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_AF**,  
**AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_CLK\_ENABLE**,  
**AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_PORT**,  
**AUDIO\_DFSDMx\_CKOUT\_PIN**, **AUDIO\_DFSDMx\_CLK\_ENABLE**,  
and **AUDIO\_DFSDMx\_DMIC\_DATIN\_PIN**.

**void HAL\_DFSDM\_FilterMspDeInit** ( DFSDM\_Filter\_HandleTypeDef

De-initializes the DFSDM filter MSP.

**Parameters:**

**hdfsdm\_filter** : DFSDM filter handle.

**Return values:**

**None**

Definition at line **1343** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_DMAX\_CLK\_DISABLE**,  
**AUDIO\_DFSDMx\_DMAX\_LEFT\_IRQ**, and  
**AUDIO\_IN\_TypeDef::hDmaDfsdmLeft**.

**void HAL\_DFSDM\_FilterMspInit** ( DFSDM\_Filter\_HandleTypeDef \* **h**

Initializes the DFSDM filter MSP.

**Parameters:**

**hdfsdm\_filter** : DFSDM filter handle.

**Return values:**

**None**

Definition at line **1304** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_CLK\_ENABLE**,  
**AUDIO\_DFSDMx\_DMAX\_CLK\_ENABLE**,  
**AUDIO\_DFSDMx\_DMAX\_LEFT\_CHANNEL**,  
**AUDIO\_DFSDMx\_DMAX\_LEFT\_IRQ**,  
**AUDIO\_DFSDMx\_DMAX\_MEM\_DATA\_SIZE**,  
**AUDIO\_DFSDMx\_DMAX\_PERIPH\_DATA\_SIZE**,  
**AUDIO\_OUT\_IRQ\_PREPRIO**, and  
**AUDIO\_IN\_TypeDef::hDmaDfsdmLeft**.

**void HAL\_SAI\_MspDeInit ( SAI\_HandleTypeDef \* **hsai** )**

SAI MSP De-init.

**Parameters:**

**hsai** : pointer to a SAI\_HandleTypeDef structure

**Return values:**

**None**

Definition at line **1109** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_SAIx\_CLK\_DISABLE**,  
**AUDIO\_SAIx\_DMAX\_CLK\_DISABLE**, **AUDIO\_SAIx\_DMAX\_IRQ**,  
**AUDIO\_SAIx\_FS\_PIN**, **AUDIO\_SAIx\_MCK\_PIN**,  
**AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_DISABLE**,  
**AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_GPIO\_PORT**,  
**AUDIO\_SAIx\_SCK\_PIN**, **AUDIO\_SAIx\_SD\_PIN**, and **hDmaSai**.

**void HAL\_SAI\_MspInit ( SAI\_HandleTypeDef \* **hsai** )**

SAI MSP Init.

**Parameters:**

**hsai** : pointer to a SAI\_HandleTypeDef structure

**Return values:**

**None**

Definition at line **1054** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_OUT\_IRQ\_PREPRIO**, **AUDIO\_SAIx**,  
**AUDIO\_SAIx\_CLK\_ENABLE**, **AUDIO\_SAIx\_DMax\_CHANNEL**,  
**AUDIO\_SAIx\_DMax\_CLK\_ENABLE**, **AUDIO\_SAIx\_DMax\_IRQ**,  
**AUDIO\_SAIx\_DMax\_MEM\_DATA\_SIZE**,  
**AUDIO\_SAIx\_DMax\_PERIPH\_DATA\_SIZE**, **AUDIO\_SAIx\_FS\_PIN**,  
**AUDIO\_SAIx\_MCK\_PIN**, **AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_AF**,  
**AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_ENABLE**,  
**AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_GPIO\_PORT**,  
**AUDIO\_SAIx\_SCK\_PIN**, **AUDIO\_SAIx\_SD\_PIN**, and **hDmaSai**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Modules](#) | [Defines](#)

## Exported Constants

[STM32L476G-DISCOVERY AUDIO](#)

**Modules**

BSP Audio Out Option
BSP Audio Sample Rate



## Defines

```
#define AUDIO_SAIx_SAI1_Block_A
#define AUDIO_SAIx_CLK_ENABLE() __HAL_RCC_SAI1_CLK_EN
#define AUDIO_SAIx_CLK_DISABLE() __HAL_RCC_SAI1_CLK_DI
#define AUDIO_SAIx_MCK_SCK_SD_FS_AF GPIO_AF13_SAI1
#define AUDIO_SAIx_MCK_SCK_SD_FS_ENABLE() __HAL_RCC_
#define AUDIO_SAIx_MCK_SCK_SD_FS_DISABLE() __HAL_RCC_
#define AUDIO_SAIx_FS_PIN GPIO_PIN_4
#define AUDIO_SAIx_SCK_PIN GPIO_PIN_5
#define AUDIO_SAIx_SD_PIN GPIO_PIN_6
#define AUDIO_SAIx_MCK_PIN GPIO_PIN_2
#define AUDIO_SAIx_MCK_SCK_SD_FS_GPIO_PORT GPIOE
#define AUDIO_SAIx_DMax_CLK_ENABLE() __HAL_RCC_DMA2_
#define AUDIO_SAIx_DMax_CLK_DISABLE() __HAL_RCC_DMA2_
#define AUDIO_SAIx_DMax_CHANNEL DMA2_Channel1
#define AUDIO_SAIx_DMax_IRQ DMA2_Channel1_IRQn
#define AUDIO_SAIx_DMax_PERIPH_DATA_SIZE DMA_PDATAAL
#define AUDIO_SAIx_DMax_MEM_DATA_SIZE DMA_MDATAALIGI
#define DMA_MAX_SZE (uint32_t)0xFFFF
#define AUDIO_SAIx_DMax_IRQHandler DMA2_Channel1_IRQHar
#define AUDIO_OUT_IRQ_PREPRIO 5 /* Select the preemption prior
#define AUDIO_SAIx_PLL_DISABLE() HAL_RCCEX_DisablePLLSA
#define AUDIO_DFSDMx_LEFT_CHANNEL DFSDM_Channel2
#define AUDIO_DFSDMx_LEFT_FILTER DFSDM_Filter0
#define AUDIO_DFSDMx_CLK_ENABLE() __HAL_RCC_DFSDM_C
#define AUDIO_DFSDMx_CLK_DISABLE() __HAL_RCC_DFSDM_C
#define AUDIO_DFSDMx_CKOUT_PIN GPIO_PIN_9
#define AUDIO_DFSDMx_DMIC_DATIN_PIN GPIO_PIN_7
#define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_PORT GPI
#define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_CLK_ENAB
#define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_CLK_DISAE
#define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_AF GPIO_AF6_DI
```

```
#define AUDIO_DFSDMx_DMAX_CLK_ENABLE() __HAL_RCC_DM
#define AUDIO_DFSDMx_DMAX_CLK_DISABLE() __HAL_RCC_DM
#define AUDIO_DFSDMx_DMAX_LEFT_CHANNEL DMA1_Channel
#define AUDIO_DFSDMx_DMAX_LEFT_IRQ DMA1_Channel4_IRQr
#define AUDIO_DFSDMx_DMAX_PERIPH_DATA_SIZE DMA_PDAT
#define AUDIO_DFSDMx_DMAX_MEM_DATA_SIZE DMA_MDATAA
#define AUDIO_DFSDM_DMAX_LEFT_IRQHandler DMA1_Channel
#define AUDIO_IN_IRQ_PREPRIO 6 /* Select the preemption priority
#define AUDIODATA_SIZE 2 /* 16-bits audio data size */
#define AUDIO_OK 0
#define AUDIO_ERROR 1
#define AUDIO_TIMEOUT 2
#define DEFAULT_AUDIO_IN_FREQ BSP_AUDIO_FREQUENCY_1
#define DEFAULT_AUDIO_IN_BIT_RESOLUTION 16
#define DEFAULT_AUDIO_IN_CHANNEL_NBR 1 /* Mono = 1, Stere
#define DEFAULT_AUDIO_IN_VOLUME 64
#define CODEC_RESET_DELAY 5
```

## Define Documentation

**#define AUDIO\_DFSDM\_DMAx\_LEFT\_IRQHandler** DMA1\_Channel1

Definition at line 163 of file stm32l476g\_discovery\_audio.h.

**#define AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_AF** GPIO\_AF6\_DFSDM1

Definition at line 153 of file stm32l476g\_discovery\_audio.h.

Referenced by HAL\_DFSDM\_ChannelMspInit().

**#define AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_CLK\_DISAB**

Definition at line 152 of file stm32l476g\_discovery\_audio.h.

**#define AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_CLK\_ENAB**

Definition at line 151 of file stm32l476g\_discovery\_audio.h.

Referenced by HAL\_DFSDM\_ChannelMspDeInit(), and  
HAL\_DFSDM\_ChannelMspInit().

**#define AUDIO\_DFSDMx\_CKOUT\_DMIC\_DATIN\_GPIO\_PORT** GPIOA

Definition at line 150 of file stm32l476g\_discovery\_audio.h.

Referenced by HAL\_DFSDM\_ChannelMspDeInit(), and  
HAL\_DFSDM\_ChannelMspInit().

**#define AUDIO\_DFSDMx\_CKOUT\_PIN** GPIO\_PIN\_9

Definition at line **148** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_ChannelMspDeInit()**, and  
**HAL\_DFSDM\_ChannelMspInit()**.

**#define AUDIO\_DFSDMx\_CLK\_DISABLE ( ) \_\_HAL\_RCC\_DFSDM**

Definition at line **147** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **AUDIO\_DFSDMx\_DeInit()**.

**#define AUDIO\_DFSDMx\_CLK\_ENABLE ( ) \_\_HAL\_RCC\_DFSDM**

Definition at line **146** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_ChannelMspInit()**, and  
**HAL\_DFSDM\_FilterMspInit()**.

**#define AUDIO\_DFSDMx\_DMAx\_CLK\_DISABLE ( ) \_\_HAL\_RCC\_I**

Definition at line **157** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_FilterMspDeInit()**.

**#define AUDIO\_DFSDMx\_DMAx\_CLK\_ENABLE ( ) \_\_HAL\_RCC\_D**

Definition at line **156** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_FilterMspInit()**.

**#define AUDIO\_DFSDMx\_DMAx\_LEFT\_CHANNEL DMA1\_Channel**

Definition at line **158** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_FilterMspInit()**.

**#define AUDIO\_DFSDMx\_DMAx\_LEFT\_IRQ DMA1\_Channel4\_IRQ**

Definition at line **159** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_FilterMspDeInit()**, and  
**HAL\_DFSDM\_FilterMspInit()**.

**#define AUDIO\_DFSDMx\_DMAx\_MEM\_DATA\_SIZE DMA\_MDATAA**

Definition at line **161** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_FilterMspInit()**.

**#define AUDIO\_DFSDMx\_DMAx\_PERIPH\_DATA\_SIZE DMA\_PDAT**

Definition at line **160** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_FilterMspInit()**.

**#define AUDIO\_DFSDMx\_DMIC\_DATIN\_PIN GPIO\_PIN\_7**

Definition at line **149** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_DFSDM\_ChannelMspDeInit()**, and  
**HAL\_DFSDM\_ChannelMspInit()**.

**#define AUDIO\_DFSDMx\_LEFT\_CHANNEL DFSDM\_Channel2**

Definition at line **144** of file **stm32l476g\_discovery\_audio.h**.

**#define AUDIO\_DFSDMx\_LEFT\_FILTER DFSDM\_Filter0**

Definition at line **145** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **AUDIO\_DFSDMx\_Init()**.

**#define AUDIO\_ERROR 1**

Definition at line **176** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **AUDIO\_DFSDMx\_DeInit()**, **AUDIO\_DFSDMx\_Init()**, **AUDIO\_SAIPLLConfig()**, **AUDIO\_SAIx\_DeInit()**, **AUDIO\_SAIx\_Init()**, **BSP\_AUDIO\_IN\_DeInit()**, **BSP\_AUDIO\_IN\_Init()**, **BSP\_AUDIO\_IN\_Pause()**, **BSP\_AUDIO\_IN\_Record()**, **BSP\_AUDIO\_IN\_Resume()**, **BSP\_AUDIO\_IN\_SetFrequency()**, **BSP\_AUDIO\_IN\_Stop()**, **BSP\_AUDIO\_OUT\_ChangeBuffer()**, **BSP\_AUDIO\_OUT\_DeInit()**, **BSP\_AUDIO\_OUT\_Init()**, **BSP\_AUDIO\_OUT\_Pause()**, **BSP\_AUDIO\_OUT\_Play()**, **BSP\_AUDIO\_OUT\_Resume()**, **BSP\_AUDIO\_OUT\_SetFrequency()**, **BSP\_AUDIO\_OUT\_SetMute()**, **BSP\_AUDIO\_OUT\_SetOutputMode()**, **BSP\_AUDIO\_OUT\_SetVolume()**, and **BSP\_AUDIO\_OUT\_Stop()**.

**#define AUDIO\_IN\_IRQ\_PREPRIO 6** /\* Select the preemption priority \*/

Definition at line **166** of file **stm32l476g\_discovery\_audio.h**.

**#define AUDIO\_OK 0**

Definition at line **175** of file **stm32l476g\_discovery\_audio.h**.

Referenced by [AUDIO\\_DFSDMx\\_DeInit\(\)](#), [AUDIO\\_DFSDMx\\_Init\(\)](#), [AUDIO\\_SAIPLLConfig\(\)](#), [AUDIO\\_SAIx\\_DeInit\(\)](#), [AUDIO\\_SAIx\\_Init\(\)](#), [BSP\\_AUDIO\\_IN\\_DeInit\(\)](#), [BSP\\_AUDIO\\_IN\\_Init\(\)](#), [BSP\\_AUDIO\\_IN\\_Pause\(\)](#), [BSP\\_AUDIO\\_IN\\_Record\(\)](#), [BSP\\_AUDIO\\_IN\\_Resume\(\)](#), [BSP\\_AUDIO\\_IN\\_SetFrequency\(\)](#), [BSP\\_AUDIO\\_IN\\_Stop\(\)](#), [BSP\\_AUDIO\\_OUT\\_ChangeBuffer\(\)](#), [BSP\\_AUDIO\\_OUT\\_DeInit\(\)](#), [BSP\\_AUDIO\\_OUT\\_Init\(\)](#), [BSP\\_AUDIO\\_OUT\\_Pause\(\)](#), [BSP\\_AUDIO\\_OUT\\_Play\(\)](#), [BSP\\_AUDIO\\_OUT\\_Resume\(\)](#), [BSP\\_AUDIO\\_OUT\\_SetFrequency\(\)](#), [BSP\\_AUDIO\\_OUT\\_SetMute\(\)](#), [BSP\\_AUDIO\\_OUT\\_SetOutputMode\(\)](#), [BSP\\_AUDIO\\_OUT\\_SetVolume\(\)](#), and [BSP\\_AUDIO\\_OUT\\_Stop\(\)](#).

**#define AUDIO\_OUT\_IRQ\_PREPRIO 5** /\* Select the preemption pri

Definition at line [135](#) of file [stm32l476g\\_discovery\\_audio.h](#).

Referenced by [HAL\\_DFSDM\\_FilterMspInit\(\)](#), and [HAL\\_SAI\\_MspInit\(\)](#).

**#define AUDIO\_SAIx SAI1\_Block\_A**

Definition at line [110](#) of file [stm32l476g\\_discovery\\_audio.h](#).

Referenced by [AUDIO\\_SAIx\\_Init\(\)](#), and [HAL\\_SAI\\_MspInit\(\)](#).

**#define AUDIO\_SAIx\_CLK\_DISABLE ( ) \_\_HAL\_RCC\_SAI1\_CLK\_**

Definition at line [112](#) of file [stm32l476g\\_discovery\\_audio.h](#).

Referenced by [HAL\\_SAI\\_MspDeInit\(\)](#).

**#define AUDIO\_SAIx\_CLK\_ENABLE ( ) \_\_HAL\_RCC\_SAI1\_CLK\_E**

Definition at line **111** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_DMAx\_CHANNEL DMA2\_Channel1**

Definition at line **126** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_DMAx\_CLK\_DISABLE ( ) \_\_HAL\_RCC\_DMA**

Definition at line **125** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspDeInit()**.

**#define AUDIO\_SAIx\_DMAx\_CLK\_ENABLE ( ) \_\_HAL\_RCC\_DMA**

Definition at line **124** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_DMAx\_IRQ DMA2\_Channel1\_IRQn**

Definition at line **127** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspDeInit()**, and **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_DMAx\_IRQHandler DMA2\_Channel1\_IRQHa**

Definition at line **132** of file **stm32l476g\_discovery\_audio.h**.



**#define AUDIO\_SAIx\_DMAx\_MEM\_DATA\_SIZE DMA\_MDATAALIGN**

Definition at line **129** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_DMAx\_PERIPH\_DATA\_SIZE DMA\_PDATAALIGN**

Definition at line **128** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_FS\_PIN GPIO\_PIN\_4**

Definition at line **117** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspDeInit()**, and **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_MCK\_PIN GPIO\_PIN\_2**

Definition at line **120** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspDeInit()**, and **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_AF GPIO\_AF13\_SAI1**

Definition at line **113** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspInit()**.

**#define AUDIO\_SAIx\_MCK\_SCK\_SD\_FS\_DISABLE ( ) \_\_HAL\_RCC**

Definition at line **116** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspDeInit()**.

```
#define AUDIO_SAIx_MCK_SCK_SD_FS_ENABLE ( ) __HAL_RCC_
```

Definition at line **115** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspInit()**.

```
#define AUDIO_SAIx_MCK_SCK_SD_FS_GPIO_PORT GPIOE
```

Definition at line **121** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspDeInit()**, and **HAL\_SAI\_MspInit()**.

```
#define AUDIO_SAIx_PLL_DISABLE ( ) HAL_RCCEX_DisablePLL
```

Definition at line **138** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **AUDIO\_DFSDMx\_DeInit()**, and  
**AUDIO\_SAIx\_DeInit()**.

```
#define AUDIO_SAIx_SCK_PIN GPIO_PIN_5
```

Definition at line **118** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **HAL\_SAI\_MspDeInit()**, and **HAL\_SAI\_MspInit()**.

```
#define AUDIO_SAIx_SD_PIN GPIO_PIN_6
```

Definition at line **119** of file **stm32l476g\_discovery\_audio.h**.

Referenced by [HAL\\_SAI\\_MspDeInit\(\)](#), and [HAL\\_SAI\\_MspInit\(\)](#).

**#define AUDIO\_TIMEOUT 2**

Definition at line [177](#) of file [stm32l476g\\_discovery\\_audio.h](#).

**#define AUDIODATA\_SIZE 2 /\* 16-bits audio data size \*/**

Definition at line [172](#) of file [stm32l476g\\_discovery\\_audio.h](#).

**#define CODEC\_RESET\_DELAY 5**

Definition at line [190](#) of file [stm32l476g\\_discovery\\_audio.h](#).

**#define DEFAULT\_AUDIO\_IN\_BIT\_RESOLUTION 16**

Definition at line [181](#) of file [stm32l476g\\_discovery\\_audio.h](#).

**#define DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR 1 /\* Mono = 1, Stereo = 2 \*/**

Definition at line [182](#) of file [stm32l476g\\_discovery\\_audio.h](#).

Referenced by [BSP\\_AUDIO\\_IN\\_Record\(\)](#),  
[BSP\\_AUDIO\\_IN\\_Resume\(\)](#),  
[HAL\\_DFSDM\\_FilterRegConvCpltCallback\(\)](#), and  
[HAL\\_DFSDM\\_FilterRegConvHalfCpltCallback\(\)](#).

**#define DEFAULT\_AUDIO\_IN\_FREQ BSP\_AUDIO\_FREQUENCY\_1**

Definition at line [180](#) of file [stm32l476g\\_discovery\\_audio.h](#).

```
#define DEFAULT_AUDIO_IN_VOLUME 64
```

Definition at line **183** of file **stm32l476g\_discovery\_audio.h**.

```
#define DMA_MAX_SIZE (uint32_t)0xFFFF
```

Definition at line **130** of file **stm32l476g\_discovery\_audio.h**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Enumerations](#)

## Exported Types

[STM32L476G-DISCOVERY GLASS LCD](#)

## Enumerations

enum	<b>DigitPosition_Typedef</b> { <b>LCD_DIGIT_POSITION_1</b> = 0, <b>LCD_DIGIT_POSITION_2</b> = 1, <b>LCD_DIGIT_POSITION_3</b> = 2, <b>LCD_DIGIT_POSITION_4</b> = 3, <b>LCD_DIGIT_POSITION_5</b> = 4, <b>LCD_DIGIT_POSITION_6</b> = 5, <b>LCD_DIGIT_MAX_NUMBER</b> = 6 }	LCD Glass digit position. <a href="#">More...</a>
enum	<b>Point_Typedef</b> { <b>POINT_OFF</b> = 0, <b>POINT_ON</b> = 1 }	LCD Glass point Warning: element values correspond to LCD Glass point. <a href="#">More...</a>
enum	<b>DoublePoint_Typedef</b> { <b>DOUBLEPOINT_OFF</b> = 0, <b>DOUBLEPOINT_ON</b> = 1 }	LCD Glass Double point Warning: element values correspond to LCD Glass Double point. <a href="#">More...</a>
enum	<b>BatteryLevel_Typedef</b> { <b>BATTERYLEVEL_OFF</b> = 0, <b>BATTERYLEVEL_1_4</b> = 1, <b>BATTERYLEVEL_1_2</b> = 2, <b>BATTERYLEVEL_3_4</b> = 3, <b>BATTERYLEVEL_FULL</b> = 4 }	LCD Glass Battery Level element values correspond to different LCD Glass battery levels. <a href="#">More...</a>
enum	<b>BarId_Typedef</b> { <b>LCD_BAR_NONE</b> = 0, <b>LCD_BAR_0</b> = (1 << 0), <b>LCD_BAR_1</b> = (1 << 1), <b>LCD_BAR_2</b> = (1 << 2), <b>LCD_BAR_3</b> = (1 << 3) }	LCD Glass Bar Id. <a href="#">More...</a>

## Enumeration Type Documentation

### enum `BarId_Typedef`

LCD Glass Bar Id.

#### Enumerator:

*LCD\_BAR\_NONE*

*LCD\_BAR\_0*

*LCD\_BAR\_1*

*LCD\_BAR\_2*

*LCD\_BAR\_3*

Definition at line **116** of file `stm32l476g_discovery_glass_lcd.h`.

### enum `BatteryLevel_Typedef`

LCD Glass Battery Level element values correspond to different LCD Glass battery levels.

#### Enumerator:

*BATTERYLEVEL\_OFF*

*BATTERYLEVEL\_1\_4*

*BATTERYLEVEL\_1\_2*

*BATTERYLEVEL\_3\_4*

*BATTERYLEVEL\_FULL*

Definition at line **104** of file `stm32l476g_discovery_glass_lcd.h`.

### enum `DigitPosition_Typedef`

LCD Glass digit position.

**Enumerator:**

*LCD\_DIGIT\_POSITION\_1*  
*LCD\_DIGIT\_POSITION\_2*  
*LCD\_DIGIT\_POSITION\_3*  
*LCD\_DIGIT\_POSITION\_4*  
*LCD\_DIGIT\_POSITION\_5*  
*LCD\_DIGIT\_POSITION\_6*  
*LCD\_DIGIT\_MAX\_NUMBER*

Definition at line **69** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**enum DoublePoint\_Typedef**

LCD Glass Double point Warning: element values correspond to LCD Glass Double point.

**Enumerator:**

*DOUBLEPOINT\_OFF*  
*DOUBLEPOINT\_ON*

Definition at line **94** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**enum Point\_Typedef**

LCD Glass point Warning: element values correspond to LCD Glass point.

**Enumerator:**

*POINT\_OFF*  
*POINT\_ON*

Definition at line **84** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).



Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## BATTERY Detection Constants

[Exported Constants](#)

## Defines

#define	<b>BATTERY_DETECTION_PIN</b>	GPIO_PIN_3
#define	<b>BATTERY_DETECTION_GPIO_PORT</b>	GPIOB
#define	<b>BATTERY_DETECTION_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOB_CLK_ENABLE()
#define	<b>BATTERY_DETECTION_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOB_CLK_DISABLE()

## Define Documentation

**#define BATTERY\_DETECTION\_GPIO\_CLK\_DISABLE ( ) \_\_HAL\_I**

Definition at line **135** of file **stm32l476g\_discovery.h**.

**#define BATTERY\_DETECTION\_GPIO\_CLK\_ENABLE ( ) \_\_HAL\_F**

Definition at line **134** of file **stm32l476g\_discovery.h**.

Referenced by **BSP\_SupplyModeDetection()**.

**#define BATTERY\_DETECTION\_GPIO\_PORT GPIOB**

Definition at line **133** of file **stm32l476g\_discovery.h**.

Referenced by **BSP\_SupplyModeDetection()**.

**#define BATTERY\_DETECTION\_PIN GPIO\_PIN\_3**

Definition at line **132** of file **stm32l476g\_discovery.h**.

Referenced by **BSP\_SupplyModeDetection()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Variables](#)

## Private Variables

[STM32L476G-DISCOVERY GLASS LCD](#)

## Variables

__IO uint8_t	<b>bLCDGlass_KeyPressed</b> = 0
LCD_HandleTypeDef	<b>LCDHandle</b>
const uint16_t	<b>CapLetterMap</b> [26]
const uint16_t	<b>NumberMap</b> [10]
uint32_t	<b>Digit</b> [4]
uint8_t	<b>LCDBar</b> = <b>BATTERYLEVEL_FULL</b>

## Variable Documentation

**\_\_IO uint8\_t bLCDGlass\_KeyPressed = 0**

Definition at line **77** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **BSP\_LCD\_GLASS\_ScrollSentence()**.

**const uint16\_t CapLetterMap[26]**

Initial value:

```
{  
    0xFE00, 0x6714, 0x1D00, 0x4714, 0x9D00, 0  
x9C00, 0x3F00, 0xFA00, 0x0014,  
  
    0x5300, 0x9841, 0x1900, 0x5A48, 0x5A09, 0  
x5F00, 0xFC00, 0x5F01, 0xFC01,  
  
    0xAF00, 0x0414, 0x5b00, 0x18C0, 0x5A81, 0  
x00C9, 0x0058, 0x05C0  
}
```

Definition at line **128** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **Convert()**.

**uint32\_t Digit[4]**

Definition at line **145** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **Convert()**, and **WriteChar()**.

**uint8\_t LCDBar = BATTERYLEVEL\_FULL**

Definition at line **148** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **BSP\_LCD\_GLASS\_BarLevelConfig()**.

## **LCD\_HandleTypeDef LCDHandle**

```
=====
=====
                                GLASS LCD MAPPING
=====
=====
LCD allows to display informations on six 14-segm
ent digits and 4 bars:

      1          2          3          4          5          6
-----
|\|/| o |\|/| o |\|/| o |\|/| o |\|/|  |\|/|  B
AR3
-- --      -- --      -- --      -- --      -- --      -- --      B
AR2
|/|\| o |/|\| o |/|\| o |/|\| o |/|\|  |/|\|  B
AR1
----- * ----- * ----- * ----- * -----      B
AR0
```

LCD segment mapping:

```
-----
-----A-----
|\|  |  /|  COL  |__|
F H  J  K B
|  \|  /|  |
--G-- --M-- COL  |__|
|  /|  \|  |
E Q  P  N C
```



```

|/  |  \|
-----D----- DP  |__|

```

An LCD character coding is based on the following matrix:

COM	0	1	2	3
SEG(n)	{ E , D , P , N }			
SEG(n+1)	{ M , C , COL , DP }			
SEG(23-n-1)	{ B , A , K , J }			
SEG(23-n)	{ G , F , Q , H }			

with n positive odd number.

The character 'A' for example is:

```

-----
LSB  { 1 , 0 , 0 , 0 }
      { 1 , 1 , 0 , 0 }
      { 1 , 1 , 0 , 0 }
MSB  { 1 , 1 , 0 , 0 }
-----
'A' = F    E    0    0 hexa

```

Definition at line **125** of file [stm32l476g\\_discovery\\_glass\\_lcd.c](#).

Referenced by [BSP\\_LCD\\_GLASS\\_BarLevelConfig\(\)](#), [BSP\\_LCD\\_GLASS\\_BlinkConfig\(\)](#), [BSP\\_LCD\\_GLASS\\_Clear\(\)](#), [BSP\\_LCD\\_GLASS\\_ClearBar\(\)](#), [BSP\\_LCD\\_GLASS\\_Contrast\(\)](#), [BSP\\_LCD\\_GLASS\\_DeInit\(\)](#), [BSP\\_LCD\\_GLASS\\_DisplayBar\(\)](#), [BSP\\_LCD\\_GLASS\\_DisplayChar\(\)](#), [BSP\\_LCD\\_GLASS\\_DisplayStrDeci\(\)](#), [BSP\\_LCD\\_GLASS\\_DisplayString\(\)](#), [BSP\\_LCD\\_GLASS\\_Init\(\)](#), and [WriteChar\(\)](#).

**const uint16\_t** [NumberMap\[10\]](#)

**Initial value:**

```
{  
    0x5F00, 0x4200, 0xF500, 0x6700, 0xEa00, 0xAF00  
    , 0xBF00, 0x04600, 0xFF00, 0xEF00  
}
```

Definition at line **139** of file **stm32l476g\_discovery\_glass\_lcd.c**.

Referenced by **Convert()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Exported Functions

[STM32L476G-DISCOVERY ACCELEROMETER](#)

## Functions

uint8\_t **BSP\_ACCELERO\_Init** (void)

Initialize Accelerometer.

void **BSP\_ACCELERO\_Reset** (void)

Reboot memory content of Accelerometer.

void **BSP\_ACCELERO\_GetXYZ** (int16\_t \*pDataXYZ)

Get XYZ angular accelerations from the Accelerometer.

## Function Documentation

**void** [BSP\\_ACCELERO\\_GetXYZ](#) ( [int16\\_t](#) \* [pDataXYZ](#) )

Get XYZ angular accelerations from the Accelerometer.

**Parameters:**

[pDataXYZ](#) Pointer on 3 angular accelerations table with  
pDataXYZ[0] = X axis, pDataXYZ[1] = Y axis,  
pDataXYZ[2] = Z axis

**Return values:**

[None](#)

Definition at line [180](#) of file [stm32l476g\\_discovery\\_accelerometer.c](#).

References [AccelerometerDrv](#).

**uint8\_t** [BSP\\_ACCELERO\\_Init](#) ( [void](#) )

Initialize Accelerometer.

**Return values:**

[ACCELERO\\_OK](#) or [ACCELERO\\_ERROR](#)

Definition at line [105](#) of file [stm32l476g\\_discovery\\_accelerometer.c](#).

References [ACCELERO\\_ERROR](#), [ACCELERO\\_OK](#), and  
[AccelerometerDrv](#).

**void** [BSP\\_ACCELERO\\_Reset](#) ( [void](#) )

Reboot memory content of Accelerometer.

**Return values:**

**None**

Definition at line **166** of file **stm32l476g\_discovery\_accelerometer.c**.

References **AccelerometerDrv**.

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## BSP Audio Sample Rate

[Exported Constants](#)

## Defines

#define	<b>BSP_AUDIO_FREQUENCY_96K</b>	SAI_AUDIO_FREQUENCY
#define	<b>BSP_AUDIO_FREQUENCY_48K</b>	SAI_AUDIO_FREQUENCY
#define	<b>BSP_AUDIO_FREQUENCY_44K</b>	SAI_AUDIO_FREQUENCY
#define	<b>BSP_AUDIO_FREQUENCY_32K</b>	SAI_AUDIO_FREQUENCY
#define	<b>BSP_AUDIO_FREQUENCY_22K</b>	SAI_AUDIO_FREQUENCY
#define	<b>BSP_AUDIO_FREQUENCY_16K</b>	SAI_AUDIO_FREQUENCY
#define	<b>BSP_AUDIO_FREQUENCY_11K</b>	SAI_AUDIO_FREQUENCY
#define	<b>BSP_AUDIO_FREQUENCY_8K</b>	SAI_AUDIO_FREQUENCY_



## Define Documentation

**#define BSP\_AUDIO\_FREQUENCY\_11K** SAI\_AUDIO\_FREQUENCY

Definition at line **101** of file **stm32l476g\_discovery\_audio.h**.

**#define BSP\_AUDIO\_FREQUENCY\_16K** SAI\_AUDIO\_FREQUENCY

Definition at line **100** of file **stm32l476g\_discovery\_audio.h**.

**#define BSP\_AUDIO\_FREQUENCY\_22K** SAI\_AUDIO\_FREQUENCY

Definition at line **99** of file **stm32l476g\_discovery\_audio.h**.

**#define BSP\_AUDIO\_FREQUENCY\_32K** SAI\_AUDIO\_FREQUENCY

Definition at line **98** of file **stm32l476g\_discovery\_audio.h**.

**#define BSP\_AUDIO\_FREQUENCY\_44K** SAI\_AUDIO\_FREQUENCY

Definition at line **97** of file **stm32l476g\_discovery\_audio.h**.

**#define BSP\_AUDIO\_FREQUENCY\_48K** SAI\_AUDIO\_FREQUENCY

Definition at line **96** of file **stm32l476g\_discovery\_audio.h**.

**#define BSP\_AUDIO\_FREQUENCY\_8K** SAI\_AUDIO\_FREQUENCY

Definition at line **102** of file **stm32l476g\_discovery\_audio.h**.

---

```
#define BSP_AUDIO_FREQUENCY_96K SAI_AUDIO_FREQUENCY
```

Definition at line **95** of file **stm32l476g\_discovery\_audio.h**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Variables](#)

## Exported Variables

[STM32L476G-DISCOVERY AUDIO](#)

## Variables

SAI_HandleTypeDef	<b>BSP_AUDIO_hSai</b>
DFSDM_Filter_HandleTypeDef	<b>BSP_AUDIO_hDfsdmLeftFilter</b>
SAI_HandleTypeDef	<b>BSP_AUDIO_hSai</b>
DFSDM_Filter_HandleTypeDef	<b>BSP_AUDIO_hDfsdmLeftFilter</b>

## Variable Documentation

### DFSDM\_Filter\_HandleTypeDef BSP\_AUDIO\_hDfsdmLeftFilter

Definition at line 266 of file [stm32l476g\\_discovery\\_audio.c](#).

Referenced by [AUDIO\\_DFSDMx\\_DeInit\(\)](#), [AUDIO\\_DFSDMx\\_Init\(\)](#), [BSP\\_AUDIO\\_IN\\_Pause\(\)](#), [BSP\\_AUDIO\\_IN\\_Record\(\)](#), [BSP\\_AUDIO\\_IN\\_Resume\(\)](#), and [BSP\\_AUDIO\\_IN\\_Stop\(\)](#).

### DFSDM\_Filter\_HandleTypeDef BSP\_AUDIO\_hDfsdmLeftFilter

Definition at line 266 of file [stm32l476g\\_discovery\\_audio.c](#).

Referenced by [AUDIO\\_DFSDMx\\_DeInit\(\)](#), [AUDIO\\_DFSDMx\\_Init\(\)](#), [BSP\\_AUDIO\\_IN\\_Pause\(\)](#), [BSP\\_AUDIO\\_IN\\_Record\(\)](#), [BSP\\_AUDIO\\_IN\\_Resume\(\)](#), and [BSP\\_AUDIO\\_IN\\_Stop\(\)](#).

### SAI\_HandleTypeDef BSP\_AUDIO\_hSai

Definition at line 263 of file [stm32l476g\\_discovery\\_audio.c](#).

Referenced by [AUDIO\\_SAIx\\_DeInit\(\)](#), [AUDIO\\_SAIx\\_Init\(\)](#), [BSP\\_AUDIO\\_OUT\\_ChangeAudioConfig\(\)](#), [BSP\\_AUDIO\\_OUT\\_ChangeBuffer\(\)](#), [BSP\\_AUDIO\\_OUT\\_Pause\(\)](#), [BSP\\_AUDIO\\_OUT\\_Play\(\)](#), [BSP\\_AUDIO\\_OUT\\_Resume\(\)](#), [BSP\\_AUDIO\\_OUT\\_SetFrequency\(\)](#), and [BSP\\_AUDIO\\_OUT\\_Stop\(\)](#).

### SAI\_HandleTypeDef BSP\_AUDIO\_hSai

Definition at line 263 of file [stm32l476g\\_discovery\\_audio.c](#).

Referenced by [AUDIO\\_SAIx\\_DeInit\(\)](#), [AUDIO\\_SAIx\\_Init\(\)](#),

**BSP\_AUDIO\_OUT\_ChangeAudioConfig(),  
BSP\_AUDIO\_OUT\_ChangeBuffer(), BSP\_AUDIO\_OUT\_Pause(),  
BSP\_AUDIO\_OUT\_Play(), BSP\_AUDIO\_OUT\_Resume(),  
BSP\_AUDIO\_OUT\_SetFrequency(), and BSP\_AUDIO\_OUT\_Stop().**

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## STM32L476G\_EVAL\_AUDIO\_Exported\_Function

[STM32L476G-DISCOVERY AUDIO](#)

## Functions

uint8_t	<b>BSP_AUDIO_IN_Init</b> (uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnINbr) Initializes micropone related peripherals.
uint8_t	<b>BSP_AUDIO_IN_DeInit</b> (void) De-Initializes microphone related peripherals.
uint8_t	<b>BSP_AUDIO_IN_Record</b> (uint16_t *pbuf, uint32_t size) Starts audio recording.
uint8_t	<b>BSP_AUDIO_IN_SetFrequency</b> (uint32_t AudioFreq) Updates the audio frequency.
void	<b>HAL_DFSDM_FilterRegConvCpltCallback</b> (DFSDM_Filter_HandleTypeDef *hdfsdm_filter) Regular conversion complete callback.
void	<b>HAL_DFSDM_FilterRegConvHalfCpltCallback</b> (DFSDM_Filter_HandleTypeDef *hdfsdm_filter) Half regular conversion complete callback.
void	<b>HAL_DFSDM_FilterErrorCallback</b> (DFSDM_Filter_HandleTypeDef *hdfsdm_filter) Error callback.
uint8_t	<b>BSP_AUDIO_IN_Stop</b> (void) Stops audio recording.
uint8_t	<b>BSP_AUDIO_IN_Pause</b> (void) Pauses the audio file stream.
uint8_t	<b>BSP_AUDIO_IN_Resume</b> (void) Resumes the audio file stream.
void	<b>BSP_AUDIO_IN_RegisterCallbacks</b> (Audio_CallbackTypeDef ErrorCallback, Audio_CallbackTypeDef HalfTransferCallback, Audio_CallbackTypeDef TransferCompleteCallback) register user callback functions



## Function Documentation

**uint8\_t BSP\_AUDIO\_IN\_DeInit ( void )**

De-Initializes microphone related peripherals.

**Return values:**

**BSP** AUDIO status

Definition at line **740** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_DeInit()**, **AUDIO\_ERROR**, and **AUDIO\_OK**.

**uint8\_t BSP\_AUDIO\_IN\_Init ( uint32\_t AudioFreq,  
uint32\_t BitRes,  
uint32\_t ChnINbr  
)**

Initializes microphone related peripherals.

**Note:**

This function assumes that the SAI input clock (through PLL\_M) is already configured and ready to be used.

**Parameters:**

**AudioFreq,:** Audio frequency to be configured for the SAI peripheral.

**BitRes,:** Audio frequency to be configured for the SAI peripheral.

**ChnINbr,:** Audio frequency to be configured for the SAI peripheral.

**Return values:**

## **BSP** AUDIO status

Definition at line **710** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_Init()**, **AUDIO\_ERROR**, **AUDIO\_OK**, **AUDIO\_SAIPLLConfig()**, **AUDIO\_IN\_TypeDef::BitResolution**, **AUDIO\_IN\_TypeDef::CbError**, **AUDIO\_IN\_TypeDef::CbHalfTransfer**, **AUDIO\_IN\_TypeDef::CbTransferComplete**, **AUDIO\_IN\_TypeDef::ChannelNbr**, and **AUDIO\_IN\_TypeDef::Frequency**.

**uint8\_t** **BSP\_AUDIO\_IN\_Pause** ( **void** )

Pauses the audio file stream.

### **Return values:**

**BSP** AUDIO status

Definition at line **905** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**, and **BSP\_AUDIO\_hDfsdmLeftFilter**.

**uint8\_t** **BSP\_AUDIO\_IN\_Record** ( **uint16\_t** \* **pbuf**,  
                                  **uint32\_t** **size**  
                                  **)**

Starts audio recording.

### **Parameters:**

**pbuf,:** Main buffer pointer for the recorded data storing  
**size,:** Current size of the recorded buffer

### **Note:**

The Right channel is start at first with synchro on start of Left channel

**Return values:**

**BSP** AUDIO status

Definition at line **761** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**, **BSP\_AUDIO\_hDfsdmLeftFilter**, **DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR**, **AUDIO\_IN\_TypeDef::LeftRecBuff**, **AUDIO\_IN\_TypeDef::pRecBuf**, and **AUDIO\_IN\_TypeDef::RecSize**.

```
void BSP_AUDIO_IN_RegisterCallbacks ( Audio_CallbackTypeDef
                                     Audio_CallbackTypeDef
                                     Audio_CallbackTypeDef
                                     )
```

register user callback functions

**Parameters:**

<b>ErrorCallback,:</b>	pointer to the error callback function
<b>HalfTransferCallback,:</b>	pointer to the half transfer callback function
<b>TransferCompleteCallback,:</b>	pointer to the transfer complete callback function

**Return values:**

**None**

Definition at line **940** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_IN\_TypeDef::CbError**, **AUDIO\_IN\_TypeDef::CbHalfTransfer**, and

**AUDIO\_IN\_TypeDef::CbTransferComplete.**

**uint8\_t BSP\_AUDIO\_IN\_Resume ( void )**

Resumes the audio file stream.

**Return values:**

**BSP** AUDIO status

Definition at line **920** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**, **BSP\_AUDIO\_hDfstmLeftFilter**, **DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR**, **AUDIO\_IN\_TypeDef::LeftRecBuff**, and **AUDIO\_IN\_TypeDef::RecSize**.

**uint8\_t BSP\_AUDIO\_IN\_SetFrequency ( uint32\_t AudioFreq )**

Updates the audio frequency.

**Parameters:**

**AudioFreq,:** Audio frequency used to record the audio stream.

**Note:**

This API should be called after the **BSP\_AUDIO\_IN\_Init()** to adjust the audio frequency.

**Return values:**

**BSP** AUDIO status

Definition at line **795** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_DeInit()**, **AUDIO\_DFSDMx\_Init()**,

**AUDIO\_ERROR**, **AUDIO\_OK**, and **AUDIO\_SAIPLLConfig()**.

**uint8\_t BSP\_AUDIO\_IN\_Stop ( void )**

Stops audio recording.

**Return values:**

**BSP** AUDIO status

Definition at line **883** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**,  
**BSP\_AUDIO\_hDfsdmLeftFilter**, and  
**AUDIO\_IN\_TypeDef::LeftRecBuff**.

**void HAL\_DFSDM\_FilterErrorCallback ( DFSDM\_Filter\_HandleType**

Error callback.

**Parameters:**

**hdfsdm\_filter** : DFSDM filter handle.

**Return values:**

**None**

Definition at line **870** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_IN\_TypeDef::CbError**.

**void HAL\_DFSDM\_FilterRegConvCpltCallback ( DFSDM\_Filter\_Han**

Regular conversion complete callback.

**Note:**

In interrupt mode, user has to read conversion value in this function using HAL\_DFSDM\_FilterGetRegularValue.

**Parameters:**

**hdfsdm\_filter** : DFSDM filter handle.

**Return values:**

**None**

Definition at line **825** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_IN\_TypeDef::CbTransferComplete**, **DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR**, **AUDIO\_IN\_TypeDef::LeftRecBuff**, **AUDIO\_IN\_TypeDef::pRecBuf**, **AUDIO\_IN\_TypeDef::RecSize**, and **SaturaLH**.

**void HAL\_DFSDM\_FilterRegConvHalfCpltCallback ( DFSDM\_Filter\_**

Half regular conversion complete callback.

**Parameters:**

**hdfsdm\_filter** : DFSDM filter handle.

**Return values:**

**None**

Definition at line **847** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_IN\_TypeDef::CbHalfTransfer**, **DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR**, **AUDIO\_IN\_TypeDef::LeftRecBuff**, **AUDIO\_IN\_TypeDef::pRecBuf**, **AUDIO\_IN\_TypeDef::RecSize**, and **SaturaLH**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Exported Functions

[STM32L476G-DISCOVERY AUDIO](#)

## Functions

uint8_t	<b>BSP_AUDIO_OUT_Init</b> (uint16_t OutputDevice, uint8_t Volume, uint32_t AudioFreq) Configures the audio codec related peripherals.
uint8_t	<b>BSP_AUDIO_OUT_DeInit</b> (void) De-Initializes audio codec related peripherals.
uint8_t	<b>BSP_AUDIO_OUT_Play</b> (uint16_t *pData, uint32_t Size) Starts playing audio stream from a data buffer for a determined size.
uint8_t	<b>BSP_AUDIO_OUT_ChangeBuffer</b> (uint16_t *pData, uint16_t Size) Sends n-Bytes on the SAI interface.
uint8_t	<b>BSP_AUDIO_OUT_Pause</b> (void) This function Pauses the audio file stream.
uint8_t	<b>BSP_AUDIO_OUT_Resume</b> (void) This function Resumes the audio file stream.
uint8_t	<b>BSP_AUDIO_OUT_Stop</b> (uint32_t Option) Stops audio playing and Power down the Audio Codec.
uint8_t	<b>BSP_AUDIO_OUT_SetVolume</b> (uint8_t Volume) Controls the current audio volume level.
uint8_t	<b>BSP_AUDIO_OUT_SetMute</b> (uint32_t Cmd) Enables or disables the MUTE mode by software.
uint8_t	<b>BSP_AUDIO_OUT_SetOutputMode</b> (uint8_t Output) Switch dynamically (while audio file is being played) the output target (speaker or headphone).
uint8_t	<b>BSP_AUDIO_OUT_SetFrequency</b> (uint32_t AudioFreq) Updates the audio frequency.
void	<b>BSP_AUDIO_OUT_ChangeAudioConfig</b> (uint32_t AudioOutOption) Changes the Audio Out Configuration.
void	<b>BSP_AUDIO_OUT_RegisterCallbacks</b> (Audio_CallbackTypeDef ErrorCallback,



	<b>Audio_CallbackTypeDef</b> HalfTransferCallback, <b>Audio_CallbackTypeDef</b> TransferCompleteCallback) register user callback functions
void	<b>HAL_SAI_TxCpltCallback</b> (SAI_HandleTypeDef *hsai) Tx Transfer completed callbacks.
void	<b>HAL_SAI_TxHalfCpltCallback</b> (SAI_HandleTypeDef *hsai) Tx Half Transfer completed callbacks.
void	<b>HAL_SAI_ErrorCallback</b> (SAI_HandleTypeDef *hsai) SAI error callbacks.
uint8_t	<b>BSP_AUDIO_IN_Init</b> (uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr) Initializes micropone related peripherals.
uint8_t	<b>BSP_AUDIO_IN_DeInit</b> (void) De-Initializes microphone related peripherals.
uint8_t	<b>BSP_AUDIO_IN_Record</b> (uint16_t *pData, uint32_t Size) Starts audio recording.
uint8_t	<b>BSP_AUDIO_IN_SetFrequency</b> (uint32_t AudioFreq) Updates the audio frequency.
uint8_t	<b>BSP_AUDIO_IN_Stop</b> (void) Stops audio recording.
uint8_t	<b>BSP_AUDIO_IN_Pause</b> (void) Pauses the audio file stream.
uint8_t	<b>BSP_AUDIO_IN_Resume</b> (void) Resumes the audio file stream.
void	<b>BSP_AUDIO_IN_RegisterCallbacks</b> ( <b>Audio_CallbackTypeDef</b> ErrorCallback, <b>Audio_CallbackTypeDef</b> HalfTransferCallback, <b>Audio_CallbackTypeDef</b> TransferCompleteCallback) register user callback functions

## Function Documentation

**uint8\_t BSP\_AUDIO\_IN\_DeInit ( void )**

De-Initializes microphone related peripherals.

**Return values:**

**BSP** AUDIO status

Definition at line **740** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_DeInit()**, **AUDIO\_ERROR**, and **AUDIO\_OK**.

**uint8\_t BSP\_AUDIO\_IN\_Init ( uint32\_t AudioFreq,  
uint32\_t BitRes,  
uint32\_t ChnINbr  
)**

Initializes microphone related peripherals.

**Note:**

This function assumes that the SAI input clock (through PLL\_M) is already configured and ready to be used.

**Parameters:**

**AudioFreq,:** Audio frequency to be configured for the SAI peripheral.

**BitRes,:** Audio frequency to be configured for the SAI peripheral.

**ChnINbr,:** Audio frequency to be configured for the SAI peripheral.

**Return values:**

## **BSP** AUDIO status

Definition at line **710** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_Init()**, **AUDIO\_ERROR**, **AUDIO\_OK**, **AUDIO\_SAIPLLConfig()**, **AUDIO\_IN\_TypeDef::BitResolution**, **AUDIO\_IN\_TypeDef::CbError**, **AUDIO\_IN\_TypeDef::CbHalfTransfer**, **AUDIO\_IN\_TypeDef::CbTransferComplete**, **AUDIO\_IN\_TypeDef::ChannelNbr**, and **AUDIO\_IN\_TypeDef::Frequency**.

**uint8\_t** **BSP\_AUDIO\_IN\_Pause** ( **void** )

Pauses the audio file stream.

### **Return values:**

**BSP** AUDIO status

Definition at line **905** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**, and **BSP\_AUDIO\_hDfsdmLeftFilter**.

**uint8\_t** **BSP\_AUDIO\_IN\_Record** ( **uint16\_t** \* **pbuf**,  
                                  **uint32\_t** **size**  
                                  **)**

Starts audio recording.

### **Parameters:**

**pbuf,:** Main buffer pointer for the recorded data storing  
**size,:** Current size of the recorded buffer

### **Note:**

The Right channel is start at first with synchro on start of Left channel

**Return values:**

**BSP** AUDIO status

Definition at line **761** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**, **BSP\_AUDIO\_hDfsdmLeftFilter**, **DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR**, **AUDIO\_IN\_TypeDef::LeftRecBuff**, **AUDIO\_IN\_TypeDef::pRecBuf**, and **AUDIO\_IN\_TypeDef::RecSize**.

```
void BSP_AUDIO_IN_RegisterCallbacks ( Audio_CallbackTypeDef
                                     Audio_CallbackTypeDef
                                     Audio_CallbackTypeDef
                                     )
```

register user callback functions

**Parameters:**

<b>ErrorCallback,:</b>	pointer to the error callback function
<b>HalfTransferCallback,:</b>	pointer to the half transfer callback function
<b>TransferCompleteCallback,:</b>	pointer to the transfer complete callback function

**Return values:**

**None**

Definition at line **940** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_IN\_TypeDef::CbError**, **AUDIO\_IN\_TypeDef::CbHalfTransfer**, and

**AUDIO\_IN\_TypeDef::CbTransferComplete.**

**uint8\_t BSP\_AUDIO\_IN\_Resume ( void )**

Resumes the audio file stream.

**Return values:**

**BSP** AUDIO status

Definition at line **920** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**, **BSP\_AUDIO\_hDfstmLeftFilter**, **DEFAULT\_AUDIO\_IN\_CHANNEL\_NBR**, **AUDIO\_IN\_TypeDef::LeftRecBuff**, and **AUDIO\_IN\_TypeDef::RecSize**.

**uint8\_t BSP\_AUDIO\_IN\_SetFrequency ( uint32\_t AudioFreq )**

Updates the audio frequency.

**Parameters:**

**AudioFreq,:** Audio frequency used to record the audio stream.

**Note:**

This API should be called after the **BSP\_AUDIO\_IN\_Init()** to adjust the audio frequency.

**Return values:**

**BSP** AUDIO status

Definition at line **795** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_DFSDMx\_DeInit()**, **AUDIO\_DFSDMx\_Init()**,

**AUDIO\_ERROR**, **AUDIO\_OK**, and **AUDIO\_SAIPLLConfig()**.

**uint8\_t BSP\_AUDIO\_IN\_Stop ( void )**

Stops audio recording.

**Return values:**

**BSP** AUDIO status

Definition at line **883** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_ERROR**, **AUDIO\_OK**,  
**BSP\_AUDIO\_hDfsdmLeftFilter**, and  
**AUDIO\_IN\_TypeDef::LeftRecBuff**.

**void BSP\_AUDIO\_OUT\_ChangeAudioConfig ( uint32\_t AudioOutOption )**

Changes the Audio Out Configuration.

**Parameters:**

**AudioOutOption,:** specifies the audio out new configuration  
This parameter can be any value of **BSP  
Audio Out Option**

**Note:**

This API should be called after the **BSP\_AUDIO\_OUT\_Init()** to  
adjust the audio out configuration.

**Return values:**

**None**

Definition at line **582** of file **stm32l476g\_discovery\_audio.c**.

References **BSP\_AUDIO\_hSai**,  
**BSP\_AUDIO\_OUT\_CIRCULARMODE**, and

## BSP\_AUDIO\_OUT\_STEREOMODE.

```
uint8_t BSP_AUDIO_OUT_ChangeBuffer ( uint16_t * pData,  
                                     uint16_t  Size  
                                     )
```

Sends n-Bytes on the SAI interface.

### Parameters:

**pData,:** pointer on PCM samples buffer  
**Size,:** number of data to be written

### Return values:

**BSP** AUDIO status

Definition at line **399** of file [stm32l476g\\_discovery\\_audio.c](#).

References [AUDIO\\_ERROR](#), [AUDIO\\_OK](#), and [BSP\\_AUDIO\\_hSai](#).

```
uint8_t BSP_AUDIO_OUT_DeInit ( void  )
```

De-Initializes audio codec related peripherals.

### Return values:

**BSP** AUDIO status

Definition at line **353** of file [stm32l476g\\_discovery\\_audio.c](#).

References [AUDIO\\_ERROR](#), [AUDIO\\_OK](#), [AUDIO\\_SAIx\\_DeInit\(\)](#), and [AUDIO\\_OUT\\_TypeDef::AudioDrv](#).

```
uint8_t BSP_AUDIO_OUT_Init ( uint16_t OutputDevice,  
                             uint8_t  Volume,  
                             uint32_t  AudioFreq
```

)

Configures the audio codec related peripherals.

**Parameters:**

- OutputDevice,:** OUTPUT\_DEVICE\_SPEAKER, OUTPUT\_DEVICE\_HEADPHONE, or OUTPUT\_DEVICE\_BOTH.
- Volume,:** Initial volume level (from 0 (Mute) to 100 (Max))
- AudioFreq,:** Audio frequency used to play the audio stream.ion.

**Return values:**

**BSP** AUDIO status

**Note:**

The SAI PLL input clock must be configure in the user application. The SAI PLL configuration done within this function assumes that the SAI PLL input clock runs at 8 MHz.

Definition at line **301** of file [stm32l476g\\_discovery\\_audio.c](#).

References [AUDIO\\_CODEC\\_Reset\(\)](#), [AUDIO\\_ERROR](#), [AUDIO\\_I2C\\_ADDRESS](#), [AUDIO\\_OK](#), [AUDIO\\_SAIPLLConfig\(\)](#), [AUDIO\\_SAIx\\_Init\(\)](#), [AUDIO\\_OUT\\_TypeDef::AudioDrv](#), [BSP\\_AUDIO\\_OUT\\_SetVolume\(\)](#), [AUDIO\\_OUT\\_TypeDef::CbError](#), [AUDIO\\_OUT\\_TypeDef::CbHalfTransfer](#), and [AUDIO\\_OUT\\_TypeDef::CbTransferComplete](#).

**uint8\_t BSP\_AUDIO\_OUT\_Pause (void )**

This function Pauses the audio file stream.

In case of using DMA, the DMA Pause feature is used.





**Audio\_CallbackTypeDef**

)

register user callback functions

**Parameters:**

<b>ErrorCallback,:</b>	pointer to the error callback function
<b>HalfTransferCallback,:</b>	pointer to the half transfer callback function
<b>TransferCompleteCallback,:</b>	pointer to the transfer complete callback function

**Return values:**

**None**

Definition at line **642** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_OUT\_TypeDef::CbError**, **AUDIO\_OUT\_TypeDef::CbHalfTransfer**, and **AUDIO\_OUT\_TypeDef::CbTransferComplete**.

**uint8\_t BSP\_AUDIO\_OUT\_Resume ( void )**

This function Resumes the audio file stream.

**Note:**

When calling **BSP\_AUDIO\_OUT\_Pause()** function for pause, only **BSP\_AUDIO\_OUT\_Resume()** function should be called for resume (use of **BSP\_AUDIO\_OUT\_Play()** function for resume could lead to unexpected behavior).

**Return values:**

**BSP** AUDIO status

Definition at line **444** of file **stm32l476g\_discovery\_audio.c**.

References [AUDIO\\_ERROR](#), [AUDIO\\_I2C\\_ADDRESS](#), [AUDIO\\_OK](#), [AUDIO\\_OUT\\_TypeDef::AudioDrv](#), and [BSP\\_AUDIO\\_hSai](#).

**uint8\_t** [BSP\\_AUDIO\\_OUT\\_SetFrequency](#) ( **uint32\_t** [AudioFreq](#) )

Updates the audio frequency.

**Parameters:**

**AudioFreq,:** Audio frequency used to play the audio stream.

**Note:**

The SAI PLL input clock must be configure in the user application. The SAI PLL configuration done within this function assumes that the SAI PLL input clock runs at 8 MHz.

**Return values:**

**BSP** AUDIO status

Definition at line [553](#) of file [stm32l476g\\_discovery\\_audio.c](#).

References [AUDIO\\_ERROR](#), [AUDIO\\_OK](#), [AUDIO\\_SAIPLLConfig\(\)](#), [BSP\\_AUDIO\\_hSai](#), and [SAIClockDivider](#).

**uint8\_t** [BSP\\_AUDIO\\_OUT\\_SetMute](#) ( **uint32\_t** [Cmd](#) )

Enables or disables the MUTE mode by software.

**Parameters:**

**Cmd,:** Could be [AUDIO\\_MUTE\\_ON](#) to mute sound or [AUDIO\\_MUTE\\_OFF](#) to unmute the codec and restore previous volume level.

**Return values:**

**BSP** AUDIO status

Definition at line **516** of file `stm32l476g_discovery_audio.c`.

References `AUDIO_ERROR`, `AUDIO_I2C_ADDRESS`, `AUDIO_OK`, and `AUDIO_OUT_TypeDef::AudioDrv`.

**uint8\_t** `BSP_AUDIO_OUT_SetOutputMode ( uint8_t Output )`

Switch dynamically (while audio file is being played) the output target (speaker or headphone).

**Parameters:**

**Output,:** The audio output target:  
OUTPUT\_DEVICE\_SPEAKER,  
OUTPUT\_DEVICE\_HEADPHONE or  
OUTPUT\_DEVICE\_BOTH

**Return values:**

**BSP** AUDIO status

Definition at line **534** of file `stm32l476g_discovery_audio.c`.

References `AUDIO_ERROR`, `AUDIO_I2C_ADDRESS`, `AUDIO_OK`, and `AUDIO_OUT_TypeDef::AudioDrv`.

**uint8\_t** `BSP_AUDIO_OUT_SetVolume ( uint8_t Volume )`

Controls the current audio volume level.

**Parameters:**

**Volume,:** Volume level to be set in percentage from 0% to 100% (0 for Mute and 100 for Max volume level).

**Return values:**

**BSP** AUDIO status

Definition at line **499** of file `stm32l476g_discovery_audio.c`.

References `AUDIO_ERROR`, `AUDIO_I2C_ADDRESS`, `AUDIO_OK`, and `AUDIO_OUT_TypeDef::AudioDrv`.

Referenced by `BSP_AUDIO_OUT_Init()`.

**uint8\_t BSP\_AUDIO\_OUT\_Stop ( uint32\_t Option )**

Stops audio playing and Power down the Audio Codec.

**Parameters:**

**Option,:** could be one of the following parameters

- `CODEC_PDWN_SW`: for software power off (by writing registers). Then no need to reconfigure the Codec after power on.
- `CODEC_PDWN_HW`: completely shut down the codec (physically). Then need to reconfigure the Codec after power on.

**Return values:**

**BSP** AUDIO status

Definition at line **470** of file `stm32l476g_discovery_audio.c`.

References `AUDIO_ERROR`, `AUDIO_I2C_ADDRESS`, `AUDIO_OK`, `AUDIO_OUT_TypeDef::AudioDrv`, and `BSP_AUDIO_hSai`.

**void HAL\_SAI\_ErrorCallback ( SAI\_HandleTypeDef \* hsai )**

SAI error callbacks.

**Parameters:**

**hsai,:** SAI handle

**Return values:**

**None**

Definition at line **684** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_OUT\_TypeDef::CbError**.

**void HAL\_SAI\_TxCpltCallback ( SAI\_HandleTypeDef \* **hsai** )**

Tx Transfer completed callbacks.

**Parameters:**

**hsai**,: SAI handle

**Return values:**

**None**

Definition at line **656** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_OUT\_TypeDef::CbTransferComplete**.

**void HAL\_SAI\_TxHalfCpltCallback ( SAI\_HandleTypeDef \* **hsai** )**

Tx Half Transfer completed callbacks.

**Parameters:**

**hsai**,: SAI handle

**Return values:**

**None**

Definition at line **670** of file **stm32l476g\_discovery\_audio.c**.

References **AUDIO\_OUT\_TypeDef::CbHalfTransfer**.

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## BSP Audio Out Option

[Exported Constants](#)



## Defines

#define	<b>BSP_AUDIO_OUT_CIRCULARMODE</b>	((uint32_t)0x00000000)
	/* BUFFER CIRCULAR MODE */	
#define	<b>BSP_AUDIO_OUT_NORMALMODE</b>	((uint32_t)0x00000002)
	/* BUFFER NORMAL MODE */	
#define	<b>BSP_AUDIO_OUT_STEREOMODE</b>	((uint32_t)0x00000004)
	STEREO MODE */	
#define	<b>BSP_AUDIO_OUT_MONOMODE</b>	((uint32_t)0x00000008)
	MONO MODE */	

---

## Define Documentation

**#define BSP\_AUDIO\_OUT\_CIRCULARMODE** ((uint32\_t)0x00000000

Definition at line **84** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **BSP\_AUDIO\_OUT\_ChangeAudioConfig()**.

**#define BSP\_AUDIO\_OUT\_MONOMODE** ((uint32\_t)0x00000008) /\*

Definition at line **87** of file **stm32l476g\_discovery\_audio.h**.

**#define BSP\_AUDIO\_OUT\_NORMALMODE** ((uint32\_t)0x00000002)

Definition at line **85** of file **stm32l476g\_discovery\_audio.h**.

**#define BSP\_AUDIO\_OUT\_STEREOMODE** ((uint32\_t)0x00000004)

Definition at line **86** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **BSP\_AUDIO\_OUT\_ChangeAudioConfig()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Exported Functions

[STM32L476G-DISCOVERY Common](#)

## Functions

uint32_t	<b>BSP_GetVersion</b> (void) This method returns the STM32L476 DISCOVERY BSP Driver revision.
<b>SupplyMode_TypeDef</b>	<b>BSP_SupplyModeDetection</b> (void) This method returns the STM32L476 DISCOVERY supply mode.
void	<b>BSP_LED_Init</b> ( <b>Led_TypeDef</b> Led) Configures LED GPIOs.
void	<b>BSP_LED_DeInit</b> ( <b>Led_TypeDef</b> Led) Unconfigures LED GPIOs.
void	<b>BSP_LED_On</b> ( <b>Led_TypeDef</b> Led) Turns selected LED On.
void	<b>BSP_LED_Off</b> ( <b>Led_TypeDef</b> Led) Turns selected LED Off.
void	<b>BSP_LED_Toggle</b> ( <b>Led_TypeDef</b> Led) Toggles the selected LED.
uint8_t	<b>BSP_JOY_Init</b> ( <b>JOYMode_TypeDef</b> Joy_Mode) Configures all buttons of the joystick in GPIO or EXTI modes.
void	<b>BSP_JOY_DeInit</b> (void) Unconfigures all GPIOs used as buttons of the joystick.
<b>JOYState_TypeDef</b>	<b>BSP_JOY_GetState</b> (void) Returns the current joystick status.

## Function Documentation

**uint32\_t** **BSP\_GetVersion** ( void )

This method returns the STM32L476 DISCOVERY BSP Driver revision.

**Return values:**

**version** : 0xXYZR (8bits for each decimal, R for RC)

Definition at line **286** of file **stm32l476g\_discovery.c**.

References **\_\_STM32L476G\_DISCOVERY\_BSP\_VERSION**.

**void** **BSP\_JOY\_DeInit** ( void )

Unconfigures all GPIOs used as buttons of the joystick.

**Return values:**

**None.**

Definition at line **512** of file **stm32l476g\_discovery.c**.

References **JOY\_PIN**, **JOY\_PORT**, **JOY\_SEL**, **JOYn**, and **JOYx\_GPIO\_CLK\_ENABLE**.

**JOYState\_TypeDef** **BSP\_JOY\_GetState** ( void )

Returns the current joystick status.

**Return values:**

**Code** of the joystick key pressed This code can be one of the following values:

- **JOY\_NONE**

- JOY\_SEL
- JOY\_DOWN
- JOY\_LEFT
- JOY\_RIGHT
- JOY\_UP

Definition at line **537** of file [stm32l476g\\_discovery.c](#).

References [JOY\\_NONE](#), [JOY\\_PIN](#), [JOY\\_PORT](#), [JOY\\_SEL](#), and [JOYn](#).

**uint8\_t BSP\_JOY\_Init ( JOYMode\_TypeDef Joy\_Mode )**

Configures all buttons of the joystick in GPIO or EXTI modes.

**Parameters:**

**Joy\_Mode,:** Joystick mode. This parameter can be one of the following values:

- JOY\_MODE\_GPIO: Joystick pins will be used as simple IOs
- JOY\_MODE\_EXTI: Joystick pins will be connected to EXTI line with interrupt generation capability

**Return values:**

**HAL\_OK,:** if all initializations are OK. Other value if error.

Definition at line **472** of file [stm32l476g\\_discovery.c](#).

References [JOY\\_IRQn](#), [JOY\\_MODE\\_EXTI](#), [JOY\\_MODE\\_GPIO](#), [JOY\\_PIN](#), [JOY\\_PORT](#), [JOY\\_SEL](#), [JOYn](#), and [JOYx\\_GPIO\\_CLK\\_ENABLE](#).

**void BSP\_LED\_DeInit ( Led\_TypeDef Led )**

Unconfigures LED GPIOs.

**Parameters:**

**Led,:** Specifies the Led to be unconfigured. This parameter can be one of following parameters:

- LED4
- LED5

**Return values:**

**None**

Definition at line **383** of file **stm32l476g\_discovery.c**.

References **LED\_PIN**, **LED\_PORT**, and **LEDx\_GPIO\_CLK\_ENABLE**.

**void BSP\_LED\_Init ( Led\_TypeDef Led )**

Configures LED GPIOs.

**Parameters:**

**Led,:** Specifies the Led to be configured. This parameter can be one of following parameters:

- LED4
- LED5

**Return values:**

**None**

Definition at line **346** of file **stm32l476g\_discovery.c**.

References **LED\_PIN**, **LED\_PORT**, and **LEDx\_GPIO\_CLK\_ENABLE**.

**void BSP\_LED\_Off ( Led\_TypeDef Led )**

Turns selected LED Off.

**Parameters:**

**Led,:** Specifies the Led to be set off. This parameter can be one of following parameters:

- LED4
- LED5

**Return values:**

**None**

Definition at line **434** of file **stm32l476g\_discovery.c**.

References **LED\_PIN**, and **LED\_PORT**.

**void BSP\_LED\_On ( Led\_TypeDef Led )**

Turns selected LED On.

**Parameters:**

**Led,:** Specifies the Led to be set on. This parameter can be one of following parameters:

- LED4
- LED5

**Return values:**

**None**

Definition at line **410** of file **stm32l476g\_discovery.c**.

References **LED\_PIN**, and **LED\_PORT**.

**void BSP\_LED\_Toggle ( Led\_TypeDef Led )**



Toggles the selected LED.

**Parameters:**

**Led,:** Specifies the Led to be toggled. This parameter can be one of following parameters:

- LED4
- LED5

**Return values:**

**None**

Definition at line **458** of file **stm32l476g\_discovery.c**.

References **LED\_PIN**, and **LED\_PORT**.

## **SupplyMode\_TypeDef BSP\_SupplyModeDetection ( void )**

This method returns the STM32L476 DISCOVERY supply mode.

**Return values:**

**Code** of current supply mode This code can be one of following:

- SUPPLY\_MODE\_EXTERNAL
- SUPPLY\_MODE\_BATTERY

Definition at line **298** of file **stm32l476g\_discovery.c**.

References **BATTERY\_DETECTION\_GPIO\_CLK\_ENABLE**, **BATTERY\_DETECTION\_GPIO\_PORT**, **BATTERY\_DETECTION\_PIN**, **SUPPLY\_MODE\_BATTERY**, **SUPPLY\_MODE\_ERROR**, and **SUPPLY\_MODE\_EXTERNAL**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Exported Functions

[STM32L476G-DISCOVERY GYROSCOPE](#)

## Functions

uint8_t	<b>BSP_GYRO_Init</b> (void)	Initialize Gyroscope.
void	<b>BSP_GYRO_DeInit</b> (void)	DeInitialize Gyroscope.
void	<b>BSP_GYRO_LowPower</b> (void)	Put Gyroscope in low power mode.
uint8_t	<b>BSP_GYRO_ReadID</b> (void)	Read ID of Gyroscope component.
void	<b>BSP_GYRO_Reset</b> (void)	Reboot memory content of Gyroscope.
void	<b>BSP_GYRO_ITConfig</b> (GYRO_InterruptConfigTypeDef *pIntConfig)	Configure Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_EnableIT</b> (uint8_t IntPin)	Enable Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_DisableIT</b> (uint8_t IntPin)	Disable Gyroscope interrupts (INT1 or INT2).
void	<b>BSP_GYRO_GetXYZ</b> (float *pfData)	Get XYZ angular acceleration from the Gyroscope.

## Function Documentation

**void** **BSP\_GYRO\_DeInit** ( void )

DeInitialize Gyroscope.

**Return values:**

**None**

Definition at line **163** of file **stm32l476g\_discovery\_gyroscope.c**.

References **GYRO\_IO\_DeInit()**.

**void** **BSP\_GYRO\_DisableIT** ( uint8\_t **IntPin** )

Disable Gyroscope interrupts (INT1 or INT2).

**Parameters:**

**IntPin,:** Interrupt pin This parameter can be:

- L3GD20\_INT1
- L3GD20\_INT2

**Return values:**

**None**

Definition at line **262** of file **stm32l476g\_discovery\_gyroscope.c**.

References **GyroscopeDrv**.

**void** **BSP\_GYRO\_EnableIT** ( uint8\_t **IntPin** )

Enable Gyroscope interrupts (INT1 or INT2).

**Parameters:**

**IntPin,:** Interrupt pin This parameter can be:

- L3GD20\_INT1
- L3GD20\_INT2

**Return values:**

**None**

Definition at line **246** of file [stm32l476g\\_discovery\\_gyroscope.c](#).

References [GyroscopeDrv](#).

**void BSP\_GYRO\_GetXYZ ( float \* **pfData** )**

Get XYZ angular acceleration from the Gyroscope.

**Parameters:**

**pfData,:** pointer on floating array

**Return values:**

**None**

Definition at line **275** of file [stm32l476g\\_discovery\\_gyroscope.c](#).

References [GyroscopeDrv](#).

**uint8\_t BSP\_GYRO\_Init ( void )**

Initialize Gyroscope.

**Return values:**

**GYRO\_OK** or GYRO\_ERROR

Definition at line **105** of file [stm32l476g\\_discovery\\_gyroscope.c](#).

References [GYRO\\_ERROR](#), [GYRO\\_OK](#), and [GyroscopeDrv](#).

**void BSP\_GYRO\_ITConfig ( GYRO\_InterruptConfigTypeDef \* pIntC**

Configure Gyroscope interrupts (INT1 or INT2).

**Parameters:**

**pIntConfig,:** pointer to a GYRO\_InterruptConfigTypeDef structure that contains the configuration setting for the L3GD20 Interrupt.

**Return values:**

**None**

Definition at line **222** of file **stm32l476g\_discovery\_gyroscope.c**.

References **GyroscopeDrv**.

**void BSP\_GYRO\_LowPower ( void )**

Put Gyroscope in low power mode.

**Return values:**

**None**

Definition at line **173** of file **stm32l476g\_discovery\_gyroscope.c**.

References **GyroscopeDrv**.

**uint8\_t BSP\_GYRO\_ReadID ( void )**

Read ID of Gyroscope component.

**Return values:**

**ID**

Definition at line **193** of file **stm32l476g\_discovery\_gyroscope.c**.

References **GyroscopeDrv**.

**void BSP\_GYRO\_Reset ( void )**

Reboot memory content of Gyroscope.

**Return values:**

**None**

Definition at line **208** of file **stm32l476g\_discovery\_gyroscope.c**.

References **GyroscopeDrv**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Exported Functions

[STM32L476G-DISCOVERY IDD](#)



## Functions

uint8_t	<b>BSP_IDD_Init</b> (void)	Configures IDD measurement component.
void	<b>BSP_IDD_DeInit</b> (void)	Unconfigures IDD measurement component.
void	<b>BSP_IDD_Reset</b> (void)	Reset Idd measurement component.
void	<b>BSP_IDD_LowPower</b> (void)	Turn Idd measurement component in low power (standby/sleep) mode.
void	<b>BSP_IDD_StartMeasure</b> (void)	Start Measurement campaign.
void	<b>BSP_IDD_Config</b> (IDD_ConfigTypeDef IddConfig)	Configure Idd component.
void	<b>BSP_IDD_GetValue</b> (uint32_t *IddValue)	Get Idd current value.
void	<b>BSP_IDD_EnableIT</b> (void)	Enable Idd interrupt that warn end of measurement.
void	<b>BSP_IDD_ClearIT</b> (void)	Clear Idd interrupt that warn end of measurement.
uint8_t	<b>BSP_IDD_GetITStatus</b> (void)	Get Idd interrupt status.
void	<b>BSP_IDD_DisableIT</b> (void)	Disable Idd interrupt that warn end of measurement.
uint8_t	<b>BSP_IDD_ErrorGetCode</b> (void)	Get Error Code .
void	<b>BSP_IDD_ErrorEnableIT</b> (void)	Enable error interrupt that warn end of measurement.
void	<b>BSP_IDD_ErrorClearIT</b> (void)	Clear Error interrupt that warn end of measurement.
uint8_t	<b>BSP_IDD_ErrorGetITStatus</b> (void)	

Get Error interrupt status.

void **BSP\_IDD\_ErrorDisableIT** (void)

Disable Error interrupt.

void **BSP\_IDD\_WakeUp** (void)

Wake up Idd measurement component.

## Function Documentation

**void** **BSP\_IDD\_ClearIT** ( **void** )

Clear Idd interrupt that warn end of measurement.

**Return values:**

**None**

Definition at line **239** of file **stm32l476g\_discovery\_idd.c**.

References **IDD\_I2C\_ADDRESS**, and **IddDrv**.

**void** **BSP\_IDD\_Config** ( **IDD\_ConfigTypeDef** **IddConfig** )

Configure Idd component.

**Parameters:**

**IddConfig**,: structure of idd parameters

**Return values:**

**None**

Definition at line **202** of file **stm32l476g\_discovery\_idd.c**.

References **IDD\_I2C\_ADDRESS**, and **IddDrv**.

Referenced by **BSP\_IDD\_Init()**.

**void** **BSP\_IDD\_DeInit** ( **void** )

Unconfigures IDD measurement component.

**Return values:**

**IDD\_OK** if no problem during deinitialization

Definition at line **153** of file **stm32l476g\_discovery\_idd.c**.

References **IDD\_I2C\_ADDRESS**, and **IddDrv**.

**void BSP\_IDD\_DisableIT ( void )**

Disable Idd interrupt that warn end of measurement.

**Return values:**

**None**

Definition at line **267** of file **stm32l476g\_discovery\_idd.c**.

References **IDD\_I2C\_ADDRESS**, and **IddDrv**.

**void BSP\_IDD\_EnableIT ( void )**

Enable Idd interrupt that warn end of measurement.

**Return values:**

**None**

Definition at line **227** of file **stm32l476g\_discovery\_idd.c**.

References **IDD\_I2C\_ADDRESS**, and **IddDrv**.

**void BSP\_IDD\_ErrorClearIT ( void )**

Clear Error interrupt that warn end of measurement.

**Return values:**

**None**

Definition at line **322** of file [stm32l476g\\_discovery\\_idd.c](#).

References [IDD\\_I2C\\_ADDRESS](#), and [IddDrv](#).

**void BSP\_IDD\_ErrorDisableIT ( void )**

Disable Error interrupt.

**Return values:**

**None**

Definition at line **350** of file [stm32l476g\\_discovery\\_idd.c](#).

References [IDD\\_I2C\\_ADDRESS](#), and [IddDrv](#).

**void BSP\_IDD\_ErrorEnableIT ( void )**

Enable error interrupt that warn end of measurement.

**Return values:**

**None**

Definition at line **310** of file [stm32l476g\\_discovery\\_idd.c](#).

References [IDD\\_I2C\\_ADDRESS](#), and [IddDrv](#).

**uint8\_t BSP\_IDD\_ErrorGetCode ( void )**

Get Error Code .

**Return values:**

**Error** code or error status

Definition at line **279** of file [stm32l476g\\_discovery\\_idd.c](#).

References [IDD\\_ERROR](#), [IDD\\_I2C\\_ADDRESS](#), and [IddDrv](#).

**uint8\_t BSP\_IDD\_ErrorGetITStatus ( void )**

Get Error interrupt status.

**Return values:**

**Status**

Definition at line [334](#) of file [stm32l476g\\_discovery\\_idd.c](#).

References [IDD\\_I2C\\_ADDRESS](#), and [IddDrv](#).

**uint8\_t BSP\_IDD\_GetITStatus ( void )**

Get Idd interrupt status.

**Return values:**

**status**

Definition at line [251](#) of file [stm32l476g\\_discovery\\_idd.c](#).

References [IDD\\_ERROR](#), [IDD\\_I2C\\_ADDRESS](#), and [IddDrv](#).

**void BSP\_IDD\_GetValue ( uint32\_t \* IddValue )**

Get Idd current value.

**Parameters:**

**IddValue,:** Pointer on u32 to store Idd. Value unit is 10 nA.

**Return values:**

**None**

Definition at line **215** of file [stm32l476g\\_discovery\\_idd.c](#).

References [IDD\\_I2C\\_ADDRESS](#), and [IddDrv](#).

**uint8\_t BSP\_IDD\_Init ( void )**

Configures IDD measurement component.

**Return values:**

**IDD\_OK** if no problem during initialization

Definition at line **91** of file [stm32l476g\\_discovery\\_idd.c](#).

References [BSP\\_IDD\\_Config\(\)](#), [DISCOVERY\\_IDD\\_AMPLI\\_GAIN](#), [DISCOVERY\\_IDD\\_SHUNT0\\_STABDELAY](#), [DISCOVERY\\_IDD\\_SHUNT0\\_VALUE](#), [DISCOVERY\\_IDD\\_SHUNT1\\_STABDELAY](#), [DISCOVERY\\_IDD\\_SHUNT1\\_VALUE](#), [DISCOVERY\\_IDD\\_SHUNT2\\_STABDELAY](#), [DISCOVERY\\_IDD\\_SHUNT2\\_VALUE](#), [DISCOVERY\\_IDD\\_SHUNT4\\_STABDELAY](#), [DISCOVERY\\_IDD\\_SHUNT4\\_VALUE](#), [DISCOVERY\\_IDD\\_VDD\\_MIN](#), [IDD\\_ERROR](#), [IDD\\_I2C\\_ADDRESS](#), [IDD\\_OK](#), and [IddDrv](#).

**void BSP\_IDD\_LowPower ( void )**

Turn Idd measurement component in low power (standby/sleep) mode.

**Return values:**

**None**

Definition at line **177** of file [stm32l476g\\_discovery\\_idd.c](#).

References [IDD\\_I2C\\_ADDRESS](#), and [IddDrv](#).

**void BSP\_IDD\_Reset ( void )**

Reset Idd measurement component.

**Return values:**

**None**

Definition at line **165** of file **stm32l476g\_discovery\_idd.c**.

References **IDD\_I2C\_ADDRESS**, and **IddDrv**.

**void BSP\_IDD\_StartMeasure ( void )**

Start Measurement campaign.

**Return values:**

**None**

Definition at line **189** of file **stm32l476g\_discovery\_idd.c**.

References **IDD\_I2C\_ADDRESS**, and **IddDrv**.

**void BSP\_IDD\_WakeUp ( void )**

Wake up Idd measurement component.

**Return values:**

**None**

Definition at line **362** of file **stm32l476g\_discovery\_idd.c**.

References **IDD\_I2C\_ADDRESS**, and **IddDrv**.



BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Exported Functions

[STM32L476G-DISCOVERY GLASS LCD](#)

## Functions

void	<b>BSP_LCD_GLASS_Init</b> (void)	Initialize the LCD GLASS relative GPIO port IOs and LCD peripheral.
void	<b>BSP_LCD_GLASS_DeInit</b> (void)	DeInitialize the LCD GLASS relative GPIO port IOs and LCD peripheral.
void	<b>BSP_LCD_GLASS_BlinkConfig</b> (uint32_t BlinkMode, uint32_t BlinkFrequency)	Configure the LCD Blink mode and Blink frequency.
void	<b>BSP_LCD_GLASS_Contrast</b> (uint32_t Contrast)	Configure the LCD contrast.
void	<b>BSP_LCD_GLASS_DisplayBar</b> (uint32_t BarId)	Display one or several bar in LCD frame buffer.
void	<b>BSP_LCD_GLASS_ClearBar</b> (uint32_t BarId)	Clear one or several bar in LCD frame buffer.
void	<b>BSP_LCD_GLASS_BarLevelConfig</b> (uint8_t BarLevel)	Configure the bar level on LCD by writing bar value in LCD frame buffer.
void	<b>BSP_LCD_GLASS_DisplayChar</b> (uint8_t *ch, <b>Point_Typedef</b> Point, <b>DoublePoint_Typedef</b> Colon, <b>DigitPosition_Typedef</b> Position)	Write a character in the LCD RAM buffer.
void	<b>BSP_LCD_GLASS_DisplayString</b> (uint8_t *ptr)	Write a character string in the LCD RAM buffer.
void	<b>BSP_LCD_GLASS_DisplayStrDeci</b> (uint16_t *ptr)	Write a character string with decimal point in the LCD RAM buffer.
void	<b>BSP_LCD_GLASS_Clear</b> (void)	Clear the whole LCD RAM buffer.
void	<b>BSP_LCD_GLASS_ScrollSentence</b> (uint8_t *ptr, uint16_t nScroll, uint16_t ScrollSpeed)	

Display a string in scrolling mode.

---

## Function Documentation

**void** **BSP\_LCD\_GLASS\_BarLevelConfig** ( **uint8\_t** **BarLevel** )

Configure the bar level on LCD by writing bar value in LCD frame buffer.

### Parameters:

- BarLevel,:** specifies the LCD GLASS Battery Level. This parameter can be one of the following values:
- BATTERYLEVEL\_OFF: LCD GLASS Battery Empty
  - BATTERYLEVEL\_1\_4: LCD GLASS Battery 1/4 Full
  - BATTERYLEVEL\_1\_2: LCD GLASS Battery 1/2 Full
  - BATTERYLEVEL\_3\_4: LCD GLASS Battery 3/4 Full
  - BATTERYLEVEL\_FULL: LCD GLASS Battery Full

### Return values:

**None**

Definition at line **377** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **BATTERYLEVEL\_1\_2**, **BATTERYLEVEL\_1\_4**, **BATTERYLEVEL\_3\_4**, **BATTERYLEVEL\_FULL**, **BATTERYLEVEL\_OFF**, **LCD\_BAR0\_2\_COM**, **LCD\_BAR0\_SEG**, **LCD\_BAR1\_3\_COM**, **LCD\_BAR1\_SEG**, **LCD\_BAR2\_SEG**, **LCD\_BAR3\_SEG**, **LCDBar**, and **LCDHandle**.

**void** **BSP\_LCD\_GLASS\_BlinkConfig** ( **uint32\_t** **BlinkMode**,  
                                  **uint32\_t** **BlinkFrequency**  
                                  **)**

Configure the LCD Blink mode and Blink frequency.

**Parameters:**

**BlinkMode,:** specifies the LCD blink mode. This parameter can be one of the following values:

- LCD\_BLINKMODE\_OFF: Blink disabled
- LCD\_BLINKMODE\_SEG0\_COM0: Blink enabled on SEG[0], COM[0] (1 pixel)
- LCD\_BLINKMODE\_SEG0\_ALLCOM: Blink enabled on SEG[0], all COM (up to 8 pixels according to the programmed duty)
- LCD\_BLINKMODE\_ALLSEG\_ALLCOM: Blink enabled on all SEG and all COM (all pixels)

**BlinkFrequency,:** specifies the LCD blink frequency.

- LCD\_BLINKFREQUENCY\_DIV8: The Blink frequency =  $f_{Lcd}/8$
- LCD\_BLINKFREQUENCY\_DIV16: The Blink frequency =  $f_{Lcd}/16$
- LCD\_BLINKFREQUENCY\_DIV32: The Blink frequency =  $f_{Lcd}/32$
- LCD\_BLINKFREQUENCY\_DIV64: The Blink frequency =  $f_{Lcd}/64$
- LCD\_BLINKFREQUENCY\_DIV128: The Blink frequency =  $f_{Lcd}/128$
- LCD\_BLINKFREQUENCY\_DIV256: The Blink frequency =  $f_{Lcd}/256$
- LCD\_BLINKFREQUENCY\_DIV512: The Blink frequency =  $f_{Lcd}/512$
- LCD\_BLINKFREQUENCY\_DIV1024: The Blink frequency =  $f_{Lcd}/1024$

**Return values:**

**None**

Definition at line **234** of file [stm32l476g\\_discovery\\_glass\\_lcd.c](#).

References [LCDHandle](#).

**void BSP\_LCD\_GLASS\_Clear ( void )**

Clear the whole LCD RAM buffer.

**Return values:**

**None**

Definition at line **528** of file [stm32l476g\\_discovery\\_glass\\_lcd.c](#).

References [LCDHandle](#).

Referenced by [BSP\\_LCD\\_GLASS\\_Init\(\)](#), and  
[BSP\\_LCD\\_GLASS\\_ScrollSentence\(\)](#).

**void BSP\_LCD\_GLASS\_ClearBar ( uint32\_t BarId )**

Clear one or several bar in LCD frame buffer.

**Parameters:**

**BarId,:** specifies the LCD GLASS Bar to display This parameter can be combination of one of the following values:

- LCD\_BAR\_0: LCD GLASS Bar 0
- LCD\_BAR\_1: LCD GLASS Bar 1
- LCD\_BAR\_2: LCD GLASS Bar 2
- LCD\_BAR\_3: LCD GLASS Bar 3

**Return values:**

**None**

Definition at line **322** of file [stm32l476g\\_discovery\\_glass\\_lcd.c](#).

References [LCD\\_BAR0\\_2\\_COM](#), [LCD\\_BAR0\\_SEG](#), [LCD\\_BAR1\\_3\\_COM](#), [LCD\\_BAR1\\_SEG](#), [LCD\\_BAR2\\_SEG](#), [LCD\\_BAR3\\_SEG](#), [LCD\\_BAR\\_0](#), [LCD\\_BAR\\_1](#), [LCD\\_BAR\\_2](#), [LCD\\_BAR\\_3](#), and [LCDHandle](#).

**void [BSP\\_LCD\\_GLASS\\_Contrast](#) ( uint32\_t [Contrast](#) )**

Configure the LCD contrast.

**Parameters:**

**[Contrast](#),** specifies the LCD contrast value. This parameter can be one of the following values:

- [LCD\\_CONTRASTLEVEL\\_0](#): Maximum Voltage = 2.60V
- [LCD\\_CONTRASTLEVEL\\_1](#): Maximum Voltage = 2.73V
- [LCD\\_CONTRASTLEVEL\\_2](#): Maximum Voltage = 2.86V
- [LCD\\_CONTRASTLEVEL\\_3](#): Maximum Voltage = 2.99V
- [LCD\\_CONTRASTLEVEL\\_4](#): Maximum Voltage = 3.12V
- [LCD\\_CONTRASTLEVEL\\_5](#): Maximum Voltage = 3.25V
- [LCD\\_CONTRASTLEVEL\\_6](#): Maximum Voltage = 3.38V
- [LCD\\_CONTRASTLEVEL\\_7](#): Maximum Voltage = 3.51V

**Return values:**

**[None](#)**

Definition at line [253](#) of file [stm32l476g\\_discovery\\_glass\\_lcd.c](#).

References [LCDHandle](#).



**void BSP\_LCD\_GLASS\_DeInit ( void )**

Deinitialize the LCD GLASS relative GPIO port IOs and LCD peripheral.

**Return values:**

**None**

Definition at line **205** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **LCD\_MspDeInit()**, and **LCDHandle**.

**void BSP\_LCD\_GLASS\_DisplayBar ( uint32\_t BarId )**

Display one or several bar in LCD frame buffer.

**Parameters:**

**BarId,:** specifies the LCD GLASS Bar to display This parameter can be one of the following values:

- BAR0: LCD GLASS Bar 0
- BAR0: LCD GLASS Bar 1
- BAR0: LCD GLASS Bar 2
- BAR0: LCD GLASS Bar 3

**Return values:**

**None**

Definition at line **268** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **LCD\_BAR0\_2\_COM**, **LCD\_BAR0\_SEG**, **LCD\_BAR1\_3\_COM**, **LCD\_BAR1\_SEG**, **LCD\_BAR2\_SEG**, **LCD\_BAR3\_SEG**, **LCD\_BAR\_0**, **LCD\_BAR\_1**, **LCD\_BAR\_2**, **LCD\_BAR\_3**, and **LCDHandle**.

**void BSP\_LCD\_GLASS\_DisplayChar ( uint8\_t \***

**ch,**

**Point\_Typedef**      **Point**  
**DoublePoint\_Typedef**      **Color**  
**DigitPosition\_Typedef**      **Posit**

)

Write a character in the LCD RAM buffer.

**Parameters:**

- ch,:** The character to display.
- Point,:** A point to add in front of char. This parameter can be one of the following values:
- POINT\_OFF: No point to add in front of char.
  - POINT\_ON: Add a point in front of char.
- Colon,:** Flag indicating if a colon character has to be added in front of displayed character. This parameter can be one of the following values:
- DOUBLEPOINT\_OFF: No colon to add in back of char.
  - DOUBLEPOINT\_ON: Add an colon in back of char.
- Position,:** Position in the LCD of the character to write. This parameter can be any value in range [1:6].

**Return values:**

**None**

**Note:**

Required preconditions: The LCD should be cleared before to start the write operation.

Definition at line **452** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **LCDHandle**, and **WriteChar()**.

**void BSP\_LCD\_GLASS\_DisplayStrDeci ( uint16\_t \* ptr )**

Write a character string with decimal point in the LCD RAM buffer.

**Parameters:**

**ptr,:** Pointer to string to display on the LCD Glass.

**Return values:**

**None**

**Note:**

Required preconditions: Char is ASCII value "ORed" with decimal point or Colon flag

Definition at line **491** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **DOT**, **DOUBLE\_DOT**, **DOUBLEPOINT\_OFF**, **DOUBLEPOINT\_ON**, **LCD\_DIGIT\_POSITION\_1**, **LCD\_DIGIT\_POSITION\_6**, **LCDHandle**, **POINT\_OFF**, **POINT\_ON**, and **WriteChar()**.

**void BSP\_LCD\_GLASS\_DisplayString ( uint8\_t \* ptr )**

Write a character string in the LCD RAM buffer.

**Parameters:**

**ptr,:** Pointer to string to display on the LCD Glass.

**Return values:**

**None**

Definition at line **465** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **DOUBLEPOINT\_OFF**, **LCD\_DIGIT\_POSITION\_1**, **LCD\_DIGIT\_POSITION\_6**, **LCDHandle**, **POINT\_OFF**, and **WriteChar()**.

Referenced by **BSP\_LCD\_GLASS\_ScrollSentence()**.

**void** **BSP\_LCD\_GLASS\_Init** ( **void** )

Initialize the LCD GLASS relative GPIO port IOs and LCD peripheral.

**Return values:**

**None**

Definition at line **174** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **BSP\_LCD\_GLASS\_Clear()**, **LCD\_MspInit()**, and **LCDHandle**.

**void** **BSP\_LCD\_GLASS\_ScrollSentence** ( **uint8\_t** \* **ptr**,  
                                          **uint16\_t** **nScroll**,  
                                          **uint16\_t** **ScrollSpeed**  
                                          )

Display a string in scrolling mode.

**Parameters:**

**ptr,:**           Pointer to string to display on the LCD Glass.  
**nScroll,:**       Specifies how many time the message will be  
                  scrolled  
**ScrollSpeed** : Specifies the speed of the scroll, low value  
                  gives higher speed

**Return values:**

**None**

**Note:**

Required preconditions: The LCD should be cleared before to start the write operation.

Definition at line **543** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **bLCDGlass\_KeyPressed**, **BSP\_LCD\_GLASS\_Clear()**,  
and **BSP\_LCD\_GLASS\_DisplayString()**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Exported Functions

[STM32L476G-DISCOVERY QSPI](#)

## Functions

uint8_t	<b>BSP_QSPI_Init</b> (void)	Initializes the QSPI interface.
uint8_t	<b>BSP_QSPI_DeInit</b> (void)	De-Initializes the QSPI interface.
uint8_t	<b>BSP_QSPI_Read</b> (uint8_t *pData, uint32_t ReadAddr, uint32_t Size)	Reads an amount of data from the QSPI memory.
uint8_t	<b>BSP_QSPI_Write</b> (uint8_t *pData, uint32_t WriteAddr, uint32_t Size)	Writes an amount of data to the QSPI memory.
uint8_t	<b>BSP_QSPI_Erase_Block</b> (uint32_t BlockAddress)	Erases the specified block of the QSPI memory.
uint8_t	<b>BSP_QSPI_Erase_Sector</b> (uint32_t Sector)	Erases the specified sector of the QSPI memory.
uint8_t	<b>BSP_QSPI_Erase_Chip</b> (void)	Erases the entire QSPI memory.
uint8_t	<b>BSP_QSPI_GetStatus</b> (void)	Reads current status of the QSPI memory.
uint8_t	<b>BSP_QSPI_GetInfo</b> (QSPI_Info *pInfo)	Return the configuration of the QSPI memory.
uint8_t	<b>BSP_QSPI_EnableMemoryMappedMode</b> (void)	Configure the QSPI in memory-mapped mode.
uint8_t	<b>BSP_QSPI_SuspendErase</b> (void)	This function suspends an ongoing erase command.
uint8_t	<b>BSP_QSPI_ResumeErase</b> (void)	This function resumes a paused erase command.

## Function Documentation

**uint8\_t BSP\_QSPI\_DeInit ( void )**

De-Initializes the QSPI interface.

**Return values:**

**QSPI** memory status

Definition at line **165** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_ERROR**, **QSPI\_MspDeInit()**, **QSPI\_OK**, and **QSPIHandle**.

**uint8\_t BSP\_QSPI\_EnableMemoryMappedMode ( void )**

Configure the QSPI in memory-mapped mode.

**Return values:**

**QSPI** memory status

Definition at line **506** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_ERROR**, **QSPI\_OK**, and **QSPIHandle**.

**uint8\_t BSP\_QSPI\_Erase\_Block ( uint32\_t BlockAddress )**

Erases the specified block of the QSPI memory.

**Parameters:**

**BlockAddress,:** Block address to erase

**Return values:**

**QSPI** memory status



Definition at line **308** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_AutoPollingMemReady()**, **QSPI\_ERROR**, **QSPI\_OK**, **QSPI\_WriteEnable()**, and **QSPIHandle**.

**uint8\_t BSP\_QSPI\_Erase\_Chip ( void )**

Erases the entire QSPI memory.

**Return values:**

**QSPI** memory status

Definition at line **397** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_AutoPollingMemReady()**, **QSPI\_ERROR**, **QSPI\_OK**, **QSPI\_WriteEnable()**, and **QSPIHandle**.

**uint8\_t BSP\_QSPI\_Erase\_Sector ( uint32\_t Sector )**

Erases the specified sector of the QSPI memory.

**Parameters:**

**Sector,:** Sector address to erase (0 to 255)

**Return values:**

**QSPI** memory status

**Note:**

This function is non blocking meaning that sector erase operation is started but not completed when the function returns. Application has to call **BSP\_QSPI\_GetStatus()** to know when the device is available again (i.e. erase operation completed).

Definition at line **356** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_ERROR**, **QSPI\_OK**, **QSPI\_WriteEnable()**, and

**QSPiHandle.**

**uint8\_t BSP\_QSPI\_GetInfo ( QSPI\_Info \* pInfo )**

Return the configuration of the QSPI memory.

**Parameters:**

**pInfo,:** pointer on the configuration structure

**Return values:**

**QSPI** memory status

Definition at line **490** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_Info::EraseSectorSize**,  
**QSPI\_Info::EraseSectorsNumber**, **QSPI\_Info::FlashSize**,  
**QSPI\_Info::ProgPageSize**, **QSPI\_Info::ProgPagesNumber**, and  
**QSPI\_OK**.

**uint8\_t BSP\_QSPI\_GetStatus ( void )**

Reads current status of the QSPI memory.

**Return values:**

**QSPI** memory status

Definition at line **437** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_BUSY**, **QSPI\_ERROR**, **QSPI\_OK**,  
**QSPI\_SUSPENDED**, and **QSPiHandle**.

Referenced by **BSP\_QSPI\_ResumeErase()**, and  
**BSP\_QSPI\_SuspendErase()**.

**uint8\_t BSP\_QSPI\_Init ( void )**

Initializes the QSPI interface.

**Return values:**

**QSPI** memory status

Definition at line **120** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_DummyCyclesCfg()**, **QSPI\_ERROR**, **QSPI\_MsplInit()**, **QSPI\_NOT\_SUPPORTED**, **QSPI\_OK**, **QSPI\_ResetMemory()**, and **QSPISHandle**.

```
uint8_t BSP_QSPI_Read ( uint8_t * pData,  
                        uint32_t ReadAddr,  
                        uint32_t Size  
                        )
```

Reads an amount of data from the QSPI memory.

**Parameters:**

**pData,:** Pointer to data to be read

**ReadAddr,:** Read start address

**Size,:** Size of data to read

**Return values:**

**QSPI** memory status

Definition at line **188** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_ERROR**, **QSPI\_OK**, and **QSPISHandle**.

```
uint8_t BSP_QSPI_ResumeErase ( void )
```

This function resumes a paused erase command.

**Return values:**

**QSPI** memory status

Definition at line **586** of file **stm32l476g\_discovery\_qspi.c**.

References **BSP\_QSPI\_GetStatus()**, **QSPI\_BUSY**, **QSPI\_ERROR**, **QSPI\_OK**, **QSPI\_SUSPENDED**, **QSPI\_WriteEnable()**, and **QSPISHandle**.

**uint8\_t BSP\_QSPI\_SuspendErase ( void )**

This function suspends an ongoing erase command.

**Return values:**

**QSPI** memory status

Definition at line **539** of file **stm32l476g\_discovery\_qspi.c**.

References **BSP\_QSPI\_GetStatus()**, **QSPI\_BUSY**, **QSPI\_ERROR**, **QSPI\_OK**, **QSPI\_SUSPENDED**, **QSPI\_WriteEnable()**, and **QSPISHandle**.

**uint8\_t BSP\_QSPI\_Write ( uint8\_t \* pData,  
uint32\_t WriteAddr,  
uint32\_t Size  
)**

Writes an amount of data to the QSPI memory.

**Parameters:**

**pData,:** Pointer to data to be written

**WriteAddr,:** Write start address

**Size,:** Size of data to write

**Return values:**

**QSPI** memory status

Definition at line **228** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_AutoPollingMemReady()**, **QSPI\_ERROR**, **QSPI\_OK**, **QSPI\_WriteEnable()**, and **QSPISHandle**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## Exported Constants

[STM32L476G-DISCOVERY GLASS LCD](#)

## Defines

#define	<b>COM_PER_DIGIT_NB</b>	4	LCD digit defintion.
#define	<b>SEG_PER_DIGIT_NB</b>	4	
#define	<b>LCD_MAP_CHAR_COM0_SEG_1ST_POS</b>	(1 << LCD_MAP_CHAR_COM0_SEG_1ST_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM0_SEG_2ND_POS</b>	(1 << LCD_MAP_CHAR_COM0_SEG_2ND_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM0_SEG_3RD_POS</b>	(1 << LCD_MAP_CHAR_COM0_SEG_3RD_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM0_SEG_4TH_POS</b>	(1 << LCD_MAP_CHAR_COM0_SEG_4TH_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM1_SEG_1ST_POS</b>	(1 << LCD_MAP_CHAR_COM1_SEG_1ST_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM1_SEG_2ND_POS</b>	(1 << LCD_MAP_CHAR_COM1_SEG_2ND_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM1_SEG_3RD_POS</b>	(1 << LCD_MAP_CHAR_COM1_SEG_3RD_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM1_SEG_4TH_POS</b>	(1 << LCD_MAP_CHAR_COM1_SEG_4TH_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM2_SEG_1ST_POS</b>	(1 << LCD_MAP_CHAR_COM2_SEG_1ST_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM2_SEG_2ND_POS</b>	(1 << LCD_MAP_CHAR_COM2_SEG_2ND_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM2_SEG_3RD_POS</b>	(1 << LCD_MAP_CHAR_COM2_SEG_3RD_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM2_SEG_4TH_POS</b>	(1 << LCD_MAP_CHAR_COM2_SEG_4TH_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM3_SEG_1ST_POS</b>	(1 << LCD_MAP_CHAR_COM3_SEG_1ST_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM3_SEG_2ND_POS</b>	(1 << LCD_MAP_CHAR_COM3_SEG_2ND_SHIFT)	
#define	<b>LCD_MAP_CHAR_COM3_SEG_3RD_POS</b>	(1 <<	

	LCD_MAP_CHAR_COM3_SEG_3RD_SHIFT)	
#define	LCD_MAP_CHAR_COM3_SEG_4TH_POS (1 << LCD_MAP_CHAR_COM3_SEG_4TH_SHIFT)	
#define	LCD_MAP_CHAR_COM0_SEG_1ST_SHIFT	0x00000000
#define	LCD_MAP_CHAR_COM0_SEG_2ND_SHIFT	0x00000001
#define	LCD_MAP_CHAR_COM0_SEG_3RD_SHIFT	0x00000002
#define	LCD_MAP_CHAR_COM0_SEG_4TH_SHIFT	0x00000003
#define	LCD_MAP_CHAR_COM1_SEG_1ST_SHIFT	0x00000004
#define	LCD_MAP_CHAR_COM1_SEG_2ND_SHIFT	0x00000005
#define	LCD_MAP_CHAR_COM1_SEG_3RD_SHIFT	0x00000006
#define	LCD_MAP_CHAR_COM1_SEG_4TH_SHIFT	0x00000007
#define	LCD_MAP_CHAR_COM2_SEG_1ST_SHIFT	0x00000008
#define	LCD_MAP_CHAR_COM2_SEG_2ND_SHIFT	0x00000009
#define	LCD_MAP_CHAR_COM2_SEG_3RD_SHIFT	0x00000010
#define	LCD_MAP_CHAR_COM2_SEG_4TH_SHIFT	0x00000011
#define	LCD_MAP_CHAR_COM3_SEG_1ST_SHIFT	0x00000012
#define	LCD_MAP_CHAR_COM3_SEG_2ND_SHIFT	0x00000013
#define	LCD_MAP_CHAR_COM3_SEG_3RD_SHIFT	0x00000014
#define	LCD_MAP_CHAR_COM3_SEG_4TH_SHIFT	0x00000015
#define	LCD_DIGIT1_COM0 LCD_COM0 LCD Digit defines.	
#define	LCD_DIGIT1_COM0_SEG_MASK ~(LCD_SEG0   LCD_SEG1   LCD_SEG22   LCD_SEG23)	
#define	LCD_DIGIT1_COM1 LCD_COM1	
#define	LCD_DIGIT1_COM1_SEG_MASK ~(LCD_SEG0   LCD_SEG1   LCD_SEG22   LCD_SEG23)	
#define	LCD_DIGIT1_COM2 LCD_COM2	
#define	LCD_DIGIT1_COM2_SEG_MASK ~(LCD_SEG0   LCD_SEG1   LCD_SEG22   LCD_SEG23)	
#define	LCD_DIGIT1_COM3 LCD_COM3	
#define	LCD_DIGIT1_COM3_SEG_MASK ~(LCD_SEG0   LCD_SEG1   LCD_SEG22   LCD_SEG23)	
#define	LCD_DIGIT2_COM0 LCD_COM0	
#define	LCD_DIGIT2_COM0_SEG_MASK ~(LCD_SEG2   LCD_SEG3   LCD_SEG20   LCD_SEG21)	



```

#define LCD_DIGIT2_COM1 LCD_COM1

#define LCD_DIGIT2_COM1_SEG_MASK ~(LCD_SEG2 |
LCD_SEG3 | LCD_SEG20 | LCD_SEG21)

#define LCD_DIGIT2_COM2 LCD_COM2

#define LCD_DIGIT2_COM2_SEG_MASK ~(LCD_SEG2 |
LCD_SEG3 | LCD_SEG20 | LCD_SEG21)

#define LCD_DIGIT2_COM3 LCD_COM3

#define LCD_DIGIT2_COM3_SEG_MASK ~(LCD_SEG2 |
LCD_SEG3 | LCD_SEG20 | LCD_SEG21)

#define LCD_DIGIT3_COM0 LCD_COM0

#define LCD_DIGIT3_COM0_SEG_MASK ~(LCD_SEG4 |
LCD_SEG5 | LCD_SEG18 | LCD_SEG19)

#define LCD_DIGIT3_COM1 LCD_COM1

#define LCD_DIGIT3_COM1_SEG_MASK ~(LCD_SEG4 |
LCD_SEG5 | LCD_SEG18 | LCD_SEG19)

#define LCD_DIGIT3_COM2 LCD_COM2

#define LCD_DIGIT3_COM2_SEG_MASK ~(LCD_SEG4 |
LCD_SEG5 | LCD_SEG18 | LCD_SEG19)

#define LCD_DIGIT3_COM3 LCD_COM3

#define LCD_DIGIT3_COM3_SEG_MASK ~(LCD_SEG4 |
LCD_SEG5 | LCD_SEG18 | LCD_SEG19)

#define LCD_DIGIT4_COM0 LCD_COM0

#define LCD_DIGIT4_COM0_SEG_MASK ~(LCD_SEG6 |
LCD_SEG17)

#define LCD_DIGIT4_COM0_1 LCD_COM0_1

#define LCD_DIGIT4_COM0_1_SEG_MASK ~(LCD_SEG7 |
LCD_SEG16)

#define LCD_DIGIT4_COM1 LCD_COM1

#define LCD_DIGIT4_COM1_SEG_MASK ~(LCD_SEG6 |
LCD_SEG17)

#define LCD_DIGIT4_COM1_1 LCD_COM1_1

#define LCD_DIGIT4_COM1_1_SEG_MASK ~(LCD_SEG7 |
LCD_SEG16)

#define LCD_DIGIT4_COM2 LCD_COM2

```

```

#define LCD_DIGIT4_COM2_SEG_MASK ~(LCD_SEG6 |
LCD_SEG17)
#define LCD_DIGIT4_COM2_1 LCD_COM2_1
#define LCD_DIGIT4_COM2_1_SEG_MASK ~(LCD_SEG7 |
LCD_SEG16)
#define LCD_DIGIT4_COM3 LCD_COM3
#define LCD_DIGIT4_COM3_SEG_MASK ~(LCD_SEG6 |
LCD_SEG17)
#define LCD_DIGIT4_COM3_1 LCD_COM3_1
#define LCD_DIGIT4_COM3_1_SEG_MASK ~(LCD_SEG7 |
LCD_SEG16)
#define LCD_DIGIT5_COM0 LCD_COM0
#define LCD_DIGIT5_COM0_SEG_MASK ~(LCD_SEG9 |
LCD_SEG14)
#define LCD_DIGIT5_COM0_1 LCD_COM0_1
#define LCD_DIGIT5_COM0_1_SEG_MASK ~(LCD_SEG8 |
LCD_SEG15)
#define LCD_DIGIT5_COM1 LCD_COM1
#define LCD_DIGIT5_COM1_SEG_MASK ~(LCD_SEG9 |
LCD_SEG14)
#define LCD_DIGIT5_COM1_1 LCD_COM1_1
#define LCD_DIGIT5_COM1_1_SEG_MASK ~(LCD_SEG8 |
LCD_SEG15)
#define LCD_DIGIT5_COM2 LCD_COM2
#define LCD_DIGIT5_COM2_SEG_MASK ~(LCD_SEG9 |
LCD_SEG14)
#define LCD_DIGIT5_COM2_1 LCD_COM2_1
#define LCD_DIGIT5_COM2_1_SEG_MASK ~(LCD_SEG8 |
LCD_SEG15)
#define LCD_DIGIT5_COM3 LCD_COM3
#define LCD_DIGIT5_COM3_SEG_MASK ~(LCD_SEG9 |
LCD_SEG14)
#define LCD_DIGIT5_COM3_1 LCD_COM3_1
#define LCD_DIGIT5_COM3_1_SEG_MASK ~(LCD_SEG8 |
LCD_SEG15)

```

```

#define LCD_DIGIT6_COM0 LCD_COM0
#define LCD_DIGIT6_COM0_SEG_MASK ~(LCD_SEG10 |
LCD_SEG11 | LCD_SEG12 | LCD_SEG13)

#define LCD_DIGIT6_COM1 LCD_COM1
#define LCD_DIGIT6_COM1_SEG_MASK ~(LCD_SEG10 |
LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
#define LCD_DIGIT6_COM2 LCD_COM2
#define LCD_DIGIT6_COM2_SEG_MASK ~(LCD_SEG10 |
LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
#define LCD_DIGIT6_COM3 LCD_COM3
#define LCD_DIGIT6_COM3_SEG_MASK ~(LCD_SEG10 |
LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
#define LCD_BAR0_2_COM LCD_COM3
LCD Bar location.

#define LCD_BAR1_3_COM LCD_COM2
#define LCD_BAR0_SEG LCD_SEG11
#define LCD_BAR1_SEG LCD_SEG11
#define LCD_BAR2_SEG LCD_SEG9
#define LCD_BAR3_SEG LCD_SEG9
#define LCD_BAR0_2_SEG_MASK ~(LCD_BAR0_SEG |
LCD_BAR2_SEG)
#define LCD_BAR1_3_SEG_MASK ~(LCD_BAR1_SEG |
LCD_BAR3_SEG)
#define LCD_COM0 MCU_LCD_COM0
LCD segments & coms redefinition.

#define LCD_COM0_1 MCU_LCD_COM0_1
#define LCD_COM1 MCU_LCD_COM1
#define LCD_COM1_1 MCU_LCD_COM1_1
#define LCD_COM2 MCU_LCD_COM2
#define LCD_COM2_1 MCU_LCD_COM2_1
#define LCD_COM3 MCU_LCD_COM3
#define LCD_COM3_1 MCU_LCD_COM3_1
#define LCD_SEG0 MCU_LCD_SEG4
#define LCD_SEG1 MCU_LCD_SEG23

```

#define	LCD_SEG2	MCU_LCD_SEG6
#define	LCD_SEG3	MCU_LCD_SEG13
#define	LCD_SEG4	MCU_LCD_SEG15
#define	LCD_SEG5	MCU_LCD_SEG29
#define	LCD_SEG6	MCU_LCD_SEG31
#define	LCD_SEG7	MCU_LCD_SEG33
#define	LCD_SEG8	MCU_LCD_SEG35
#define	LCD_SEG9	MCU_LCD_SEG25
#define	LCD_SEG10	MCU_LCD_SEG17
#define	LCD_SEG11	MCU_LCD_SEG8
#define	LCD_SEG12	MCU_LCD_SEG9
#define	LCD_SEG13	MCU_LCD_SEG26
#define	LCD_SEG14	MCU_LCD_SEG24
#define	LCD_SEG15	MCU_LCD_SEG34
#define	LCD_SEG16	MCU_LCD_SEG32
#define	LCD_SEG17	MCU_LCD_SEG30
#define	LCD_SEG18	MCU_LCD_SEG28
#define	LCD_SEG19	MCU_LCD_SEG14
#define	LCD_SEG20	MCU_LCD_SEG12
#define	LCD_SEG21	MCU_LCD_SEG5
#define	LCD_SEG22	MCU_LCD_SEG22
#define	LCD_SEG23	MCU_LCD_SEG3
#define	LCD_SEG0_SHIFT	MCU_LCD_SEG4_SHIFT
#define	LCD_SEG1_SHIFT	MCU_LCD_SEG23_SHIFT
#define	LCD_SEG2_SHIFT	MCU_LCD_SEG6_SHIFT
#define	LCD_SEG3_SHIFT	MCU_LCD_SEG13_SHIFT
#define	LCD_SEG4_SHIFT	MCU_LCD_SEG15_SHIFT
#define	LCD_SEG5_SHIFT	MCU_LCD_SEG29_SHIFT
#define	LCD_SEG6_SHIFT	MCU_LCD_SEG31_SHIFT
#define	LCD_SEG7_SHIFT	MCU_LCD_SEG33_SHIFT
#define	LCD_SEG8_SHIFT	MCU_LCD_SEG35_SHIFT
#define	LCD_SEG9_SHIFT	MCU_LCD_SEG25_SHIFT
#define	LCD_SEG10_SHIFT	MCU_LCD_SEG17_SHIFT
#define	LCD_SEG11_SHIFT	MCU_LCD_SEG8_SHIFT

```

#define LCD_SEG12_SHIFT MCU_LCD_SEG9_SHIFT
#define LCD_SEG13_SHIFT MCU_LCD_SEG26_SHIFT
#define LCD_SEG14_SHIFT MCU_LCD_SEG24_SHIFT
#define LCD_SEG15_SHIFT MCU_LCD_SEG34_SHIFT
#define LCD_SEG16_SHIFT MCU_LCD_SEG32_SHIFT
#define LCD_SEG17_SHIFT MCU_LCD_SEG30_SHIFT
#define LCD_SEG18_SHIFT MCU_LCD_SEG28_SHIFT
#define LCD_SEG19_SHIFT MCU_LCD_SEG14_SHIFT
#define LCD_SEG20_SHIFT MCU_LCD_SEG12_SHIFT
#define LCD_SEG21_SHIFT MCU_LCD_SEG5_SHIFT
#define LCD_SEG22_SHIFT MCU_LCD_SEG22_SHIFT
#define LCD_SEG23_SHIFT MCU_LCD_SEG3_SHIFT
#define MCU_LCD_COM0 LCD_RAM_REGISTER0
STM32 LCD segments & coms definitions.
#define MCU_LCD_COM0_1 LCD_RAM_REGISTER1
#define MCU_LCD_COM1 LCD_RAM_REGISTER2
#define MCU_LCD_COM1_1 LCD_RAM_REGISTER3
#define MCU_LCD_COM2 LCD_RAM_REGISTER4
#define MCU_LCD_COM2_1 LCD_RAM_REGISTER5
#define MCU_LCD_COM3 LCD_RAM_REGISTER6
#define MCU_LCD_COM3_1 LCD_RAM_REGISTER7
#define MCU_LCD_COM4 LCD_RAM_REGISTER8
#define MCU_LCD_COM4_1 LCD_RAM_REGISTER9
#define MCU_LCD_COM5 LCD_RAM_REGISTER10
#define MCU_LCD_COM5_1 LCD_RAM_REGISTER11
#define MCU_LCD_COM6 LCD_RAM_REGISTER12
#define MCU_LCD_COM6_1 LCD_RAM_REGISTER13
#define MCU_LCD_COM7 LCD_RAM_REGISTER14
#define MCU_LCD_COM7_1 LCD_RAM_REGISTER15
#define MCU_LCD_SEG0 (1U << MCU_LCD_SEG0_SHIFT)
#define MCU_LCD_SEG1 (1U << MCU_LCD_SEG1_SHIFT)
#define MCU_LCD_SEG2 (1U << MCU_LCD_SEG2_SHIFT)
#define MCU_LCD_SEG3 (1U << MCU_LCD_SEG3_SHIFT)
#define MCU_LCD_SEG4 (1U << MCU_LCD_SEG4_SHIFT)

```

#define	<b>MCU_LCD_SEG5</b>	(1U << MCU_LCD_SEG5_SHIFT)
#define	<b>MCU_LCD_SEG6</b>	(1U << MCU_LCD_SEG6_SHIFT)
#define	<b>MCU_LCD_SEG7</b>	(1U << MCU_LCD_SEG7_SHIFT)
#define	<b>MCU_LCD_SEG8</b>	(1U << MCU_LCD_SEG8_SHIFT)
#define	<b>MCU_LCD_SEG9</b>	(1U << MCU_LCD_SEG9_SHIFT)
#define	<b>MCU_LCD_SEG10</b>	(1U << MCU_LCD_SEG10_SHIFT)
#define	<b>MCU_LCD_SEG11</b>	(1U << MCU_LCD_SEG11_SHIFT)
#define	<b>MCU_LCD_SEG12</b>	(1U << MCU_LCD_SEG12_SHIFT)
#define	<b>MCU_LCD_SEG13</b>	(1U << MCU_LCD_SEG13_SHIFT)
#define	<b>MCU_LCD_SEG14</b>	(1U << MCU_LCD_SEG14_SHIFT)
#define	<b>MCU_LCD_SEG15</b>	(1U << MCU_LCD_SEG15_SHIFT)
#define	<b>MCU_LCD_SEG16</b>	(1U << MCU_LCD_SEG16_SHIFT)
#define	<b>MCU_LCD_SEG17</b>	(1U << MCU_LCD_SEG17_SHIFT)
#define	<b>MCU_LCD_SEG18</b>	(1U << MCU_LCD_SEG18_SHIFT)
#define	<b>MCU_LCD_SEG19</b>	(1U << MCU_LCD_SEG19_SHIFT)
#define	<b>MCU_LCD_SEG20</b>	(1U << MCU_LCD_SEG20_SHIFT)
#define	<b>MCU_LCD_SEG21</b>	(1U << MCU_LCD_SEG21_SHIFT)
#define	<b>MCU_LCD_SEG22</b>	(1U << MCU_LCD_SEG22_SHIFT)
#define	<b>MCU_LCD_SEG23</b>	(1U << MCU_LCD_SEG23_SHIFT)
#define	<b>MCU_LCD_SEG24</b>	(1U << MCU_LCD_SEG24_SHIFT)
#define	<b>MCU_LCD_SEG25</b>	(1U << MCU_LCD_SEG25_SHIFT)
#define	<b>MCU_LCD_SEG26</b>	(1U << MCU_LCD_SEG26_SHIFT)
#define	<b>MCU_LCD_SEG27</b>	(1U << MCU_LCD_SEG27_SHIFT)
#define	<b>MCU_LCD_SEG28</b>	(1U << MCU_LCD_SEG28_SHIFT)
#define	<b>MCU_LCD_SEG29</b>	(1U << MCU_LCD_SEG29_SHIFT)
#define	<b>MCU_LCD_SEG30</b>	(1U << MCU_LCD_SEG30_SHIFT)
#define	<b>MCU_LCD_SEG31</b>	(1U << MCU_LCD_SEG31_SHIFT)
#define	<b>MCU_LCD_SEG32</b>	(1U << MCU_LCD_SEG32_SHIFT)
#define	<b>MCU_LCD_SEG33</b>	(1U << MCU_LCD_SEG33_SHIFT)
#define	<b>MCU_LCD_SEG34</b>	(1U << MCU_LCD_SEG34_SHIFT)
#define	<b>MCU_LCD_SEG35</b>	(1U << MCU_LCD_SEG35_SHIFT)
#define	<b>MCU_LCD_SEG36</b>	(1U << MCU_LCD_SEG36_SHIFT)
#define	<b>MCU_LCD_SEG37</b>	(1U << MCU_LCD_SEG37_SHIFT)

#define	MCU_LCD_SEG38	(1U << MCU_LCD_SEG38_SHIFT)
#define	MCU_LCD_SEG0_SHIFT	0
#define	MCU_LCD_SEG1_SHIFT	1
#define	MCU_LCD_SEG2_SHIFT	2
#define	MCU_LCD_SEG3_SHIFT	3
#define	MCU_LCD_SEG4_SHIFT	4
#define	MCU_LCD_SEG5_SHIFT	5
#define	MCU_LCD_SEG6_SHIFT	6
#define	MCU_LCD_SEG7_SHIFT	7
#define	MCU_LCD_SEG8_SHIFT	8
#define	MCU_LCD_SEG9_SHIFT	9
#define	MCU_LCD_SEG10_SHIFT	10
#define	MCU_LCD_SEG11_SHIFT	11
#define	MCU_LCD_SEG12_SHIFT	12
#define	MCU_LCD_SEG13_SHIFT	13
#define	MCU_LCD_SEG14_SHIFT	14
#define	MCU_LCD_SEG15_SHIFT	15
#define	MCU_LCD_SEG16_SHIFT	16
#define	MCU_LCD_SEG17_SHIFT	17
#define	MCU_LCD_SEG18_SHIFT	18
#define	MCU_LCD_SEG19_SHIFT	19
#define	MCU_LCD_SEG20_SHIFT	20
#define	MCU_LCD_SEG21_SHIFT	21
#define	MCU_LCD_SEG22_SHIFT	22
#define	MCU_LCD_SEG23_SHIFT	23
#define	MCU_LCD_SEG24_SHIFT	24
#define	MCU_LCD_SEG25_SHIFT	25
#define	MCU_LCD_SEG26_SHIFT	26
#define	MCU_LCD_SEG27_SHIFT	27
#define	MCU_LCD_SEG28_SHIFT	28
#define	MCU_LCD_SEG29_SHIFT	29
#define	MCU_LCD_SEG30_SHIFT	30
#define	MCU_LCD_SEG31_SHIFT	31
#define	MCU_LCD_SEG32_SHIFT	0



```

#define MCU_LCD_SEG33_SHIFT 1
#define MCU_LCD_SEG34_SHIFT 2
#define MCU_LCD_SEG35_SHIFT 3
#define MCU_LCD_SEG36_SHIFT 4
#define MCU_LCD_SEG37_SHIFT 5
#define MCU_LCD_SEG38_SHIFT 6
#define MCU_LCD_SEG39_SHIFT 7
#define MCU_LCD_SEG40_SHIFT 8
#define MCU_LCD_SEG41_SHIFT 9
#define MCU_LCD_SEG42_SHIFT 10
#define MCU_LCD_SEG43_SHIFT 11
#define LCD_GPIO_BANKA_PINS
    LCD Pins definition.
#define LCD_GPIO_BANKB_PINS
#define LCD_GPIO_BANKC_PINS
#define LCD_GPIO_BANKD_PINS
#define SCROLL_SPEED_HIGH 150
#define SCROLL_SPEED_MEDIUM 300
#define SCROLL_SPEED_LOW 450
#define DOT ((uint16_t) 0x8000 ) /* for add decimal point in string */
#define DOUBLE_DOT ((uint16_t) 0x4000) /* for add decimal point
    in string */
#define C_OPENPARMAP ((uint16_t) 0x0028)
#define C_CLOSEPARMAP ((uint16_t) 0x0011)
#define C_DMAP ((uint16_t) 0xf300)
#define C_MMAP ((uint16_t) 0xb210)
#define C_NMAP ((uint16_t) 0x2210)
#define C_UMAP ((uint16_t) 0x6084)
#define C_STAR ((uint16_t) 0xA0DD)
#define C_MINUS ((uint16_t) 0xA000)
#define C_PLUS ((uint16_t) 0xA014)
#define C_SLATCH ((uint16_t) 0x00c0)
#define C_PERCENT_1 ((uint16_t) 0xec00)
#define C_PERCENT_2 ((uint16_t) 0xb300)

```



```
#define C_FULL ((uint16_t) 0xffdd)
```

---

## Define Documentation

**#define C\_CLOSEPARMAP** ((uint16\_t) 0x0011)

Definition at line **469** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_DMAP** ((uint16\_t) 0xf300)

Definition at line **472** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_FULL** ((uint16\_t) 0xffdd)

Definition at line **501** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_MINUS** ((uint16\_t) 0xA000)

Definition at line **487** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_MMAP** ((uint16\_t) 0xb210)

Definition at line **475** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_NMAP** ((uint16\_t) 0x2210)

Definition at line **478** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_OPENPARMAP** ((uint16\_t) 0x0028)

Definition at line **466** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_PERCENT\_1** ((uint16\_t) 0xec00)

Definition at line **496** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_PERCENT\_2** ((uint16\_t) 0xb300)

Definition at line **499** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_PLUS** ((uint16\_t) 0xA014)

Definition at line **490** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

**#define C\_SLATCH** ((uint16\_t) 0x00c0)

Definition at line **493** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

```
#define C_STAR ((uint16_t) 0xA0DD)
```

Definition at line **484** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

```
#define C_UMAP ((uint16_t) 0x6084)
```

Definition at line **481** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **Convert()**.

```
#define COM_PER_DIGIT_NB 4
```

LCD digit defintion.

Specifies number of COM to address a digit

Definition at line **134** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define DOT ((uint16_t) 0x8000 ) /* for add decimal point in string
```

Definition at line **462** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **BSP\_LCD\_GLASS\_DisplayStrDeci()**.

```
#define DOUBLE_DOT ((uint16_t) 0x4000) /* for add decimal point
```

Definition at line **463** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **BSP\_LCD\_GLASS\_DisplayStrDeci()**.

**#define LCD\_BAR0\_2\_COM LCD\_COM3**

LCD Bar location.

Definition at line **246** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **BSP\_LCD\_GLASS\_BarLevelConfig()**,  
**BSP\_LCD\_GLASS\_ClearBar()**, and  
**BSP\_LCD\_GLASS\_DisplayBar()**.

**#define LCD\_BAR0\_2\_SEG\_MASK ~(LCD\_BAR0\_SEG | LCD\_BAR**

Definition at line **252** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_BAR0\_SEG LCD\_SEG11**

Definition at line **248** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **BSP\_LCD\_GLASS\_BarLevelConfig()**,  
**BSP\_LCD\_GLASS\_ClearBar()**, and  
**BSP\_LCD\_GLASS\_DisplayBar()**.

**#define LCD\_BAR1\_3\_COM LCD\_COM2**

Definition at line **247** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **BSP\_LCD\_GLASS\_BarLevelConfig()**,  
**BSP\_LCD\_GLASS\_ClearBar()**, and  
**BSP\_LCD\_GLASS\_DisplayBar()**.

```
#define LCD_BAR1_3_SEG_MASK ~(LCD_BAR1_SEG | LCD_BAR
```

Definition at line **253** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_BAR1_SEG LCD_SEG11
```

Definition at line **249** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **BSP\_LCD\_GLASS\_BarLevelConfig()**,  
**BSP\_LCD\_GLASS\_ClearBar()**, and  
**BSP\_LCD\_GLASS\_DisplayBar()**.

```
#define LCD_BAR2_SEG LCD_SEG9
```

Definition at line **250** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **BSP\_LCD\_GLASS\_BarLevelConfig()**,  
**BSP\_LCD\_GLASS\_ClearBar()**, and  
**BSP\_LCD\_GLASS\_DisplayBar()**.

```
#define LCD_BAR3_SEG LCD_SEG9
```

Definition at line **251** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **BSP\_LCD\_GLASS\_BarLevelConfig()**,  
**BSP\_LCD\_GLASS\_ClearBar()**, and  
**BSP\_LCD\_GLASS\_DisplayBar()**.

```
#define LCD_COM0 MCU_LCD_COM0
```

LCD segments & coms redefinition.

LCD component segments & coms are not necessarily link to MCU segments & coms output.

Definition at line **260** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_COM0_1 MCU_LCD_COM0_1
```

Definition at line **261** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_COM1 MCU_LCD_COM1
```

Definition at line **262** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_COM1_1 MCU_LCD_COM1_1
```

Definition at line **263** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_COM2 MCU_LCD_COM2
```

Definition at line **264** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_COM2_1 MCU_LCD_COM2_1
```

Definition at line **265** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_COM3 MCU_LCD_COM3
```

Definition at line **266** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_COM3_1 MCU_LCD_COM3_1
```

Definition at line **267** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_DIGIT1\_COM0 LCD\_COM0**

LCD Digit defines.

Definition at line **173** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT1\_COM0\_SEG\_MASK ~(LCD\_SEG0 | LCD\_SEG**

Definition at line **174** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT1\_COM1 LCD\_COM1**

Definition at line **175** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT1\_COM1\_SEG\_MASK ~(LCD\_SEG0 | LCD\_SEG**

Definition at line **176** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT1\_COM2 LCD\_COM2**

Definition at line **177** of file **stm32l476g\_discovery\_glass\_lcd.h**.



Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT1\_COM2\_SEG\_MASK** ~(LCD\_SEG0 | LCD\_SEG

Definition at line **178** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT1\_COM3** LCD\_COM3

Definition at line **179** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT1\_COM3\_SEG\_MASK** ~(LCD\_SEG0 | LCD\_SEG

Definition at line **180** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT2\_COM0** LCD\_COM0

Definition at line **182** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT2\_COM0\_SEG\_MASK** ~(LCD\_SEG2 | LCD\_SEG

Definition at line **183** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT2\_COM1 LCD\_COM1**

Definition at line **184** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT2\_COM1\_SEG\_MASK ~(LCD\_SEG2 | LCD\_SEG**

Definition at line **185** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT2\_COM2 LCD\_COM2**

Definition at line **186** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT2\_COM2\_SEG\_MASK ~(LCD\_SEG2 | LCD\_SEG**

Definition at line **187** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT2\_COM3 LCD\_COM3**

Definition at line **188** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT2\_COM3\_SEG\_MASK ~(LCD\_SEG2 | LCD\_SEG**

Definition at line **189** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT3\_COM0 LCD\_COM0**

Definition at line **191** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT3\_COM0\_SEG\_MASK ~(LCD\_SEG4 | LCD\_SEG**

Definition at line **192** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT3\_COM1 LCD\_COM1**

Definition at line **193** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT3\_COM1\_SEG\_MASK ~(LCD\_SEG4 | LCD\_SEG**

Definition at line **194** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT3\_COM2 LCD\_COM2**

Definition at line **195** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT3\_COM2\_SEG\_MASK** ~(LCD\_SEG4 | LCD\_SEG

Definition at line **196** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT3\_COM3** LCD\_COM3

Definition at line **197** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT3\_COM3\_SEG\_MASK** ~(LCD\_SEG4 | LCD\_SEG

Definition at line **198** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT4\_COM0** LCD\_COM0

Definition at line **200** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT4\_COM0\_1** LCD\_COM0\_1

Definition at line **202** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT4\_COM0\_1\_SEG\_MASK** ~(LCD\_SEG7 | LCD\_SEG6)

Definition at line 203 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT4\_COM0\_SEG\_MASK** ~(LCD\_SEG6 | LCD\_SEG5)

Definition at line 201 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT4\_COM1** LCD\_COM1

Definition at line 204 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT4\_COM1\_1** LCD\_COM1\_1

Definition at line 206 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT4\_COM1\_1\_SEG\_MASK** ~(LCD\_SEG7 | LCD\_SEG6)

Definition at line 207 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT4\_COM1\_SEG\_MASK** ~(LCD\_SEG6 | LCD\_SEG5)

Definition at line **205** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT4\_COM2 LCD\_COM2**

Definition at line **208** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT4\_COM2\_1 LCD\_COM2\_1**

Definition at line **210** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT4\_COM2\_1\_SEG\_MASK ~(LCD\_SEG7 | LCD\_SEG6)**

Definition at line **211** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT4\_COM2\_SEG\_MASK ~(LCD\_SEG6 | LCD\_SEG5)**

Definition at line **209** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT4\_COM3 LCD\_COM3**

Definition at line **212** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT4\_COM3\_1 LCD\_COM3\_1**

Definition at line **214** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT4\_COM3\_1\_SEG\_MASK ~(LCD\_SEG7 | LCD\_SI**

Definition at line **215** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT4\_COM3\_SEG\_MASK ~(LCD\_SEG6 | LCD\_SEG**

Definition at line **213** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT5\_COM0 LCD\_COM0**

Definition at line **217** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT5\_COM0\_1 LCD\_COM0\_1**

Definition at line **219** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT5\_COM0\_1\_SEG\_MASK** ~(LCD\_SEG8 | LCD\_SEG9)

Definition at line 220 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT5\_COM0\_SEG\_MASK** ~(LCD\_SEG9 | LCD\_SEG10)

Definition at line 218 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT5\_COM1** LCD\_COM1

Definition at line 221 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT5\_COM1\_1** LCD\_COM1\_1

Definition at line 223 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT5\_COM1\_1\_SEG\_MASK** ~(LCD\_SEG8 | LCD\_SEG9)

Definition at line 224 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT5\_COM1\_SEG\_MASK** ~(LCD\_SEG9 | LCD\_SEG10)



Definition at line **222** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT5\_COM2 LCD\_COM2**

Definition at line **225** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT5\_COM2\_1 LCD\_COM2\_1**

Definition at line **227** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT5\_COM2\_1\_SEG\_MASK ~(LCD\_SEG8 | LCD\_SEG9)**

Definition at line **228** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT5\_COM2\_SEG\_MASK ~(LCD\_SEG9 | LCD\_SEG10)**

Definition at line **226** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_DIGIT5\_COM3 LCD\_COM3**

Definition at line **229** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT5\_COM3\_1 LCD\_COM3\_1**

Definition at line **231** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT5\_COM3\_1\_SEG\_MASK ~(LCD\_SEG8 | LCD\_SEG9)**

Definition at line **232** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT5\_COM3\_SEG\_MASK ~(LCD\_SEG9 | LCD\_SEG10)**

Definition at line **230** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT6\_COM0 LCD\_COM0**

Definition at line **234** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT6\_COM0\_SEG\_MASK ~(LCD\_SEG10 | LCD\_SEG11)**

Definition at line **235** of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

Referenced by [WriteChar\(\)](#).

**#define LCD\_DIGIT6\_COM1 LCD\_COM1**

Definition at line 236 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT6\_COM1\_SEG\_MASK ~(LCD\_SEG10 | LCD\_SE**

Definition at line 237 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT6\_COM2 LCD\_COM2**

Definition at line 238 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT6\_COM2\_SEG\_MASK ~(LCD\_SEG10 | LCD\_SE**

Definition at line 239 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT6\_COM3 LCD\_COM3**

Definition at line 240 of file `stm32l476g_discovery_glass_lcd.h`.

Referenced by `WriteChar()`.

**#define LCD\_DIGIT6\_COM3\_SEG\_MASK ~(LCD\_SEG10 | LCD\_SE**

Definition at line **241** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_GPIO\_BANKA\_PINS**

**Value:**

```
(GPIO_PIN_6 | GPIO_PIN_7 | GPIO_PIN_8 | \
                                GPIO_PIN_9 | GPIO_P
IN_10 | GPIO_PIN_15)
```

LCD Pins definition.

Definition at line **434** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **LCD\_MspDeInit()**, and **LCD\_MspInit()**.

**#define LCD\_GPIO\_BANKB\_PINS**

**Value:**

```
(GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_4 | \
                                GPIO_PIN_5 | GPIO_P
IN_9 | GPIO_PIN_12 | \
                                GPIO_PIN_13 | GPIO_
PIN_14 | GPIO_PIN_15)
```

Definition at line **436** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **LCD\_MspDeInit()**, and **LCD\_MspInit()**.

**#define LCD\_GPIO\_BANKC\_PINS**

**Value:**

```
(GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 | \
                                GPIO_PIN_6 | GPIO_P
```

```
IN_7 | GPIO_PIN_8)
```

Definition at line **439** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **LCD\_MspDeInit()**, and **LCD\_MspInit()**.

**#define LCD\_GPIO\_BANKD\_PINS**

**Value:**

```
(GPIO_PIN_8 | GPIO_PIN_9 | GPIO_PIN_10 | \
                                GPIO_PIN_11 | GPIO_
PIN_12 | GPIO_PIN_13 | \
                                GPIO_PIN_14 | GPIO_
PIN_15)
```

Definition at line **441** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **LCD\_MspDeInit()**, and **LCD\_MspInit()**.

**#define LCD\_MAP\_CHAR\_COM0\_SEG\_1ST\_POS (1 << LCD\_MAP**

Definition at line **137** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM0\_SEG\_1ST\_SHIFT 0x00000000**

Definition at line **153** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM0\_SEG\_2ND\_POS (1 << LCD\_MAP**

Definition at line **138** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM0\_SEG\_2ND\_SHIFT 0x00000001**

Definition at line **154** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM0\_SEG\_3RD\_POS (1 << LCD\_MAP**

Definition at line **139** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM0\_SEG\_3RD\_SHIFT 0x00000002**

Definition at line **155** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM0\_SEG\_4TH\_POS (1 << LCD\_MAP**

Definition at line **140** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM0\_SEG\_4TH\_SHIFT 0x00000003**

Definition at line **156** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM1\_SEG\_1ST\_POS (1 << LCD\_MAP**

Definition at line **141** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM1\_SEG\_1ST\_SHIFT 0x00000004**

Definition at line **157** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM1\_SEG\_2ND\_POS (1 << LCD\_MAP**

Definition at line **142** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM1\_SEG\_2ND\_SHIFT 0x00000005**

Definition at line **158** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM1\_SEG\_3RD\_POS (1 << LCD\_MAP**

Definition at line **143** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM1\_SEG\_3RD\_SHIFT 0x00000006**

Definition at line **159** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM1\_SEG\_4TH\_POS (1 << LCD\_MAP**

Definition at line **144** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM1\_SEG\_4TH\_SHIFT 0x00000007**

Definition at line **160** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM2\_SEG\_1ST\_POS (1 << LCD\_MAP**

Definition at line **145** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM2\_SEG\_1ST\_SHIFT 0x00000008**

Definition at line **161** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM2\_SEG\_2ND\_POS (1 << LCD\_MAP**

Definition at line **146** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM2\_SEG\_2ND\_SHIFT 0x00000009**

Definition at line **162** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM2\_SEG\_3RD\_POS (1 << LCD\_MAP**

Definition at line **147** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM2\_SEG\_3RD\_SHIFT 0x00000010**

Definition at line **163** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM2\_SEG\_4TH\_POS (1 << LCD\_MAP**

Definition at line **148** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM2\_SEG\_4TH\_SHIFT 0x00000011**

Definition at line **164** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM3\_SEG\_1ST\_POS (1 << LCD\_MAP**

Definition at line **149** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM3\_SEG\_1ST\_SHIFT 0x00000012**



Definition at line **165** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM3\_SEG\_2ND\_POS (1 << LCD\_MAP**

Definition at line **150** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM3\_SEG\_2ND\_SHIFT 0x00000013**

Definition at line **166** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM3\_SEG\_3RD\_POS (1 << LCD\_MAP**

Definition at line **151** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM3\_SEG\_3RD\_SHIFT 0x00000014**

Definition at line **167** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM3\_SEG\_4TH\_POS (1 << LCD\_MAP**

Definition at line **152** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_MAP\_CHAR\_COM3\_SEG\_4TH\_SHIFT 0x00000015**

Definition at line **168** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG0 MCU\_LCD\_SEG4**

Definition at line **278** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_SEG0_SHIFT MCU_LCD_SEG4_SHIFT
```

Definition at line **302** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

```
#define LCD_SEG1 MCU_LCD_SEG23
```

Definition at line **279** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_SEG10 MCU_LCD_SEG17
```

Definition at line **288** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_SEG10_SHIFT MCU_LCD_SEG17_SHIFT
```

Definition at line **312** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

```
#define LCD_SEG11 MCU_LCD_SEG8
```

Definition at line **289** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define LCD_SEG11_SHIFT MCU_LCD_SEG8_SHIFT
```

Definition at line **313** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG12 MCU\_LCD\_SEG9**

Definition at line **290** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG12\_SHIFT MCU\_LCD\_SEG9\_SHIFT**

Definition at line **314** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG13 MCU\_LCD\_SEG26**

Definition at line **291** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG13\_SHIFT MCU\_LCD\_SEG26\_SHIFT**

Definition at line **315** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG14 MCU\_LCD\_SEG24**

Definition at line **292** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG14\_SHIFT MCU\_LCD\_SEG24\_SHIFT**

Definition at line **316** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG15 MCU\_LCD\_SEG34**

Definition at line **293** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG15\_SHIFT MCU\_LCD\_SEG34\_SHIFT**

Definition at line **317** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG16 MCU\_LCD\_SEG32**

Definition at line **294** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG16\_SHIFT MCU\_LCD\_SEG32\_SHIFT**

Definition at line **318** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG17 MCU\_LCD\_SEG30**

Definition at line **295** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG17\_SHIFT MCU\_LCD\_SEG30\_SHIFT**

Definition at line **319** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG18 MCU\_LCD\_SEG28**

Definition at line **296** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG18\_SHIFT MCU\_LCD\_SEG28\_SHIFT**

Definition at line **320** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG19 MCU\_LCD\_SEG14**

Definition at line **297** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG19\_SHIFT MCU\_LCD\_SEG14\_SHIFT**

Definition at line **321** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG1\_SHIFT MCU\_LCD\_SEG23\_SHIFT**

Definition at line **303** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG2 MCU\_LCD\_SEG6**

Definition at line **280** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG20 MCU\_LCD\_SEG12**

Definition at line **298** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG20\_SHIFT MCU\_LCD\_SEG12\_SHIFT**

Definition at line **322** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG21 MCU\_LCD\_SEG5**

Definition at line **299** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG21\_SHIFT MCU\_LCD\_SEG5\_SHIFT**

Definition at line **323** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG22 MCU\_LCD\_SEG22**

Definition at line **300** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG22\_SHIFT MCU\_LCD\_SEG22\_SHIFT**

Definition at line **324** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG23 MCU\_LCD\_SEG3**

Definition at line **301** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG23\_SHIFT MCU\_LCD\_SEG3\_SHIFT**

Definition at line **325** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG2\_SHIFT MCU\_LCD\_SEG6\_SHIFT**

Definition at line **304** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG3 MCU\_LCD\_SEG13**

Definition at line **281** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG3\_SHIFT MCU\_LCD\_SEG13\_SHIFT**

Definition at line **305** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG4 MCU\_LCD\_SEG15**

Definition at line **282** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG4\_SHIFT MCU\_LCD\_SEG15\_SHIFT**

Definition at line **306** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG5 MCU\_LCD\_SEG29**

Definition at line **283** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG5\_SHIFT MCU\_LCD\_SEG29\_SHIFT**

Definition at line **307** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG6 MCU\_LCD\_SEG31**

Definition at line **284** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG6\_SHIFT MCU\_LCD\_SEG31\_SHIFT**

Definition at line **308** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG7 MCU\_LCD\_SEG33**

Definition at line **285** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG7\_SHIFT MCU\_LCD\_SEG33\_SHIFT**



Definition at line **309** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG8 MCU\_LCD\_SEG35**

Definition at line **286** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG8\_SHIFT MCU\_LCD\_SEG35\_SHIFT**

Definition at line **310** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define LCD\_SEG9 MCU\_LCD\_SEG25**

Definition at line **287** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define LCD\_SEG9\_SHIFT MCU\_LCD\_SEG25\_SHIFT**

Definition at line **311** of file **stm32l476g\_discovery\_glass\_lcd.h**.

Referenced by **WriteChar()**.

**#define MCU\_LCD\_COM0 LCD\_RAM\_REGISTER0**

STM32 LCD segments & coms definitions.

Definition at line **330** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_COM0\_1 LCD\_RAM\_REGISTER1**

Definition at line 331 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_COM1 LCD\_RAM\_REGISTER2**

Definition at line 332 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_COM1\_1 LCD\_RAM\_REGISTER3**

Definition at line 333 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_COM2 LCD\_RAM\_REGISTER4**

Definition at line 334 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_COM2\_1 LCD\_RAM\_REGISTER5**

Definition at line 335 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_COM3 LCD\_RAM\_REGISTER6**

Definition at line 336 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_COM3\_1 LCD\_RAM\_REGISTER7**

Definition at line 337 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_COM4 LCD\_RAM\_REGISTER8**

Definition at line **338** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_COM4\_1 LCD\_RAM\_REGISTER9**

Definition at line **339** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_COM5 LCD\_RAM\_REGISTER10**

Definition at line **340** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_COM5\_1 LCD\_RAM\_REGISTER11**

Definition at line **341** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_COM6 LCD\_RAM\_REGISTER12**

Definition at line **342** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_COM6\_1 LCD\_RAM\_REGISTER13**

Definition at line **343** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_COM7 LCD\_RAM\_REGISTER14**

Definition at line **344** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_COM7\_1 LCD\_RAM\_REGISTER15**

Definition at line **345** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG0 (1U << MCU\_LCD\_SEG0\_SHIFT)**

Definition at line **346** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG0\_SHIFT 0**

Definition at line **385** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG1 (1U << MCU\_LCD\_SEG1\_SHIFT)**

Definition at line **347** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG10 (1U << MCU\_LCD\_SEG10\_SHIFT)**

Definition at line **356** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG10\_SHIFT 10**

Definition at line **395** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG11 (1U << MCU\_LCD\_SEG11\_SHIFT)**

Definition at line **357** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG11\_SHIFT 11**

Definition at line **396** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG12 (1U << MCU\_LCD\_SEG12\_SHIFT)**

Definition at line 358 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG12\_SHIFT 12**

Definition at line 397 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG13 (1U << MCU\_LCD\_SEG13\_SHIFT)**

Definition at line 359 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG13\_SHIFT 13**

Definition at line 398 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG14 (1U << MCU\_LCD\_SEG14\_SHIFT)**

Definition at line 360 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG14\_SHIFT 14**

Definition at line 399 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG15 (1U << MCU\_LCD\_SEG15\_SHIFT)**

Definition at line 361 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG15\_SHIFT 15**

Definition at line **400** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG16 (1U << MCU_LCD_SEG16_SHIFT)
```

Definition at line **362** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG16_SHIFT 16
```

Definition at line **401** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG17 (1U << MCU_LCD_SEG17_SHIFT)
```

Definition at line **363** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG17_SHIFT 17
```

Definition at line **402** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG18 (1U << MCU_LCD_SEG18_SHIFT)
```

Definition at line **364** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG18_SHIFT 18
```

Definition at line **403** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG19 (1U << MCU_LCD_SEG19_SHIFT)
```

Definition at line **365** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG19\_SHIFT 19**

Definition at line 404 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG1\_SHIFT 1**

Definition at line 386 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG2 (1U << MCU\_LCD\_SEG2\_SHIFT)**

Definition at line 348 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG20 (1U << MCU\_LCD\_SEG20\_SHIFT)**

Definition at line 366 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG20\_SHIFT 20**

Definition at line 405 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG21 (1U << MCU\_LCD\_SEG21\_SHIFT)**

Definition at line 367 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG21\_SHIFT 21**

Definition at line 406 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG22 (1U << MCU\_LCD\_SEG22\_SHIFT)**

Definition at line 368 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG22\_SHIFT 22**

Definition at line 407 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG23 (1U << MCU\_LCD\_SEG23\_SHIFT)**

Definition at line 369 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG23\_SHIFT 23**

Definition at line 408 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG24 (1U << MCU\_LCD\_SEG24\_SHIFT)**

Definition at line 370 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG24\_SHIFT 24**

Definition at line 409 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG25 (1U << MCU\_LCD\_SEG25\_SHIFT)**

Definition at line 371 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG25\_SHIFT 25**



Definition at line **410** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG26 (1U << MCU\_LCD\_SEG26\_SHIFT)**

Definition at line **372** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG26\_SHIFT 26**

Definition at line **411** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG27 (1U << MCU\_LCD\_SEG27\_SHIFT)**

Definition at line **373** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG27\_SHIFT 27**

Definition at line **412** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG28 (1U << MCU\_LCD\_SEG28\_SHIFT)**

Definition at line **374** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG28\_SHIFT 28**

Definition at line **413** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG29 (1U << MCU\_LCD\_SEG29\_SHIFT)**

Definition at line **375** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG29\_SHIFT 29**

Definition at line 414 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG2\_SHIFT 2**

Definition at line 387 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG3 (1U << MCU\_LCD\_SEG3\_SHIFT)**

Definition at line 349 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG30 (1U << MCU\_LCD\_SEG30\_SHIFT)**

Definition at line 376 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG30\_SHIFT 30**

Definition at line 415 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG31 (1U << MCU\_LCD\_SEG31\_SHIFT)**

Definition at line 377 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG31\_SHIFT 31**

Definition at line 416 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG32 (1U << MCU\_LCD\_SEG32\_SHIFT)**

Definition at line **378** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG32\_SHIFT 0**

Definition at line **417** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG33 (1U << MCU\_LCD\_SEG33\_SHIFT)**

Definition at line **379** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG33\_SHIFT 1**

Definition at line **418** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG34 (1U << MCU\_LCD\_SEG34\_SHIFT)**

Definition at line **380** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG34\_SHIFT 2**

Definition at line **419** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG35 (1U << MCU\_LCD\_SEG35\_SHIFT)**

Definition at line **381** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG35\_SHIFT 3**

Definition at line **420** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG36 (1U << MCU\_LCD\_SEG36\_SHIFT)**

Definition at line **382** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG36\_SHIFT 4**

Definition at line **421** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG37 (1U << MCU\_LCD\_SEG37\_SHIFT)**

Definition at line **383** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG37\_SHIFT 5**

Definition at line **422** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG38 (1U << MCU\_LCD\_SEG38\_SHIFT)**

Definition at line **384** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG38\_SHIFT 6**

Definition at line **423** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG39\_SHIFT 7**

Definition at line **424** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG3\_SHIFT 3**

Definition at line 388 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG4 (1U << MCU\_LCD\_SEG4\_SHIFT)**

Definition at line 350 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG40\_SHIFT 8**

Definition at line 425 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG41\_SHIFT 9**

Definition at line 426 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG42\_SHIFT 10**

Definition at line 427 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG43\_SHIFT 11**

Definition at line 428 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG4\_SHIFT 4**

Definition at line 389 of file [stm32l476g\\_discovery\\_glass\\_lcd.h](#).

**#define MCU\_LCD\_SEG5 (1U << MCU\_LCD\_SEG5\_SHIFT)**

Definition at line **351** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG5\_SHIFT 5**

Definition at line **390** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG6 (1U << MCU\_LCD\_SEG6\_SHIFT)**

Definition at line **352** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG6\_SHIFT 6**

Definition at line **391** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG7 (1U << MCU\_LCD\_SEG7\_SHIFT)**

Definition at line **353** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG7\_SHIFT 7**

Definition at line **392** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG8 (1U << MCU\_LCD\_SEG8\_SHIFT)**

Definition at line **354** of file **stm32l476g\_discovery\_glass\_lcd.h**.

**#define MCU\_LCD\_SEG8\_SHIFT 8**

Definition at line **393** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG9 (1U << MCU_LCD_SEG9_SHIFT)
```

Definition at line **355** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define MCU_LCD_SEG9_SHIFT 9
```

Definition at line **394** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define SCROLL_SPEED_HIGH 150
```

Definition at line **458** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define SCROLL_SPEED_LOW 450
```

Definition at line **460** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define SCROLL_SPEED_MEDIUM 300
```

Definition at line **459** of file **stm32l476g\_discovery\_glass\_lcd.h**.

```
#define SEG_PER_DIGIT_NB 4
```

Specifies number of SEG to address a digit

Definition at line **135** of file **stm32l476g\_discovery\_glass\_lcd.h**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Enumerations](#)

## Exported Constants

[STM32L476G-DISCOVERY COMPASS](#)



## Enumerations

```
enum COMPASS_StatusTypeDef { COMPASS_OK = 0,  
                             COMPASS_ERROR = 1, COMPASS_TIMEOUT = 2 }
```

## Enumeration Type Documentation

enum **COMPASS\_StatusTypeDef**

**Enumerator:**

*COMPASS\_OK*

*COMPASS\_ERROR*

*COMPASS\_TIMEOUT*

Definition at line **73** of file **stm32l476g\_discovery\_compass.h**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Private Functions

[STM32L476G-DISCOVERY GLASS LCD](#)

## Functions

---

static void **Convert** (uint8\_t \*Char, **Point\_Typedef** Point, **DoublePoint\_Typedef** Colon)  
Convert an ascii char to the a LCD digit.

---

static void **WriteChar** (uint8\_t \*ch, **Point\_Typedef** Point, **DoublePoint\_Typedef** Colon, **DigitPosition\_Typedef** Position)  
Write a character in the LCD frame buffer.

---

static void **LCD\_MspInit** (LCD\_HandleTypeDef \*hlcd)  
Initialize the LCD MSP.

---

static void **LCD\_MspDeInit** (LCD\_HandleTypeDef \*hlcd)  
DeInitialize the LCD MSP.

---

## Function Documentation

```
static void Convert ( uint8_t *          Char,  
                    Point_TypeDef      Point,  
                    DoublePoint_TypeDef Colon  
                    )                  [static]
```

Convert an ascii char to the a LCD digit.

### Parameters:

**Char,:** a char to display.

**Point,:** a point to add in front of char This parameter can be:  
POINT\_OFF or POINT\_ON

**Colon** : flag indicating if a colon character has to be added in  
front of displayed character. This parameter can be:  
DOUBLEPOINT\_OFF or DOUBLEPOINT\_ON.

### Return values:

**None**

Definition at line **711** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **ASCII\_CHAR\_0**, **ASCII\_CHAR\_APOSTROPHE**,  
**ASCII\_CHAR\_AT\_SYMBOL**, **ASCII\_CHAR\_LEFT\_OPEN\_BRACE**,  
**ASCII\_CHAR\_LEFT\_OPEN\_BRACKET**, **C\_CLOSEPARMAP**,  
**C\_DMAP**, **C\_FULL**, **C\_MINUS**, **C\_MMAP**, **C\_NMAP**,  
**C\_OPENPARMAP**, **C\_PERCENT\_1**, **C\_PERCENT\_2**, **C\_PLUS**,  
**C\_SLATCH**, **C\_STAR**, **C\_UMAP**, **CapLetterMap**, **Digit**,  
**DOUBLEPOINT\_ON**, **NumberMap**, and **POINT\_ON**.

Referenced by **WriteChar()**.

```
static void LCD_MspDeInit ( LCD_HandleTypeDef * hLCD ) [static]
```

DeInitialize the LCD MSP.

**Parameters:**

**hlcd,**: LCD handle

**Return values:**

**None**

Definition at line **669** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **LCD\_GPIO\_BANKA\_PINS**, **LCD\_GPIO\_BANKB\_PINS**, **LCD\_GPIO\_BANKC\_PINS**, and **LCD\_GPIO\_BANKD\_PINS**.

Referenced by **BSP\_LCD\_GLASS\_DeInit()**.

**static void LCD\_MspInit ( LCD\_HandleTypeDef \* **hlcd** ) [static]**

Initialize the LCD MSP.

**Parameters:**

**hlcd,**: LCD handle

**Return values:**

**None**

Definition at line **603** of file **stm32l476g\_discovery\_glass\_lcd.c**.

References **LCD\_GPIO\_BANKA\_PINS**, **LCD\_GPIO\_BANKB\_PINS**, **LCD\_GPIO\_BANKC\_PINS**, and **LCD\_GPIO\_BANKD\_PINS**.

Referenced by **BSP\_LCD\_GLASS\_Init()**.

**static void WriteChar ( uint8\_t \* **ch**,  
Point\_Typedef **Point**,**

```

        DoublePoint_Typedef Colon,
        DigitPosition_Typedef Position
    )
    [static]

```

Write a character in the LCD frame buffer.

#### Parameters:

- ch,:** the character to display.
- Point,:** a point to add in front of char This parameter can be: POINT\_OFF or POINT\_ON
- Colon,:** flag indicating if a colon character has to be added in front of displayed character. This parameter can be: DOUBLEPOINT\_OFF or DOUBLEPOINT\_ON.
- Position,:** position in the LCD of the character to write [1:6]

#### Return values:

**None**

Definition at line 828 of file [stm32l476g\\_discovery\\_glass\\_lcd.c](#).

References [Convert\(\)](#), [Digit](#), [LCD\\_DIGIT1\\_COM0](#), [LCD\\_DIGIT1\\_COM0\\_SEG\\_MASK](#), [LCD\\_DIGIT1\\_COM1](#), [LCD\\_DIGIT1\\_COM1\\_SEG\\_MASK](#), [LCD\\_DIGIT1\\_COM2](#), [LCD\\_DIGIT1\\_COM2\\_SEG\\_MASK](#), [LCD\\_DIGIT1\\_COM3](#), [LCD\\_DIGIT1\\_COM3\\_SEG\\_MASK](#), [LCD\\_DIGIT2\\_COM0](#), [LCD\\_DIGIT2\\_COM0\\_SEG\\_MASK](#), [LCD\\_DIGIT2\\_COM1](#), [LCD\\_DIGIT2\\_COM1\\_SEG\\_MASK](#), [LCD\\_DIGIT2\\_COM2](#), [LCD\\_DIGIT2\\_COM2\\_SEG\\_MASK](#), [LCD\\_DIGIT2\\_COM3](#), [LCD\\_DIGIT2\\_COM3\\_SEG\\_MASK](#), [LCD\\_DIGIT3\\_COM0](#), [LCD\\_DIGIT3\\_COM0\\_SEG\\_MASK](#), [LCD\\_DIGIT3\\_COM1](#), [LCD\\_DIGIT3\\_COM1\\_SEG\\_MASK](#), [LCD\\_DIGIT3\\_COM2](#), [LCD\\_DIGIT3\\_COM2\\_SEG\\_MASK](#), [LCD\\_DIGIT3\\_COM3](#), [LCD\\_DIGIT3\\_COM3\\_SEG\\_MASK](#), [LCD\\_DIGIT4\\_COM0](#), [LCD\\_DIGIT4\\_COM0\\_1](#), [LCD\\_DIGIT4\\_COM0\\_1\\_SEG\\_MASK](#), [LCD\\_DIGIT4\\_COM0\\_SEG\\_MASK](#), [LCD\\_DIGIT4\\_COM1](#), [LCD\\_DIGIT4\\_COM1\\_1](#), [LCD\\_DIGIT4\\_COM1\\_1\\_SEG\\_MASK](#),

LCD\_DIGIT4\_COM1\_SEG\_MASK, LCD\_DIGIT4\_COM2,  
LCD\_DIGIT4\_COM2\_1, LCD\_DIGIT4\_COM2\_1\_SEG\_MASK,  
LCD\_DIGIT4\_COM2\_SEG\_MASK, LCD\_DIGIT4\_COM3,  
LCD\_DIGIT4\_COM3\_1, LCD\_DIGIT4\_COM3\_1\_SEG\_MASK,  
LCD\_DIGIT4\_COM3\_SEG\_MASK, LCD\_DIGIT5\_COM0,  
LCD\_DIGIT5\_COM0\_1, LCD\_DIGIT5\_COM0\_1\_SEG\_MASK,  
LCD\_DIGIT5\_COM0\_SEG\_MASK, LCD\_DIGIT5\_COM1,  
LCD\_DIGIT5\_COM1\_1, LCD\_DIGIT5\_COM1\_1\_SEG\_MASK,  
LCD\_DIGIT5\_COM1\_SEG\_MASK, LCD\_DIGIT5\_COM2,  
LCD\_DIGIT5\_COM2\_1, LCD\_DIGIT5\_COM2\_1\_SEG\_MASK,  
LCD\_DIGIT5\_COM2\_SEG\_MASK, LCD\_DIGIT5\_COM3,  
LCD\_DIGIT5\_COM3\_1, LCD\_DIGIT5\_COM3\_1\_SEG\_MASK,  
LCD\_DIGIT5\_COM3\_SEG\_MASK, LCD\_DIGIT6\_COM0,  
LCD\_DIGIT6\_COM0\_SEG\_MASK, LCD\_DIGIT6\_COM1,  
LCD\_DIGIT6\_COM1\_SEG\_MASK, LCD\_DIGIT6\_COM2,  
LCD\_DIGIT6\_COM2\_SEG\_MASK, LCD\_DIGIT6\_COM3,  
LCD\_DIGIT6\_COM3\_SEG\_MASK, LCD\_DIGIT\_POSITION\_1,  
LCD\_DIGIT\_POSITION\_2, LCD\_DIGIT\_POSITION\_3,  
LCD\_DIGIT\_POSITION\_4, LCD\_DIGIT\_POSITION\_5,  
LCD\_DIGIT\_POSITION\_6, LCD\_SEG0\_SHIFT, LCD\_SEG10\_SHIFT,  
LCD\_SEG11\_SHIFT, LCD\_SEG12\_SHIFT, LCD\_SEG13\_SHIFT,  
LCD\_SEG14\_SHIFT, LCD\_SEG15\_SHIFT, LCD\_SEG16\_SHIFT,  
LCD\_SEG17\_SHIFT, LCD\_SEG18\_SHIFT, LCD\_SEG19\_SHIFT,  
LCD\_SEG1\_SHIFT, LCD\_SEG20\_SHIFT, LCD\_SEG21\_SHIFT,  
LCD\_SEG22\_SHIFT, LCD\_SEG23\_SHIFT, LCD\_SEG2\_SHIFT,  
LCD\_SEG3\_SHIFT, LCD\_SEG4\_SHIFT, LCD\_SEG5\_SHIFT,  
LCD\_SEG6\_SHIFT, LCD\_SEG7\_SHIFT, LCD\_SEG8\_SHIFT,  
LCD\_SEG9\_SHIFT, and LCDHandle.

Referenced by **BSP\_LCD\_GLASS\_DisplayChar()**,  
**BSP\_LCD\_GLASS\_DisplayStrDeci()**, and  
**BSP\_LCD\_GLASS\_DisplayString()**.



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## Private Macros

[STM32L476G-DISCOVERY AUDIO](#)

## Defines

#define	<b>SAIClockDivider</b> (__FREQUENCY__)
#define	<b>DFSDMOverSampling</b> (__FREQUENCY__)
#define	<b>DFSDMClockDivider</b> (__FREQUENCY__)
#define	<b>DFSDMFilterOrder</b> (__FREQUENCY__)
#define	<b>DFSDMRightBitShift</b> (__FREQUENCY__)
#define	<b>SaturaLH</b> (N, L, H) (((N)<(L))? (L):(((N)>(H))? (H):(N)))

## Define Documentation

**#define DFSDMClockDivider ( \_\_FREQUENCY\_\_ )**

**Value:**

```
(__FREQUENCY__ == AUDIO_FREQUENCY_8K) ? 24 \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_11K) ?
4 \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_16K) ?
24 \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_22K) ?
4 \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_32K) ?
24 \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_44K) ?
4 \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_48K) ?
32 : 32 \
```

Definition at line **208** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **AUDIO\_DFSDMx\_Init()**.

**#define DFSDMFilterOrder ( \_\_FREQUENCY\_\_ )**

**Value:**

```
(__FREQUENCY__ == AUDIO_FREQUENCY_8K) ? DFSDM_FI
LTER_SINC3_ORDER \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_11K) ?
DFSDM_FILTER_SINC3_ORDER \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_16K) ?
DFSDM_FILTER_SINC3_ORDER \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_22K) ?
DFSDM_FILTER_SINC3_ORDER \
      : (__FREQUENCY__ == AUDIO_FREQUENCY_32K) ?
DFSDM_FILTER_SINC4_ORDER \
```

```

        : ( __FREQUENCY__ == AUDIO_FREQUENCY_44K) ?
DFSDM_FILTER_SINC4_ORDER \
        : ( __FREQUENCY__ == AUDIO_FREQUENCY_48K) ?
DFSDM_FILTER_SINC4_ORDER : DFSDM_FILTER_SINC5_ORDER \

```

Definition at line **217** of file [stm32l476g\\_discovery\\_audio.c](#).

Referenced by [AUDIO\\_DFSDMx\\_Init\(\)](#).

**#define DFSDMOverSampling ( \_\_FREQUENCY\_\_ )**

**Value:**

```

( __FREQUENCY__ == AUDIO_FREQUENCY_8K) ? 256 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_11K) ?
256 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_16K) ?
128 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_22K) ?
128 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_32K) ?
64 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_44K) ?
64 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_48K) ?
32 : 16 \

```

Definition at line **199** of file [stm32l476g\\_discovery\\_audio.c](#).

Referenced by [AUDIO\\_DFSDMx\\_Init\(\)](#).

**#define DFSDMRightBitShift ( \_\_FREQUENCY\_\_ )**

**Value:**

```

( __FREQUENCY__ == AUDIO_FREQUENCY_8K) ? 2 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_11K) ?

```

```

3 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_16K ) ?
2 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_22K ) ?
0 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_32K ) ?
3 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_44K ) ?
3 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_48K ) ?
7 : 0 \

```

Definition at line **226** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **AUDIO\_DFSDMx\_Init()**.

**#define SAIClockDivider ( \_\_FREQUENCY\_\_ )**

**Value:**

```

( __FREQUENCY__ == AUDIO_FREQUENCY_8K ) ? 12 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_11K ) ?
2 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_16K ) ?
6 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_22K ) ?
1 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_32K ) ?
3 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_44K ) ?
0 \
    : ( __FREQUENCY__ == AUDIO_FREQUENCY_48K ) ?
2 : 1 \

```

Definition at line **189** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **AUDIO\_SAIx\_Init()**, and  
**BSP\_AUDIO\_OUT\_SetFrequency()**.

```
#define SaturateLH ( N,  
                    L,  
                    H  
                    )  (((N)<(L))?(L):(((N)>(H))?(H):(N)))
```

Definition at line **236** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **HAL\_DFSDM\_FilterRegConvCpltCallback()**, and  
**HAL\_DFSDM\_FilterRegConvHalfCpltCallback()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## Exported Defines

[STM32L476G-DISCOVERY IDD](#)

## Defines

#define	<b>DISCOVERY_IDD_SHUNT0_VALUE</b>	((uint16_t) 1000)	Shunt values on discovery in milli ohms.
#define	<b>DISCOVERY_IDD_SHUNT1_VALUE</b>	((uint16_t) 24)	
#define	<b>DISCOVERY_IDD_SHUNT2_VALUE</b>	((uint16_t) 620)	
#define	<b>DISCOVERY_IDD_SHUNT4_VALUE</b>	((uint16_t) 10000)	
#define	<b>DISCOVERY_IDD_SHUNT0_STABDELAY</b>	((uint8_t) 149)	Shunt stabilization delay on discovery in milli ohms.
#define	<b>DISCOVERY_IDD_SHUNT1_STABDELAY</b>	((uint8_t) 149)	
#define	<b>DISCOVERY_IDD_SHUNT2_STABDELAY</b>	((uint8_t) 149)	
#define	<b>DISCOVERY_IDD_SHUNT4_STABDELAY</b>	((uint8_t) 255)	
#define	<b>DISCOVERY_IDD_AMPLI_GAIN</b>	((uint16_t) 4967)	IDD Ampli Gain on discovery.
#define	<b>DISCOVERY_IDD_VDD_MIN</b>	((uint16_t) 2000)	IDD Vdd Min on discovery.



## Define Documentation

**#define DISCOVERY\_IDD\_AMPLI\_GAIN** ((uint16\_t) 4967)

IDD Ampli Gain on discovery.

value is gain \* 100

Definition at line **109** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_SHUNT0\_STABDELAY** ((uint8\_t) 149)

Shunt stabilization delay on discovery in milli ohms.

value in millisec

Definition at line **100** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_SHUNT0\_VALUE** ((uint16\_t) 1000)

Shunt values on discovery in milli ohms.

value in milliohm

Definition at line **92** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_SHUNT1\_STABDELAY** ((uint8\_t) 149)

value in millisec

Definition at line **101** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_SHUNT1\_VALUE ((uint16\_t) 24)**

value in ohm

Definition at line **93** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_SHUNT2\_STABDELAY ((uint8\_t) 149)**

value in millisec

Definition at line **102** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_SHUNT2\_VALUE ((uint16\_t) 620)**

value in ohm

Definition at line **94** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_SHUNT4\_STABDELAY ((uint8\_t) 255)**

value in millisec

Definition at line **103** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_SHUNT4\_VALUE ((uint16\_t) 10000)**

value in ohm

Definition at line **95** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

**#define DISCOVERY\_IDD\_VDD\_MIN ((uint16\_t) 2000)**

IDD Vdd Min on discovery.

value in millivolt

Definition at line **117** of file **stm32l476g\_discovery\_idd.h**.

Referenced by **BSP\_IDD\_Init()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## Exported Macros

[STM32L476G-DISCOVERY AUDIO](#)

## Defines

```
#define DMA_MAX(_X_) (((_X_) <= DMA_MAX_SIZE)?  
(_X_):DMA_MAX_SIZE)
```

---

## Define Documentation

```
#define DMA_MAX ( _X_ ) (((_X_) <= DMA_MAX_SIZE)? (_X_):DMA_MAX_SIZE)
```

Definition at line **209** of file **stm32l476g\_discovery\_audio.h**.

Referenced by **BSP\_AUDIO\_OUT\_Play()**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## **BUTTON Constants**

[Exported Constants](#)

## Defines

#define	<b>JOYn</b>	5
#define	<b>RIGHT_JOY_PIN</b>	GPIO_PIN_2 /* PA.02 */ Joystick Right push-button.
#define	<b>RIGHT_JOY_GPIO_PORT</b>	GPIOA
#define	<b>RIGHT_JOY_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOA_CLK_ENABLE()
#define	<b>RIGHT_JOY_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOA_CLK_DISABLE()
#define	<b>RIGHT_JOY_EXTI_IRQn</b>	EXTI2_IRQn
#define	<b>LEFT_JOY_PIN</b>	GPIO_PIN_1 /* PA.01 */ Joystick Left push-button.
#define	<b>LEFT_JOY_GPIO_PORT</b>	GPIOA
#define	<b>LEFT_JOY_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOA_CLK_ENABLE()
#define	<b>LEFT_JOY_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOA_CLK_DISABLE()
#define	<b>LEFT_JOY_EXTI_IRQn</b>	EXTI1_IRQn
#define	<b>UP_JOY_PIN</b>	GPIO_PIN_3 /* PA.03 */ Joystick Up push-button.
#define	<b>UP_JOY_GPIO_PORT</b>	GPIOA
#define	<b>UP_JOY_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOA_CLK_ENABLE()
#define	<b>UP_JOY_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOA_CLK_DISABLE()
#define	<b>UP_JOY_EXTI_IRQn</b>	EXTI3_IRQn
#define	<b>DOWN_JOY_PIN</b>	GPIO_PIN_5 /* PA.05 */ Joystick Down push-button.
#define	<b>DOWN_JOY_GPIO_PORT</b>	GPIOA
#define	<b>DOWN_JOY_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOA_CLK_ENABLE()
#define	<b>DOWN_JOY_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOA_CLK_DISABLE()
#define	<b>DOWN_JOY_EXTI_IRQn</b>	EXTI9_5_IRQn
#define	<b>SEL_JOY_PIN</b>	GPIO_PIN_0 /* PA.00 */ Joystick Sel push-button.
#define	<b>SEL_JOY_GPIO_PORT</b>	GPIOA
#define	<b>SEL_JOY_GPIO_CLK_ENABLE()</b>	__HAL_RCC_GPIOA_CLK_ENABLE()
#define	<b>SEL_JOY_GPIO_CLK_DISABLE()</b>	__HAL_RCC_GPIOA_CLK_DISABLE()
#define	<b>SEL_JOY_EXTI_IRQn</b>	EXTI0_IRQn



```
#define JOYx_GPIO_CLK_ENABLE(__JOY__)  
#define JOYx_GPIO_CLK_DISABLE(__JOY__)  
#define JOY_ALL_PINS (RIGHT_JOY_PIN | LEFT_JOY_PIN | UP_J  
DOWN_JOY_PIN | SEL_JOY_PIN)
```

---

## Define Documentation

**#define DOWN\_JOY\_EXTI\_IRQn** EXTI9\_5\_IRQn

Definition at line 223 of file [stm32l476g\\_discovery.h](#).

**#define DOWN\_JOY\_GPIO\_CLK\_DISABLE ( )** \_\_HAL\_RCC\_GPIOA

Definition at line 222 of file [stm32l476g\\_discovery.h](#).

**#define DOWN\_JOY\_GPIO\_CLK\_ENABLE ( )** \_\_HAL\_RCC\_GPIOA

Definition at line 221 of file [stm32l476g\\_discovery.h](#).

**#define DOWN\_JOY\_GPIO\_PORT** GPIOA

Definition at line 220 of file [stm32l476g\\_discovery.h](#).

**#define DOWN\_JOY\_PIN** GPIO\_PIN\_5 /\* PA.05 \*/

Joystick Down push-button.

Definition at line 219 of file [stm32l476g\\_discovery.h](#).

**#define JOY\_ALL\_PINS** (RIGHT\_JOY\_PIN | LEFT\_JOY\_PIN | UP\_J

Definition at line 246 of file [stm32l476g\\_discovery.h](#).

**#define JOYn** 5

Definition at line **187** of file **stm32l476g\_discovery.h**.

Referenced by **BSP\_JOY\_DeInit()**, **BSP\_JOY\_GetState()**, and **BSP\_JOY\_Init()**.

**#define JOYx\_GPIO\_CLK\_DISABLE ( \_\_JOY\_\_ )**

**Value:**

```
do { if((__JOY__) == JOY_SEL)    { SEL_JOY_GPIO_CLK_DISABLE();    } else \
                                                                    if
((__JOY__) == JOY_DOWN)    { DOWN_JOY_GPIO_CLK_DISABLE();    } else \
                                                                    if
((__JOY__) == JOY_LEFT)    { LEFT_JOY_GPIO_CLK_DISABLE();    } else \
                                                                    if
((__JOY__) == JOY_RIGHT)    { RIGHT_JOY_GPIO_CLK_DISABLE();    } else \
                                                                    if
((__JOY__) == JOY_UP)    { UP_JOY_GPIO_CLK_DISABLE();    } } while(0)
```

Definition at line **240** of file **stm32l476g\_discovery.h**.

**#define JOYx\_GPIO\_CLK\_ENABLE ( \_\_JOY\_\_ )**

**Value:**

```
do { if((__JOY__) == JOY_SEL)    { SEL_JOY_GPIO_CLK_ENABLE();    } else \
                                                                    if
((__JOY__) == JOY_DOWN)    { DOWN_JOY_GPIO_CLK_ENABLE();    } else \
                                                                    if
((__JOY__) == JOY_LEFT)    { LEFT_JOY_GPIO_CLK_ENABLE();    }
```

```

LE()); } else \
                                                    if
((__JOY__) == JOY_RIGHT) { RIGHT_JOY_GPIO_CLK_ENA
BLE(); } else \
                                                    if
((__JOY__) == JOY_UP)      { UP_JOY_GPIO_CLK_ENABLE
()); } } while(0)

```

Definition at line **234** of file **stm32l476g\_discovery.h**.

Referenced by **BSP\_JOY\_DeInit()**, and **BSP\_JOY\_Init()**.

```

#define LEFT_JOY_EXTI_IRQn  EXTI1_IRQn

```

Definition at line **205** of file **stm32l476g\_discovery.h**.

```

#define LEFT_JOY_GPIO_CLK_DISABLE ( )  __HAL_RCC_GPIOA_

```

Definition at line **204** of file **stm32l476g\_discovery.h**.

```

#define LEFT_JOY_GPIO_CLK_ENABLE ( )  __HAL_RCC_GPIOA_

```

Definition at line **203** of file **stm32l476g\_discovery.h**.

```

#define LEFT_JOY_GPIO_PORT  GPIOA

```

Definition at line **202** of file **stm32l476g\_discovery.h**.

```

#define LEFT_JOY_PIN  GPIO_PIN_1 /* PA.01 */

```

Joystick Left push-button.

Definition at line **201** of file **stm32l476g\_discovery.h**.

```
#define RIGHT_JOY_EXTI_IRQn  EXTI2_IRQn
```

Definition at line **196** of file **stm32l476g\_discovery.h**.

```
#define RIGHT_JOY_GPIO_CLK_DISABLE ( )  __HAL_RCC_GPIOA_
```

Definition at line **195** of file **stm32l476g\_discovery.h**.

```
#define RIGHT_JOY_GPIO_CLK_ENABLE ( )  __HAL_RCC_GPIOA
```

Definition at line **194** of file **stm32l476g\_discovery.h**.

```
#define RIGHT_JOY_GPIO_PORT  GPIOA
```

Definition at line **193** of file **stm32l476g\_discovery.h**.

```
#define RIGHT_JOY_PIN  GPIO_PIN_2 /* PA.02 */
```

Joystick Right push-button.

Definition at line **192** of file **stm32l476g\_discovery.h**.

```
#define SEL_JOY_EXTI_IRQn  EXTI0_IRQn
```

Definition at line **232** of file **stm32l476g\_discovery.h**.

```
#define SEL_JOY_GPIO_CLK_DISABLE ( )  __HAL_RCC_GPIOA_ (
```

Definition at line **231** of file **stm32l476g\_discovery.h**.

```
#define SEL_JOY_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIOA_C
```

Definition at line **230** of file **stm32l476g\_discovery.h**.

```
#define SEL_JOY_GPIO_PORT GPIOA
```

Definition at line **229** of file **stm32l476g\_discovery.h**.

```
#define SEL_JOY_PIN GPIO_PIN_0 /* PA.00 */
```

Joystick Sel push-button.

Definition at line **228** of file **stm32l476g\_discovery.h**.

```
#define UP_JOY_EXTI_IRQn EXTI3_IRQn
```

Definition at line **214** of file **stm32l476g\_discovery.h**.

```
#define UP_JOY_GPIO_CLK_DISABLE ( ) __HAL_RCC_GPIOA_C
```

Definition at line **213** of file **stm32l476g\_discovery.h**.

```
#define UP_JOY_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIOA_C
```

Definition at line **212** of file **stm32l476g\_discovery.h**.

```
#define UP_JOY_GPIO_PORT GPIOA
```

Definition at line **211** of file **stm32l476g\_discovery.h**.

```
#define UP_JOY_PIN GPIO_PIN_3 /* PA.03 */
```

Joystick Up push-button.

Definition at line **210** of file **stm32l476g\_discovery.h**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Private Functions

[STM32L476G-DISCOVERY Common](#)



## Functions

void	<b>EEPROM_I2C_IO_Init</b> (void)
HAL_StatusTypeDef	<b>EEPROM_I2C_IO_WriteData</b> (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize)
HAL_StatusTypeDef	<b>EEPROM_I2C_IO_ReadData</b> (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize)
HAL_StatusTypeDef	<b>EEPROM_I2C_IO_IsDeviceReady</b> (uint16_t DevAddress, uint32_t Trials)
void	<b>MAGNETO_IO_ITConfig</b> (void)
void	<b>IOE_Init</b> (void)
void	<b>IOE_ITConfig</b> (void)
void	<b>IOE_Delay</b> (uint32_t Delay)
void	<b>IOE_Write</b> (uint8_t Addr, uint8_t Reg, uint8_t Value)
uint8_t	<b>IOE_Read</b> (uint8_t Addr, uint8_t Reg)
uint16_t	<b>IOE_ReadMultiple</b> (uint8_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length)

## Function Documentation

**void** **EEPROM\_I2C\_IO\_Init** ( **void** )

---

**HAL\_StatusTypeDef** **EEPROM\_I2C\_IO\_IsDeviceReady** ( **uint16\_t** **DevAd**  
**uint32\_t** **Tr**  
)

---

**HAL\_StatusTypeDef** **EEPROM\_I2C\_IO\_ReadData** ( **uint16\_t** **DevAd**  
**uint16\_t** **MemAc**  
**uint8\_t\*** **pBuffer**  
**uint32\_t** **BufferS**  
)

---

**HAL\_StatusTypeDef** **EEPROM\_I2C\_IO\_WriteData** ( **uint16\_t** **DevAd**  
**uint16\_t** **MemAc**  
**uint8\_t\*** **pBuffer**  
**uint32\_t** **BufferS**  
)

---

**void** **IOE\_Delay** ( **uint32\_t** **Delay** )

---

**void** **IOE\_Init** ( **void** )

---

**void** **IOE\_ITConfig** ( **void** )

---

**uint8\_t** **IOE\_Read** ( **uint8\_t** **Addr**,  
**uint8\_t** **Reg**  
)

---

```
uint16_t IOE_ReadMultiple ( uint8_t  Addr,  
                             uint8_t  Reg,  
                             uint8_t * Buffer,  
                             uint16_t Length  
                             )
```

```
void IOE_Write ( uint8_t Addr,  
                 uint8_t Reg,  
                 uint8_t Value  
                 )
```

```
void MAGNETO_IO_ITConfig ( void )
```

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Enumerations](#)

## Exported Types

[STM32L476G-DISCOVERY GYROSCOPE](#)

## Enumerations

```
enum GYRO_StatusTypeDef { GYRO_OK = 0, GYRO_ERROR = 1,  
  GYRO_TIMEOUT = 2 }
```

---

## Enumeration Type Documentation

enum **GYRO\_StatusTypeDef**

**Enumerator:**

*GYRO\_OK*

*GYRO\_ERROR*

*GYRO\_TIMEOUT*

Definition at line **67** of file **stm32l476g\_discovery\_gyroscope.h**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Variables](#)

## Private Variables

[STM32L476G-DISCOVERY GYROSCOPE](#)

## Variables

---

static GYRO_DrvTypeDef *	<b>GyroscopeDrv</b>
--------------------------	---------------------

---



## Variable Documentation

**GYRO\_DrvTypeDef\* GyroscopeDrv** [static]

Definition at line **82** of file **stm32l476g\_discovery\_gyroscope.c**.

Referenced by **BSP\_GYRO\_DisableIT()**, **BSP\_GYRO\_EnableIT()**, **BSP\_GYRO\_GetXYZ()**, **BSP\_GYRO\_Init()**, **BSP\_GYRO\_ITConfig()**, **BSP\_GYRO\_LowPower()**, **BSP\_GYRO\_ReadID()**, and **BSP\_GYRO\_Reset()**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Variables](#)

## Private Variables

[STM32L476G-DISCOVERY AUDIO](#)

## Variables

static <b>AUDIO_OUT_TypeDef</b>	<b>hAudioOut</b>
---------------------------------	------------------

static <b>AUDIO_IN_TypeDef</b>	<b>hAudioIn</b>
--------------------------------	-----------------

static <b>DMA_HandleTypeDef</b>	<b>hDmaSai</b>
---------------------------------	----------------

## Variable Documentation

**AUDIO\_IN\_TypeDef hAudioIn** [static]

Definition at line **250** of file **stm32l476g\_discovery\_audio.c**.

**AUDIO\_OUT\_TypeDef hAudioOut** [static]

Definition at line **247** of file **stm32l476g\_discovery\_audio.c**.

**DMA\_HandleTypeDef hDmaSai** [static]

Definition at line **253** of file **stm32l476g\_discovery\_audio.c**.

Referenced by **HAL\_SAI\_MspDeInit()**, and **HAL\_SAI\_MspInit()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Enumerations](#)

## IDD Config

[Exported Types](#)

## Enumerations

```
enum  IDD_StatusTypeDef { IDD_OK = 0, IDD_TIMEOUT = 1,  
    IDD_ZERO_VALUE = 2, IDD_ERROR = 0xFF }
```

## Enumeration Type Documentation

enum **IDD\_StatusTypeDef**

**Enumerator:**

*IDD\_OK*

*IDD\_TIMEOUT*

*IDD\_ZERO\_VALUE*

*IDD\_ERROR*

Definition at line **70** of file **stm32l476g\_discovery\_idd.h**.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Variables](#)

## Private Variables

[STM32L476G-DISCOVERY IDD](#)



## Variables

```
static IDD_DrvTypeDef * IddDrv
```

---

## Variable Documentation

**IDD\_DrvTypeDef\* IddDrv** [static]

Definition at line 69 of file `stm32l476g_discovery_idd.c`.

Referenced by `BSP_IDD_ClearIT()`, `BSP_IDD_Config()`, `BSP_IDD_DeInit()`, `BSP_IDD_DisableIT()`, `BSP_IDD_EnableIT()`, `BSP_IDD_ErrorClearIT()`, `BSP_IDD_ErrorDisableIT()`, `BSP_IDD_ErrorEnableIT()`, `BSP_IDD_ErrorGetCode()`, `BSP_IDD_ErrorGetITStatus()`, `BSP_IDD_GetITStatus()`, `BSP_IDD_GetValue()`, `BSP_IDD_Init()`, `BSP_IDD_LowPower()`, `BSP_IDD_Reset()`, `BSP_IDD_StartMeasure()`, and `BSP_IDD_WakeUp()`.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Enumerations](#)

## Exported Types

[STM32L476G-DISCOVERY Common](#)

## Enumerations

```
enum Led_TypeDef { LED4 = 0, LED5 = 1, LED_RED = LED4,
LED_GREEN = LED5 }
```

LED Types Definition. [More...](#)

```
enum JOYState_TypeDef {
    JOY_SEL = 0, JOY_LEFT = 1, JOY_RIGHT = 2,
    JOY_DOWN = 3,
    JOY_UP = 4, JOY_NONE = 5
}
```

JOYSTICK Types Definition. [More...](#)

```
enum JOYMode_TypeDef { JOY_MODE_GPIO = 0,  
                      JOY_MODE_EXTI = 1 }
```

```
enum SupplyMode_TypeDef { SUPPLY_MODE_ERROR = 0,  
    SUPPLY_MODE_EXTERNAL = 1,  
    SUPPLY_MODE_BATTERY = 2 }
```

## Enumeration Type Documentation

### enum **JOYMode\_TypeDef**

#### Enumerator:

*JOY\_MODE\_GPIO*

*JOY\_MODE\_EXTI*

Definition at line **108** of file **stm32l476g\_discovery.h**.

### enum **JOYState\_TypeDef**

JOYSTICK Types Definition.

#### Enumerator:

*JOY\_SEL*

*JOY\_LEFT*

*JOY\_RIGHT*

*JOY\_DOWN*

*JOY\_UP*

*JOY\_NONE*

Definition at line **98** of file **stm32l476g\_discovery.h**.

### enum **Led\_TypeDef**

LED Types Definition.

#### Enumerator:

*LED4*

*LED5*

*LED\_RED*

*LED\_GREEN*

Definition at line **78** of file **stm32l476g\_discovery.h**.

## **enum SupplyMode\_TypeDef**

### **Enumerator:**

*SUPPLY\_MODE\_ERROR*

*SUPPLY\_MODE\_EXTERNAL*

*SUPPLY\_MODE\_BATTERY*

Definition at line **114** of file **stm32l476g\_discovery.h**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## LED Constants

[Exported Constants](#)

## Defines

#define	LEDn	2
#define	LED4_PIN	GPIO_PIN_2
#define	LED4_GPIO_PORT	GPIOB
#define	LED4_GPIO_CLK_ENABLE()	__HAL_RCC_GPIOB_CLK_EN
#define	LED4_GPIO_CLK_DISABLE()	__HAL_RCC_GPIOB_CLK_D
#define	LED5_PIN	GPIO_PIN_8
#define	LED5_GPIO_PORT	GPIOE
#define	LED5_GPIO_CLK_ENABLE()	__HAL_RCC_GPIOE_CLK_E
#define	LED5_GPIO_CLK_DISABLE()	__HAL_RCC_GPIOE_CLK_D
#define	LEDx_GPIO_CLK_ENABLE(__LED__)	
#define	LEDx_GPIO_CLK_DISABLE(__LED__)	



## Define Documentation

**#define LED4\_GPIO\_CLK\_DISABLE ( )** \_\_HAL\_RCC\_GPIOB\_CLK

Definition at line **149** of file **stm32l476g\_discovery.h**.

**#define LED4\_GPIO\_CLK\_ENABLE ( )** \_\_HAL\_RCC\_GPIOB\_CLK

Definition at line **148** of file **stm32l476g\_discovery.h**.

**#define LED4\_GPIO\_PORT** GPIOB

Definition at line **147** of file **stm32l476g\_discovery.h**.

**#define LED4\_PIN** GPIO\_PIN\_2

Definition at line **146** of file **stm32l476g\_discovery.h**.

**#define LED5\_GPIO\_CLK\_DISABLE ( )** \_\_HAL\_RCC\_GPIOE\_CLK

Definition at line **154** of file **stm32l476g\_discovery.h**.

**#define LED5\_GPIO\_CLK\_ENABLE ( )** \_\_HAL\_RCC\_GPIOE\_CLK

Definition at line **153** of file **stm32l476g\_discovery.h**.

**#define LED5\_GPIO\_PORT** GPIOE

Definition at line **152** of file **stm32l476g\_discovery.h**.

**#define LED5\_PIN** GPIO\_PIN\_8

Definition at line **151** of file **stm32l476g\_discovery.h**.

**#define LEDn** 2

Definition at line **143** of file **stm32l476g\_discovery.h**.

**#define LEDx\_GPIO\_CLK\_DISABLE** ( \_\_LED\_\_ )

Value:

```
do { if((__LED__) == LED4) { LED4_GPIO_CLK_DISABLE  
(); } else \n  
if  
((__LED__) == LED5) { LED5_GPIO_CLK_DISABLE(); }  
} while(0)
```

Definition at line **159** of file **stm32l476g\_discovery.h**.

**#define LEDx\_GPIO\_CLK\_ENABLE** ( \_\_LED\_\_ )

Value:

```
do { if((__LED__) == LED4) { LED4_GPIO_CLK_ENABLE  
(); } else \n  
if  
((__LED__) == LED5) { LED5_GPIO_CLK_ENABLE(); } }  
while(0)
```

Definition at line **156** of file **stm32l476g\_discovery.h**.

Referenced by **BSP\_LED\_DeInit()**, and **BSP\_LED\_Init()**.



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Functions](#)

## Private Functions

[STM32L476G-DISCOVERY QSPI](#)

## Functions

static void	<b>QSPI_MspInit</b> (void) Initializes the QSPI MSP.
static void	<b>QSPI_MspDeInit</b> (void) De-Initializes the QSPI MSP.
static uint8_t	<b>QSPI_ResetMemory</b> (QSPI_HandleTypeDef *hqspi) This function reset the QSPI memory.
static uint8_t	<b>QSPI_DummyCyclesCfg</b> (QSPI_HandleTypeDef *hqspi) This function configure the dummy cycles on memory side.
static uint8_t	<b>QSPI_WriteEnable</b> (QSPI_HandleTypeDef *hqspi) This function send a Write Enable and wait it is effective.
static uint8_t	<b>QSPI_AutoPollingMemReady</b> (QSPI_HandleTypeDef *hqspi, uint32_t Timeout) This function read the SR of the memory and wait the EOP.

## Function Documentation

```
static uint8_t QSPI_AutoPollingMemReady ( QSPI_HandleTypeDef *  
                                           uint32_t  
                                           )
```

This function read the SR of the memory and wait the EOP.

### Parameters:

**hqspi,:** QSPI handle  
**Timeout,:** Timeout for auto-polling

### Return values:

**None**

Definition at line **859** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_ERROR**, **QSPI\_OK**, and **QSPIHandle**.

Referenced by **BSP\_QSPI\_Erase\_Block()**,  
**BSP\_QSPI\_Erase\_Chip()**, **BSP\_QSPI\_Write()**, and  
**QSPI\_ResetMemory()**.

```
static uint8_t QSPI_DummyCyclesCfg ( QSPI_HandleTypeDef * hqs
```

This function configure the dummy cycles on memory side.

### Parameters:

**hqspi,:** QSPI handle

### Return values:

**None**

Definition at line **754** of file **stm32l476g\_discovery\_qspi.c**.

References [QSPI\\_ERROR](#), [QSPI\\_OK](#), [QSPI\\_WriteEnable\(\)](#), and [QSPIHandle](#).

Referenced by [BSP\\_QSPI\\_Init\(\)](#).

**static void** [QSPI\\_MspDeInit](#) ( void ) [static]

De-Initializes the QSPI MSP.

**Return values:**

**None**

Definition at line [677](#) of file [stm32l476g\\_discovery\\_qspi.c](#).

Referenced by [BSP\\_QSPI\\_DeInit\(\)](#).

**static void** [QSPI\\_MsplInit](#) ( void ) [static]

Initializes the QSPI MSP.

**Return values:**

**None**

Definition at line [645](#) of file [stm32l476g\\_discovery\\_qspi.c](#).

Referenced by [BSP\\_QSPI\\_Init\(\)](#).

**static uint8\_t** [QSPI\\_ResetMemory](#) ( QSPI\_HandleTypeDef \* **hqspi** )

This function reset the QSPI memory.

**Parameters:**

**hqspi**,: QSPI handle

**Return values:**

**None**

Definition at line **712** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_AutoPollingMemReady()**, **QSPI\_ERROR**, **QSPI\_OK**, and **QSPIHandle**.

Referenced by **BSP\_QSPI\_Init()**.

**static uint8\_t QSPI\_WriteEnable ( QSPI\_HandleTypeDef \* **hqspi** ) [s**

This function send a Write Enable and wait it is effective.

**Parameters:**

**hqspi**,: QSPI handle

**Return values:**

**None**

Definition at line **813** of file **stm32l476g\_discovery\_qspi.c**.

References **QSPI\_ERROR**, **QSPI\_OK**, and **QSPIHandle**.

Referenced by **BSP\_QSPI\_Erase\_Block()**,  
**BSP\_QSPI\_Erase\_Chip()**, **BSP\_QSPI\_Erase\_Sector()**,  
**BSP\_QSPI\_ResumeErase()**, **BSP\_QSPI\_SuspendErase()**,  
**BSP\_QSPI\_Write()**, and **QSPI\_DummyCyclesCfg()**.



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Defines](#)

## Exported Constants

[STM32L476G-DISCOVERY QSPI](#)

## Defines

#define	<b>QSPI_OK</b>	((uint8_t)0x00)
#define	<b>QSPI_ERROR</b>	((uint8_t)0x01)
#define	<b>QSPI_BUSY</b>	((uint8_t)0x02)
#define	<b>QSPI_NOT_SUPPORTED</b>	((uint8_t)0x04)
#define	<b>QSPI_SUSPENDED</b>	((uint8_t)0x08)

## Define Documentation

**#define QSPI\_BUSY** ((uint8\_t)0x02)

Definition at line **70** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_GetStatus()**,  
**BSP\_QSPI\_ResumeErase()**, and **BSP\_QSPI\_SuspendErase()**.

**#define QSPI\_ERROR** ((uint8\_t)0x01)

Definition at line **69** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_DeInit()**,  
**BSP\_QSPI\_EnableMemoryMappedMode()**,  
**BSP\_QSPI\_Erase\_Block()**, **BSP\_QSPI\_Erase\_Chip()**,  
**BSP\_QSPI\_Erase\_Sector()**, **BSP\_QSPI\_GetStatus()**,  
**BSP\_QSPI\_Init()**, **BSP\_QSPI\_Read()**, **BSP\_QSPI\_ResumeErase()**,  
**BSP\_QSPI\_SuspendErase()**, **BSP\_QSPI\_Write()**,  
**QSPI\_AutoPollingMemReady()**, **QSPI\_DummyCyclesCfg()**,  
**QSPI\_ResetMemory()**, and **QSPI\_WriteEnable()**.

**#define QSPI\_NOT\_SUPPORTED** ((uint8\_t)0x04)

Definition at line **71** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_Init()**.

**#define QSPI\_OK** ((uint8\_t)0x00)

Definition at line **68** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_DeInit()**,

**BSP\_QSPI\_EnableMemoryMappedMode(),  
BSP\_QSPI\_Erase\_Block(), BSP\_QSPI\_Erase\_Chip(),  
BSP\_QSPI\_Erase\_Sector(), BSP\_QSPI\_GetInfo(),  
BSP\_QSPI\_GetStatus(), BSP\_QSPI\_Init(), BSP\_QSPI\_Read(),  
BSP\_QSPI\_ResumeErase(), BSP\_QSPI\_SuspendErase(),  
BSP\_QSPI\_Write(), QSPI\_AutoPollingMemReady(),  
QSPI\_DummyCyclesCfg(), QSPI\_ResetMemory(), and  
QSPI\_WriteEnable().**

**#define QSPI\_SUSPENDED ((uint8\_t)0x08)**

Definition at line **72** of file **stm32l476g\_discovery\_qspi.h**.

Referenced by **BSP\_QSPI\_GetStatus(),  
BSP\_QSPI\_ResumeErase(), and BSP\_QSPI\_SuspendErase().**

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Variables](#)

## Private Variables

[STM32L476G-DISCOVERY QSPI](#)

## Variables

QSPI_HandleTypeDef	<b>QSPIHandle</b>
--------------------	-------------------

---

## Variable Documentation

### QSPI\_HandleTypeDef QSPIHandle

Definition at line **87** of file **stm32l476g\_discovery\_qspi.c**.

Referenced by **BSP\_QSPI\_DeInit()**,  
**BSP\_QSPI\_EnableMemoryMappedMode()**,  
**BSP\_QSPI\_Erase\_Block()**, **BSP\_QSPI\_Erase\_Chip()**,  
**BSP\_QSPI\_Erase\_Sector()**, **BSP\_QSPI\_GetStatus()**,  
**BSP\_QSPI\_Init()**, **BSP\_QSPI\_Read()**, **BSP\_QSPI\_ResumeErase()**,  
**BSP\_QSPI\_SuspendErase()**, **BSP\_QSPI\_Write()**,  
**QSPI\_AutoPollingMemReady()**, **QSPI\_DummyCyclesCfg()**,  
**QSPI\_ResetMemory()**, and **QSPI\_WriteEnable()**.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Drivers</a>				

## Drivers Directory Reference



## Directories

directory **BSP**

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Drivers</a>	<a href="#">BSP</a>			

## BSP Directory Reference

## Directories

directory **STM32L476G-Discovery**

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		

## STM32L476G-Discovery Directory Reference

---

## Files

file [stm32l476g\\_discovery.c](#) [code]

This file provides a set of firmware functions to manage Leds, push-button and joystick of STM32L476G-Discovery board (MB1184)

file [stm32l476g\\_discovery.h](#) [code]

This file contains definitions for STM32L476G\_DISCOVERY's LEDs, push-buttons hardware resources (MB1184).

file [stm32l476g\\_discovery\\_accelerometer.c](#) [code]

This file provides a set of functions needed to manage the ACCELEROMETER MEMS available on STM32L476G-Discovery Kit.

file [stm32l476g\\_discovery\\_accelerometer.h](#) [code]

This file contains definitions for [stm32l476g\\_discovery\\_accelerometer.c](#) firmware driver.

file [stm32l476g\\_discovery\\_audio.c](#) [code]

This file provides a set of functions needed to manage the Audio driver for the STM32L476G-Discovery board.

file [stm32l476g\\_discovery\\_audio.h](#) [code]

This file contains the common defines and functions prototypes

for the [stm32l476g\\_discovery\\_audio.c](#) driver.

---

file [stm32l476g\\_discovery\\_compass.c](#) [code]

This file provides a set of functions needed to manage the E-Compass (ACCELEROMETER + MAGNETOMETER) MEMS LSM303C available on STM32L476G-Discovery board.

---

file [stm32l476g\\_discovery\\_compass.h](#) [code]

This file contains definitions for [stm32l476g\\_discovery\\_compass.c](#) firmware driver.

---

file [stm32l476g\\_discovery\\_glass\\_lcd.c](#) [code]

This file provides a set of functions needed to manage the LCD Glass driver for the STM32L476G-Discovery board.

---

file [stm32l476g\\_discovery\\_glass\\_lcd.h](#) [code]

Header file for [stm32l476g\\_discovery\\_glass\\_lcd.c](#) module.

---

file [stm32l476g\\_discovery\\_gyroscope.c](#) [code]

This file provides a set of functions needed to manage the L3GD20 MEMS accelerometer available on STM32L476G-Discovery board.

---

file [stm32l476g\\_discovery\\_gyroscope.h](#) [code]

This file contains definitions for [stm32l476g\\_discovery\\_gyroscope.c](#) firmware driver.

---

file [stm32l476g\\_discovery\\_idd.c](#) [code]

This file provides a set of firmware functions to manage the Idd measurement driver for STM32L476G-Discovery board.

---

file [stm32l476g\\_discovery\\_idd.h](#) [code]

Header file for [stm32l476g\\_discovery\\_idd.c](#) module.

---

file [stm32l476g\\_discovery\\_qspi.c](#) [code]

This file includes a standard driver for the N25Q128A QSPI memory mounted on STM32L476G-Discovery board.

---

file [stm32l476g\\_discovery\\_qspi.h](#) [code]

This file contains the common defines and functions prototypes for the [stm32l476g\\_discovery\\_qspi.c](#) driver.

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery.h

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery.h
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file contains definitions
for STM32L476G_DISCOVERY's LEDs,
00008      *           push-buttons hardware resources
(MB1184).
00009      ****
00010      * @attention
00011      *
00012      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
icroelectronics</center></h2>
00013      *
00014      * Redistribution and use in source and bin
ary forms, with or without modification,
00015      * are permitted provided that the followin
g conditions are met:
```



00016 \* 1. Redistributions of source code must  
retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

```

00035      *
00036      ****
00037      */
00038
00039 /* Define to prevent recursive inclusion ---
-----*/
00040 #ifndef __STM32L476G_DISCOVERY_H
00041 #define __STM32L476G_DISCOVERY_H
00042
00043 #ifdef __cplusplus
00044     extern "C" {
00045 #endif
00046
00047 /**
00048     * @brief Define for STM32L476G_DISCOVERY
board
00049     */
00050 #if !defined (USE_STM32L476G_DISCO_REVC) &&
!defined (USE_STM32L476G_DISCO_REVB) && !defined (
USE_STM32L476G_DISCO_REVA)
00051 #define USE_STM32L476G_DISCO_REVC
00052 #endif
00053
00054
00055 /* Includes -----
-----*/
00056 #include "stm32l4xx_hal.h"
00057
00058 /** @addtogroup BSP
00059     * @{
00060     */
00061
00062 /** @addtogroup STM32L476G_DISCOVERY
00063     * @{
00064     */
00065

```

```

00066 /** @addtogroup STM32L476G_DISCOVERY_Common
00067     * @{
00068     */
00069
00070 /** @defgroup STM32L476G_DISCOVERY_Exported_
Types Exported Types
00071     * @{
00072     */
00073
00074 /**
00075     * @brief LED Types Definition
00076     */
00077 #if defined (USE_STM32L476G_DISCO_REVC) || d
efined (USE_STM32L476G_DISCO_REVB)
00078 typedef enum
00079 {
00080     LED4 = 0,
00081     LED5 = 1,
00082     LED_RED    = LED4,
00083     LED_GREEN  = LED5
00084 }Led_TypeDef;
00085 #elif defined (USE_STM32L476G_DISCO_REVA)
00086 typedef enum
00087 {
00088     LED3 = 0,
00089     LED4 = 1,
00090     LED_RED    = LED3,
00091     LED_GREEN  = LED4
00092 }Led_TypeDef;
00093 #endif
00094
00095 /**
00096     * @brief JOYSTICK Types Definition
00097     */
00098 typedef enum
00099 {
00100     JOY_SEL    = 0,

```

```

00101     JOY_LEFT    = 1,
00102     JOY_RIGHT   = 2,
00103     JOY_DOWN     = 3,
00104     JOY_UP       = 4,
00105     JOY_NONE     = 5
00106 }JOYState_TypeDef;
00107
00108 typedef enum
00109 {
00110     JOY_MODE_GPIO = 0,
00111     JOY_MODE_EXTI = 1
00112 }JOYMode_TypeDef;
00113
00114 typedef enum
00115 {
00116     SUPPLY_MODE_ERROR = 0,
00117     SUPPLY_MODE_EXTERNAL = 1,
00118     SUPPLY_MODE_BATTERY = 2
00119 }SupplyMode_TypeDef;
00120
00121 /**
00122  * @}
00123  */
00124
00125 /** @defgroup STM32L476G_DISCOVERY_Exported_
Constants Exported Constants
00126  * @{
00127  */
00128
00129 /** @defgroup STM32L476G_DISCOVERY_BATTERY B
ATTERY Detection Constants
00130  * @{
00131  */
00132 #define BATTERY_DETECTION_PIN
GPIO_PIN_3
00133 #define BATTERY_DETECTION_GPIO_PORT
GPIOB

```

```

00134 #define BATTERY_DETECTION_GPIO_CLK_ENABLE()
    __HAL_RCC_GPIOB_CLK_ENABLE()
00135 #define BATTERY_DETECTION_GPIO_CLK_DISABLE()
    __HAL_RCC_GPIOB_CLK_DISABLE()
00136 /**
00137     * @}
00138     */
00139
00140 /** @defgroup STM32L476G_DISCOVERY_LED LED C
onstants
00141     * @{
00142     */
00143 #define LEDn                                2
00144
00145 #if defined (USE_STM32L476G_DISCO_REVC) || d
efined (USE_STM32L476G_DISCO_REVB)
00146 #define LED4_PIN                            GP
IO_PIN_2
00147 #define LED4_GPIO_PORT                      GP
IOB
00148 #define LED4_GPIO_CLK_ENABLE()              __
HAL_RCC_GPIOB_CLK_ENABLE()
00149 #define LED4_GPIO_CLK_DISABLE()            __
HAL_RCC_GPIOB_CLK_DISABLE()
00150
00151 #define LED5_PIN                            GP
IO_PIN_8
00152 #define LED5_GPIO_PORT                      GP
IOE
00153 #define LED5_GPIO_CLK_ENABLE()              __
HAL_RCC_GPIOE_CLK_ENABLE()
00154 #define LED5_GPIO_CLK_DISABLE()            __
HAL_RCC_GPIOE_CLK_DISABLE()
00155
00156 #define LEDx_GPIO_CLK_ENABLE(__LED__)      do
{ if((__LED__) == LED4) { LED4_GPIO_CLK_ENABLE();
} else \

```

```

00157     if((__LED__) == LED5) { LED5_GPIO_CLK_ENABLE();
    } } while(0)
00158
00159 #define LEDx_GPIO_CLK_DISABLE(__LED__)    do
    { if((__LED__) == LED4) { LED4_GPIO_CLK_DISABLE()
    ; } else \
00160     if((__LED__) == LED5) { LED5_GPIO_CLK_DISABLE()
    ; } } while(0)
00161
00162 #elif defined (USE_STM32L476G_DISCO_REVA)
00163 #define LED3_PIN                                GP
IO_PIN_2
00164 #define LED3_GPIO_PORT                                GP
IOB
00165 #define LED3_GPIO_CLK_ENABLE()                                __
HAL_RCC_GPIOB_CLK_ENABLE()
00166 #define LED3_GPIO_CLK_DISABLE()                                __
HAL_RCC_GPIOB_CLK_DISABLE()
00167
00168 #define LED4_PIN                                GP
IO_PIN_8
00169 #define LED4_GPIO_PORT                                GP
IOE
00170 #define LED4_GPIO_CLK_ENABLE()                                __
HAL_RCC_GPIOE_CLK_ENABLE()
00171 #define LED4_GPIO_CLK_DISABLE()                                __
HAL_RCC_GPIOE_CLK_DISABLE()
00172
00173 #define LEDx_GPIO_CLK_ENABLE(__LED__)    do
    { if((__LED__) == LED3) { LED3_GPIO_CLK_ENABLE();
    } else \
00174     if((__LED__) == LED4) { LED4_GPIO_CLK_ENABLE();
    } } while(0)
00175

```

```

00176 #define LEDx_GPIO_CLK_DISABLE(__LED__)    do
    { if((__LED__) == LED3) { LED3_GPIO_CLK_DISABLE()
; } else \
00177     if((__LED__) == LED4) { LED4_GPIO_CLK_DISABLE()
; } } while(0)
00178
00179 #endif
00180 /**
00181  * @}
00182  */
00183
00184 /** @defgroup STM32L476G_DISCOVERY_BUTTON  B
UTTON Constants
00185  * @{
00186  */
00187 #define JOYn                                5
00188
00189 /**
00190  * @brief Joystick Right push-button
00191  */
00192 #define RIGHT_JOY_PIN                        GP
IO_PIN_2  /* PA.02 */
00193 #define RIGHT_JOY_GPIO_PORT                GP
IOA
00194 #define RIGHT_JOY_GPIO_CLK_ENABLE()        __
HAL_RCC_GPIOA_CLK_ENABLE()
00195 #define RIGHT_JOY_GPIO_CLK_DISABLE()       __
HAL_RCC_GPIOA_CLK_DISABLE()
00196 #define RIGHT_JOY_EXTI_IRQn                EX
TI2_IRQn
00197
00198 /**
00199  * @brief Joystick Left push-button
00200  */
00201 #define LEFT_JOY_PIN                        GP
IO_PIN_1  /* PA.01 */

```

00202	#define LEFT_JOY_GPIO_PORT	GP
	IOA	
00203	#define LEFT_JOY_GPIO_CLK_ENABLE()	—
	HAL_RCC_GPIOA_CLK_ENABLE()	
00204	#define LEFT_JOY_GPIO_CLK_DISABLE()	—
	HAL_RCC_GPIOA_CLK_DISABLE()	
00205	#define LEFT_JOY_EXTI_IRQn	EX
	TI1_IRQn	
00206		
00207	/**	
00208	* @brief Joystick Up push-button	
00209	*/	
00210	#define UP_JOY_PIN	GP
	IO_PIN_3 /* PA.03 */	
00211	#define UP_JOY_GPIO_PORT	GP
	IOA	
00212	#define UP_JOY_GPIO_CLK_ENABLE()	—
	HAL_RCC_GPIOA_CLK_ENABLE()	
00213	#define UP_JOY_GPIO_CLK_DISABLE()	—
	HAL_RCC_GPIOA_CLK_DISABLE()	
00214	#define UP_JOY_EXTI_IRQn	EX
	TI3_IRQn	
00215		
00216	/**	
00217	* @brief Joystick Down push-button	
00218	*/	
00219	#define DOWN_JOY_PIN	GP
	IO_PIN_5 /* PA.05 */	
00220	#define DOWN_JOY_GPIO_PORT	GP
	IOA	
00221	#define DOWN_JOY_GPIO_CLK_ENABLE()	—
	HAL_RCC_GPIOA_CLK_ENABLE()	
00222	#define DOWN_JOY_GPIO_CLK_DISABLE()	—
	HAL_RCC_GPIOA_CLK_DISABLE()	
00223	#define DOWN_JOY_EXTI_IRQn	EX
	TI9_5_IRQn	
00224		



```

00225 /**
00226  * @brief Joystick Sel push-button
00227  */
00228 #define SEL_JOY_PIN                                GP
IO_PIN_0      /* PA.00 */
00229 #define SEL_JOY_GPIO_PORT                          GP
IOA
00230 #define SEL_JOY_GPIO_CLK_ENABLE()                  __
HAL_RCC_GPIOA_CLK_ENABLE()
00231 #define SEL_JOY_GPIO_CLK_DISABLE()                  __
HAL_RCC_GPIOA_CLK_DISABLE()
00232 #define SEL_JOY_EXTI_IRQn                          EX
TIO_IRQn
00233
00234 #define JOYx_GPIO_CLK_ENABLE(__JOY__)              do
{ if((__JOY__) == JOY_SEL)    { SEL_JOY_GPIO_CLK_E
NABLE();  } else \
00235
    if((__JOY__) == JOY_DOWN) { DOWN_JOY_GPIO_CLK_
ENABLE(); } else \
00236
    if((__JOY__) == JOY_LEFT) { LEFT_JOY_GPIO_CLK_
ENABLE(); } else \
00237
    if((__JOY__) == JOY_RIGHT) { RIGHT_JOY_GPIO_CLK
_ENABLE(); } else \
00238
    if((__JOY__) == JOY_UP)    { UP_JOY_GPIO_CLK_EN
ABLE(); } } while(0)
00239
00240 #define JOYx_GPIO_CLK_DISABLE(__JOY__)            do
{ if((__JOY__) == JOY_SEL)    { SEL_JOY_GPIO_CLK_D
ISABLE();  } else \
00241
    if((__JOY__) == JOY_DOWN) { DOWN_JOY_GPIO_CLK_
DISABLE(); } else \
00242

```

```

        if((__JOY__) == JOY_LEFT) { LEFT_JOY_GPIO_CLK_
DISABLE(); } else \
00243
        if((__JOY__) == JOY_RIGHT) { RIGHT_JOY_GPIO_CLK
_DISABLE(); } else \
00244
        if((__JOY__) == JOY_UP)      { UP_JOY_GPIO_CLK_DI
SABLE(); } } while(0)
00245
00246 #define JOY_ALL_PINS                                (R
IGHT_JOY_PIN | LEFT_JOY_PIN | UP_JOY_PIN | DOWN_JO
Y_PIN | SEL_JOY_PIN)
00247
00248 /**
00249  * @}
00250  */
00251
00252 /** @defgroup STM32L476G_DISCOVERY_BUS  BUS
Constants
00253  * @{
00254  */
00255 #if defined(HAL_SPI_MODULE_ENABLED)
00256 /**##### SPI2 #####
#####*/
00257 #define DISCOVERY_SPIx
SPI2
00258 #define DISCOVERY_SPIx_CLOCK_ENABLE()
__HAL_RCC_SPI2_CLK_ENABLE()
00259 #define DISCOVERY_SPIx_CLOCK_DISABLE()
__HAL_RCC_SPI2_CLK_DISABLE()
00260 #define DISCOVERY_SPIx_GPIO_PORT
GPIOD /* GPIOD */
00261 #define DISCOVERY_SPIx_AF
GPIO_AF5_SPI2
00262 #define DISCOVERY_SPIx_GPIO_CLK_ENABLE()
__HAL_RCC_GPIOD_CLK_ENABLE()
00263 #define DISCOVERY_SPIx_GPIO_CLK_DISABLE()

```

```

    __HAL_RCC_GPIOD_CLK_DISABLE()
00264 #define DISCOVERY_SPIx_GPIO_FORCE_RESET()
    __HAL_RCC_SPI2_FORCE_RESET()
00265 #define DISCOVERY_SPIx_GPIO_RELEASE_RESET()
    __HAL_RCC_SPI2_RELEASE_RESET()
00266 #define DISCOVERY_SPIx_SCK_PIN
    GPIO_PIN_1          /* PD.01 */
00267 #define DISCOVERY_SPIx_MISO_PIN
    GPIO_PIN_3          /* PD.03 */
00268 #define DISCOVERY_SPIx_MOSI_PIN
    GPIO_PIN_4          /* PD.04 */
00269
00270 /* Maximum Timeout values for flags waiting
loops. These timeouts are not based
00271    on accurate values, they just guarantee t
hat the application will not remain
00272    stuck if the SPI communication is corrupt
ed.
00273    You may modify these timeout values depen
ding on CPU frequency and application
00274    conditions (interrupts routines ...). */
00275 #define SPIx_TIMEOUT_MAX
    ((uint32_t)0x1000)
00276 /* Read/Write command */
00277 #define READWRITE_CMD
    ((uint8_t)0x80)
00278 /* Multiple byte read/write command */
00279 #define MULTIPLEBYTE_CMD
    ((uint8_t)0x40)
00280 /* Dummy Byte Send by the SPI Master device
in order to generate the Clock to the Slave device
*/
00281 #define DUMMY_BYTE
    ((uint8_t)0x00)
00282
00283 #endif /* HAL_SPI_MODULE_ENABLED */
00284

```

```

00285 #if defined(HAL_I2C_MODULE_ENABLED)
00286 /*##### I2C1 #####
#####*/
00287 /* User can use this section to tailor I2C1
instance used and associated
00288     resources */
00289 /* Definition for I2C1 Pins */
00290 #define DISCOVERY_I2C1_SCL_GPIO_PORT
        GPIOB
00291 #define DISCOVERY_I2C1_SDA_GPIO_PORT
        GPIOB
00292 #if defined (USE_STM32L476G_DISCO_REVC) || d
efined (USE_STM32L476G_DISCO_REVB)
00293 #define DISCOVERY_I2C1_SCL_PIN
        GPIO_PIN_6
00294 #define DISCOVERY_I2C1_SDA_PIN
        GPIO_PIN_7
00295 #elif defined (USE_STM32L476G_DISCO_REVA)
00296 #define DISCOVERY_I2C1_SCL_PIN
        GPIO_PIN_8
00297 #define DISCOVERY_I2C1_SDA_PIN
        GPIO_PIN_9
00298 #endif
00299 #define DISCOVERY_I2C1_SCL_SDA_AF
        GPIO_AF4_I2C1
00300
00301 /* Definition for I2C1 clock resources */
00302 #define DISCOVERY_I2C1
        I2C1
00303 #define DISCOVERY_I2C1_CLK_ENABLE()
        __HAL_RCC_I2C1_CLK_ENABLE()
00304 #define DISCOVERY_I2C1_CLK_DISABLE()
        __HAL_RCC_I2C1_CLK_DISABLE()
00305 #define DISCOVERY_I2C1_SDA_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00306 #define DISCOVERY_I2C1_SCL_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()

```

```

00307 #define DISCOVERY_I2C1_SDA_GPIO_CLK_DISABLE(
)    __HAL_RCC_GPIOB_CLK_DISABLE()
00308 #define DISCOVERY_I2C1_SCL_GPIO_CLK_DISABLE(
)    __HAL_RCC_GPIOB_CLK_DISABLE()
00309 #define DISCOVERY_I2C1_FORCE_RESET()
    __HAL_RCC_I2C1_FORCE_RESET()
00310 #define DISCOVERY_I2C1_RELEASE_RESET()
    __HAL_RCC_I2C1_RELEASE_RESET()
00311
00312 /* Definition for I2C1's NVIC */
00313 #define DISCOVERY_I2C1_EV_IRQn
    I2C1_EV_IRQn
00314 #define DISCOVERY_I2C1_EV_IRQHandler
    I2C1_EV_IRQHandler
00315 #define DISCOVERY_I2C1_ER_IRQn
    I2C1_ER_IRQn
00316 #define DISCOVERY_I2C1_ER_IRQHandler
    I2C1_ER_IRQHandler
00317
00318 /* I2C TIMING Register define when I2C clock
    source is SYSCLK */
00319 /* I2C TIMING is calculated in case of the I
    2C Clock source is the SYSCLK = 80 MHz */
00320 /* Set 0x90112626 value to reach 100 KHz spe
    ed (Rise time = 640ns, Fall time = 20ns) */
00321 #ifndef DISCOVERY_I2C1_TIMING
00322     #define DISCOVERY_I2C1_TIMING
        0x90112626
00323 #endif /* DISCOVERY_I2C1_TIMING */
00324
00325 /* I2C clock speed configuration (in Hz)
00326     WARNING:
00327     Make sure that this define is not already
    declared in other files (ie.
00328     stm324xg_discovery.h file). It can be use
    d in parallel by other modules. */
00329 #ifndef BSP_I2C_SPEED

```

```

00330  #define BSP_I2C_SPEED
        100000
00331  #endif /* BSP_I2C_SPEED */
00332
00333
00334  /* Audio codec I2C address */
00335  #define AUDIO_I2C_ADDRESS
        ((uint16_t) 0x94)
00336
00337  /* Maximum Timeout values for flags waiting
loops. These timeouts are not based
00338      on accurate values, they just guarantee t
hat the application will not remain
00339      stuck if the I2C communication is corrupt
ed.
00340      You may modify these timeout values depen
ding on CPU frequency and application
00341      conditions (interrupts routines ...). */

00342  #define DISCOVERY_I2C1_TIMEOUT_MAX
        3000
00343
00344
00345  /*##### I2C2 #####
#####*/
00346  /* User can use this section to tailor I2C2
instance used and associated
00347      resources */
00348  /* Definition for I2C2 Pins */
00349  #define DISCOVERY_I2C2_SCL_PIN
        GPIO_PIN_10
00350  #define DISCOVERY_I2C2_SCL_GPIO_PORT
        GPIOB
00351  #define DISCOVERY_I2C2_SDA_PIN
        GPIO_PIN_11
00352  #define DISCOVERY_I2C2_SDA_GPIO_PORT
        GPIOB

```

```

00353 #define DISCOVERY_I2C2_SCL_SDA_AF
        GPIO_AF4_I2C2
00354 /* Definition for I2C2 clock resources */
00355 #define DISCOVERY_I2C2
        I2C2
00356 #define DISCOVERY_I2C2_CLK_ENABLE()
        __HAL_RCC_I2C2_CLK_ENABLE()
00357 #define DISCOVERY_I2C2_CLK_DISABLE()
        __HAL_RCC_I2C2_CLK_DISABLE()
00358 #define DISCOVERY_I2C2_SDA_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00359 #define DISCOVERY_I2C2_SCL_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00360 #define DISCOVERY_I2C2_SDA_GPIO_CLK_DISABLE(
)    __HAL_RCC_GPIOB_CLK_DISABLE()
00361 #define DISCOVERY_I2C2_SCL_GPIO_CLK_DISABLE(
)    __HAL_RCC_GPIOB_CLK_DISABLE()
00362 #define DISCOVERY_I2C2_FORCE_RESET()
        __HAL_RCC_I2C2_FORCE_RESET()
00363 #define DISCOVERY_I2C2_RELEASE_RESET()
        __HAL_RCC_I2C2_RELEASE_RESET()
00364
00365 /* Definition for I2C2's NVIC */
00366 #define DISCOVERY_I2C2_EV_IRQn
        I2C2_EV_IRQn
00367 #define DISCOVERY_I2C2_ER_IRQn
        I2C2_ER_IRQn
00368
00369 /* I2C TIMING Register define when I2C clock
        source is SYSCLK */
00370 /* I2C TIMING is calculated in case of the I
        2C Clock source is the SYSCLK = 80 MHz */
00371 /* Set 0x90112626 value to reach 100 KHz spe
        ed (Rise time = 640ns, Fall time = 20ns) */
00372 #ifndef DISCOVERY_I2C2_TIMING
00373     #define DISCOVERY_I2C2_TIMING
        0x90112626

```

```

00374 #endif /* DISCOVERY_I2C2_TIMING */
00375
00376 /* I2C clock speed configuration (in Hz)
00377     WARNING:
00378     Make sure that this define is not already
00379     declared in other files (ie.
00380     stm324xg_discovery.h file). It can be use
00381     d in parallel by other modules. */
00382 #ifndef BSP_I2C_SPEED
00383 #define BSP_I2C_SPEED
00384     100000
00385 #endif /* BSP_I2C_SPEED */
00386
00387 #define IDD_I2C_ADDRESS
00388     ((uint16_t) 0x84)
00389
00390 /* Maximum Timeout values for flags waiting
00391     loops. These timeouts are not based
00392     on accurate values, they just guarantee t
00393     hat the application will not remain
00394     stuck if the I2C communication is corrupt
00395     ed.
00396     You may modify these timeout values depen
00397     ding on CPU frequency and application
00398     conditions (interrupts routines ...). */
00399
00400 #define DISCOVERY_I2C2_TIMEOUT_MAX
00401     3000
00402 #endif /* HAL_I2C_MODULE_ENABLED */
00403
00404 /*##### Accelerometer #####
00405     #####*/
00406 /**
00407     * @brief Accelerometer Chip Select macro
00408     definition
00409     */
00410 #define ACCELERO_CS_LOW()

```



```

        HAL_GPIO_WritePin(ACCELERO_CS_GPIO_PORT, ACCEL
ERO_CS_PIN, GPIO_PIN_RESET)
00399 #define ACCELERO_CS_HIGH()
        HAL_GPIO_WritePin(ACCELERO_CS_GPIO_PORT, ACCEL
ERO_CS_PIN, GPIO_PIN_SET)
00400
00401 /**
00402  * @brief Accelerometer SPI Interface pins
00403  */
00404 #define ACCELERO_CS_GPIO_PORT
        GPIOE /* GPIOE */
00405 #define ACCELERO_CS_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOE_CLK_ENABLE()
00406 #define ACCELERO_CS_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOE_CLK_DISABLE()
00407 #define ACCELERO_CS_PIN
        GPIO_PIN_0 /* PE.00 */
00408
00409 /**
00410  * @brief Accelerometer Interrupt pins
00411  */
00412 #define ACCELERO_XLINT_GPIO_PORT
        GPIOE /* GPIOE */
00413 #define ACCELERO_XLINT_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOE_CLK_ENABLE()
00414 #define ACCELERO_XLINT_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOE_CLK_DISABLE()
00415 #define ACCELERO_XLINT_PIN
        GPIO_PIN_1 /* PE.01 */
00416 #define ACCELERO_XLINT_EXTI_IRQn
        EXTI1_IRQn
00417
00418 /*##### Magnetometer #####
#####*/
00419 /**
00420  * @brief Magnetometer Chip Select macro d
efinition

```

```

00421     */
00422 #define MAGNETO_CS_LOW()
        HAL_GPIO_WritePin(MAGNETO_CS_GPIO_PORT, MAGNET
0_CS_PIN, GPIO_PIN_RESET)
00423 #define MAGNETO_CS_HIGH()
        HAL_GPIO_WritePin(MAGNETO_CS_GPIO_PORT, MAGNET
0_CS_PIN, GPIO_PIN_SET)
00424
00425 /**
00426  * @brief Magnetometer SPI Interface pins
00427  */
00428 #define MAGNETO_CS_GPIO_PORT
        GPIOC                                /* GPIOC */
00429 #define MAGNETO_CS_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOC_CLK_ENABLE()
00430 #define MAGNETO_CS_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOC_CLK_DISABLE()
00431 #define MAGNETO_CS_PIN
        GPIO_PIN_0                          /* PC.00 */
00432
00433
00434 /**
00435  * @brief Magnetometer Interrupt pins
00436  */
00437 #define MAGNETO_INT_GPIO_PORT
        GPIOC                                /* GPIOC */
00438 #define MAGNETO_INT_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOC_CLK_ENABLE()
00439 #define MAGNETO_INT_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOC_CLK_DISABLE()
00440 #define MAGNETO_INT1_PIN
        GPIO_PIN_1                          /* PC.01 */
00441 #define MAGNETO_INT1_EXTI_IRQn
        EXTI1_IRQn
00442
00443 #define MAGNETO_DRDY_GPIO_PORT
        GPIOC                                /* GPIOC */

```

```

00444 #define MAGNETO_DRDY_GPIO_CLK_ENABLE()
    __HAL_RCC_GPIOC_CLK_ENABLE()
00445 #define MAGNETO_DRDY_GPIO_CLK_DISABLE()
    __HAL_RCC_GPIOC_CLK_DISABLE()
00446 #define MAGNETO_DRDY_PIN
    GPIO_PIN_2                                /* PC.01 */
00447
00448
00449 /*##### Audio Codec #####
#####*/
00450 /**
00451  * @brief Audio codec chip reset definition

00452  */
00453 /* Audio codec power on/off macro definition
    */
00454 #define CODEC_AUDIO_POWER_OFF()          HAL_GPI
O_WritePin(AUDIO_RESET_GPIO, AUDIO_RESET_PIN, GPIO
_PIN_RESET)
00455 #define CODEC_AUDIO_POWER_ON()          HAL_GPI
O_WritePin(AUDIO_RESET_GPIO, AUDIO_RESET_PIN, GPIO
_PIN_SET)
00456
00457 /* Audio Reset Pin definition */
00458 #define AUDIO_RESET_GPIO_CLK_ENABLE()
    __HAL_RCC_GPIOE_CLK_ENABLE()
00459 #define AUDIO_RESET_GPIO_CLK_DISABLE()
    __HAL_RCC_GPIOE_CLK_DISABLE()
00460 #define AUDIO_RESET_PIN
    GPIO_PIN_3
00461 #define AUDIO_RESET_GPIO
    GPIOE
00462
00463 /*##### Gyroscope #####
#####*/
00464 /**
00465  * @brief Gyroscope Chip Select macro defi

```

```

nitition
00466     */
00467 #define GYRO_CS_LOW()
        HAL_GPIO_WritePin(GYRO_CS_GPIO_PORT, GYRO_CS_P
IN, GPIO_PIN_RESET)
00468 #define GYRO_CS_HIGH()
        HAL_GPIO_WritePin(GYRO_CS_GPIO_PORT, GYRO_CS_P
IN, GPIO_PIN_SET)
00469
00470 /**
00471  * @brief Gyroscope SPI Interface pins
00472  */
00473 #define GYRO_CS_GPIO_PORT
        GPIOD                                /* GPIOD */
00474 #define GYRO_CS_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOD_CLK_ENABLE()
00475 #define GYRO_CS_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOD_CLK_DISABLE()
00476 #define GYRO_CS_PIN
        GPIO_PIN_7                            /* PD.07 */
00477
00478 /**
00479  * @brief Gyroscope Interrupt pins
00480  */
00481 #if defined (USE_STM32L476G_DISCO_REVC) || d
efined (USE_STM32L476G_DISCO_REVB)
00482 #define GYRO_INT1_GPIO_PORT
        GPIOD                                /* GPIOD */
00483 #define GYRO_INT1_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOD_CLK_ENABLE()
00484 #define GYRO_INT1_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOD_CLK_DISABLE()
00485 #define GYRO_INT1_PIN
        GPIO_PIN_2                            /* PD.02 */
00486 #define GYRO_INT1_EXTI_IRQn
        EXTI2_IRQn
00487 #define GYRO_INT2_GPIO_PORT

```

```

        GPIOB                                /* GPIOB */
00488 #define GYRO_INT2_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00489 #define GYRO_INT2_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOB_CLK_DISABLE()
00490 #define GYRO_INT2_PIN
        GPIO_PIN_8                          /* PB.08 */
00491 #define GYRO_INT2_EXTI_IRQn
        EXTI9_5_IRQn
00492 #elif defined (USE_STM32L476G_DISCO_REVA)
00493 #define GYRO_INT1_GPIO_PORT
        GPIOB                                /* GPIOB */
00494 #define GYRO_INT1_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00495 #define GYRO_INT1_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOB_CLK_DISABLE()
00496 #define GYRO_INT1_PIN
        GPIO_PIN_6                          /* PB.06 */
00497 #define GYRO_INT1_EXTI_IRQn
        EXTI9_5_IRQn
00498 #define GYRO_INT2_GPIO_PORT
        GPIOB                                /* GPIOB */
00499 #define GYRO_INT2_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00500 #define GYRO_INT2_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOB_CLK_DISABLE()
00501 #define GYRO_INT2_PIN
        GPIO_PIN_7                          /* PB.07 */
00502 #define GYRO_INT2_EXTI_IRQn
        EXTI9_5_IRQn
00503 #endif
00504
00505 /*##### Idd #####
#####*/
00506 /**
00507  * @brief Idd current measurement interfac
e pins

```

```

00508     */
00509 #define IDD_INT_GPIO_PORT
        GPIOC                                /* GPIOC */
00510 #define IDD_INT_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOC_CLK_ENABLE()
00511 #define IDD_INT_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOC_CLK_DISABLE()
00512 #define IDD_INT_PIN
        GPIO_PIN_13                          /* PC.13 */
00513 #define IDD_INT_EXTI_IRQn
        EXTI15_10_IRQn
00514
00515 #define IDD_WAKEUP_GPIO_PORT
        GPIOA                                /* GPIOA */
00516 #define IDD_WAKEUP_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOA_CLK_ENABLE()
00517 #define IDD_WAKEUP_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOA_CLK_DISABLE()
00518 #define IDD_WAKEUP_PIN
        GPIO_PIN_4                          /* PA.04 */
00519
00520 /**
00521     * @}
00522     */
00523
00524 /**
00525     * @}
00526     */
00527
00528
00529 /** @defgroup STM32L476G_DISCOVERY_Exported_
Functions Exported Functions
00530     * @{
00531     */
00532 uint32_t                                     BSP_GetVersion(void)
;
00533 SupplyMode_TypeDef                         BSP_SupplyModeDetect

```

```

ion(void);
00534 void
eDef Led);
00535 void
ypeDef Led);
00536 void
ef Led);
00537 void
Def Led);
00538 void
ypeDef Led);
00539 uint8_t
_TypeDef Joy_Mode);
00540 void
;
00541 JOYState_TypeDef
);
00542
00543 /**
00544  * @}
00545  */
00546
00547 /**
00548  * @}
00549  */
00550
00551 /**
00552  * @}
00553  */
00554
00555 /**
00556  * @}
00557  */
00558
00559 #ifdef __cplusplus
00560 }
00561 #endif

```

```

BSP_LED_Init(Led_Typ
BSP_LED_DeInit(Led_T
BSP_LED_On(Led_TypeD
BSP_LED_Off(Led_Type
BSP_LED_Toggle(Led_T
BSP_JOY_Init(JOYMode
BSP_JOY_DeInit(void)
BSP_JOY_GetState(void

```

```
00562
00563 #endif /* __STM32L476G_DISCOVERY_H */
00564
00565 /***** (C) COPYRIGHT STMicroelectronics *****/
00566 *****END OF FILE*****/
```

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery.c

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery.c
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file provides a set of fir
00008      *            mware functions to manage Leds,
00009      *            push-button and joystick of STM
00010      *            32L476G-Discovery board (MB1184)
00011      *            ****
00012      * @attention
00013      *
00014      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00015      * icroelectronics</center></h2>
00016      *
00017      * Redistribution and use in source and bin
00018      * ary forms, with or without modification,
00019      * are permitted provided that the followin
00020      * g conditions are met:
```

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

```

00035      *
00036      ****
00037      */
00038
00039 /* Includes -----
----- */
00040 #include "stm32l476g_discovery.h"
00041
00042 /** @addtogroup BSP
00043     * @{
00044     */
00045
00046 /** @defgroup STM32L476G_DISCOVERY STM32L476
G-DISCOVERY
00047     * @{
00048     */
00049
00050 /** @defgroup STM32L476G_DISCOVERY_Common ST
M32L476G-DISCOVERY Common
00051     * @{
00052     */
00053
00054 /** @defgroup STM32L476G_DISCOVERY_Private_T
ypesDefinitions Private Types Definitions
00055     * @brief This file provides firmware funct
ions to manage Leds, push-buttons,
00056     *          COM ports, SD card on SPI and tem
perature sensor (TS751) available on
00057     *          STM32L476G-DISCOVERY discoveryuat
ion board from STMicroelectronics.
00058     * @{
00059     */
00060
00061 /**
00062     * @}
00063     */

```

```

00064
00065 /** @defgroup STM32L476G_DISCOVERY_Private_D
efines Private Defines
00066     * @{
00067     */
00068
00069 /**
00070  * @brief STM32L476G DISCOVERY BSP Driver ve
rsion number $VERSION$
00071     */
00072 #define __STM32L476G_DISCOVERY_BSP_VERSION_M
AIN      (0x00) /*!< [31:24] main version */
00073 #define __STM32L476G_DISCOVERY_BSP_VERSION_S
UB1      (0x00) /*!< [23:16] sub1 version */
00074 #define __STM32L476G_DISCOVERY_BSP_VERSION_S
UB2      (0x00) /*!< [15:8] sub2 version */
00075 #define __STM32L476G_DISCOVERY_BSP_VERSION_R
C        (0x01) /*!< [7:0] release candidate */
00076 #define __STM32L476G_DISCOVERY_BSP_VERSION
((__STM32L476G_DISCOVERY_BSP_VERSION_MAI
N << 24)\
00077
| (__STM32L476G_DISCOVERY_BSP_VERSION_SUB
1 << 16)\
00078
| (__STM32L476G_DISCOVERY_BSP_VERSION_SUB
2 << 8 )\
00079
| (__STM32L476G_DISCOVERY_BSP_VERSION_RC))

00080 /**
00081     * @}
00082     */
00083
00084
00085 /** @defgroup STM32L476G_DISCOVERY_Private_M
acros Private Macros

```

```

00086      * @{
00087      */
00088
00089 /**
00090      * @}
00091      */
00092
00093
00094 /** @defgroup STM32L476G_DISCOVERY_Exported_
Variables Exported Variables
00095      * @{
00096      */
00097
00098 /**
00099      * @brief LED variables
00100      */
00101 #if defined (USE_STM32L476G_DISCO_REVC) || d
efined (USE_STM32L476G_DISCO_REVB)
00102 GPIO_TypeDef*   LED_PORT[LEDn] = {LED4_GPIO
_PORT,
00103                                     LED5_GPIO_
_PORT};
00104
00105 const uint16_t   LED_PIN[LEDn] = {LED4_PIN,
00106                                     LED5_PIN};
00107 #elif defined (USE_STM32L476G_DISCO_REVA)
00108 GPIO_TypeDef*   LED_PORT[LEDn] = {LED3_GPIO
_PORT,
00109                                     LED4_GPIO_
_PORT};
00110
00111 const uint16_t   LED_PIN[LEDn] = {LED3_PIN,
00112                                     LED4_PIN};
00113 #endif
00114
00115
00116 /**

```

```

00117  * @brief JOYSTICK variables
00118  */
00119  GPIO_TypeDef* JOY_PORT[JOYn] = {SEL_JOY_GP
IO_PORT,
00120                                  DOWN_JOY_G
PIO_PORT,
00121                                  LEFT_JOY_G
PIO_PORT,
00122                                  RIGHT_JOY_
GPIO_PORT,
00123                                  UP_JOY_GPI
O_PORT};
00124
00125  const uint16_t JOY_PIN[JOYn] = {SEL_JOY_PIN
,
00126                                  LEFT_JOY_P
IN,
00127                                  RIGHT_JOY_
PIN,
00128                                  DOWN_JOY_P
IN,
00129                                  UP_JOY_PIN
};
00130
00131  const uint8_t JOY_IRQn[JOYn] = {SEL_JOY_EX
TI_IRQn,
00132                                  LEFT_JOY_E
XTI_IRQn,
00133                                  RIGHT_JOY_
EXTI_IRQn,
00134                                  DOWN_JOY_E
XTI_IRQn,
00135                                  UP_JOY_EXT
I_IRQn};
00136
00137  /**
00138  * @brief BUS variables

```

```

00139  */
00140 #if defined(HAL_I2C_MODULE_ENABLED)
00141 uint32_t I2c1Timeout = DISCOVERY_I2C2_TIMEOUT_MAX; /*<! Value of Timeout when I2C1 communication fails */
00142 uint32_t I2c2Timeout = DISCOVERY_I2C2_TIMEOUT_MAX; /*<! Value of Timeout when I2C2 communication fails */
00143 static I2C_HandleTypeDef I2c1Handle;
00144 static I2C_HandleTypeDef I2c2Handle;
00145 #endif /* HAL_I2C_MODULE_ENABLED */
00146
00147 #if defined(HAL_SPI_MODULE_ENABLED)
00148
00149 /* LL definition */
00150 #define __SPI_DIRECTION_2LINES(__HANDLE__)
    do{\
00151     CLEAR_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RXONLY | SPI_CR1_BIDIMODE | SPI_CR1_BIDIOE);\
00152     }while(0);
00153
00154 #define __SPI_DIRECTION_2LINES_RXONLY(__HANDLE__) do{\
00155     CLEAR_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RXONLY | SPI_CR1_BIDIMODE | SPI_CR1_BIDIOE);\
00156     SET_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RXONLY);\
00157     }while(0);
00158
00159 #define __SPI_DIRECTION_1LINE_TX(__HANDLE__)
    do{\
00160

```

```

    CLEAR_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RX
ONLY | SPI_CR1_BIDIMODE | SPI_CR1_BIDIOE);\
00161
    SET_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_BIDI
MODE | SPI_CR1_BIDIOE);\
00162
    }while(0);
00163
00164 #define __SPI_DIRECTION_1LINE_RX(__HANDLE__)
do {\
00165
    CLEAR_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_RX
ONLY | SPI_CR1_BIDIMODE | SPI_CR1_BIDIOE);\
00166
    SET_BIT((__HANDLE__)->Instance->CR1, SPI_CR1_BIDI
MODE);\
00167
    } while(0);
00168
00169
00170 uint32_t SpixTimeout = SPIx_TIMEOUT_MAX;
    /*<! Value of Timeout when SPI communicati
on fails */
00171 static SPI_HandleTypeDef SpiHandle;
00172 #endif /* HAL_SPI_MODULE_ENABLED */
00173
00174 /**
00175  * @}
00176  */
00177
00178 /** @defgroup STM32L476G_DISCOVERY_Private_F
unctionPrototypes Private Functions
00179  * @{
00180  */
00181 /** ***** Bus functions
***** */
00182 /** I2C2 bus function */

```



```

00183 #if defined(HAL_I2C_MODULE_ENABLED)
00184 static void I2C2_Init(void);
00185 static void I2C2_MspInit(I2C_H
andleTypeDef *hi2c);
00186 static void I2C2_DeInit(void);
00187 static void I2C2_MspDeInit(I2C
_HandleTypeDef *hi2c);
00188 static void I2C2_WriteData(uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t Value);
00189 static HAL_StatusTypeDef I2C2_WriteBuffer(uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length);
00190 static uint8_t I2C2_ReadData(uint16_t Addr, uint16_t Reg, uint16_t RegSize);
00191 static HAL_StatusTypeDef I2C2_ReadBuffer(uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length);
00192 static void I2C2_Error (void);
00193
00194 static void I2C1_Init(void);
00195 static void I2C1_MspInit(I2C_H
andleTypeDef *hi2c);
00196 static void I2C1_DeInit(void);
00197 static void I2C1_MspDeInit(I2C
_HandleTypeDef *hi2c);
00198 static HAL_StatusTypeDef I2C1_WriteBuffer(uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length);
00199 static HAL_StatusTypeDef I2C1_ReadBuffer(uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length);
00200 static void I2C1_Error (void);
00201 #endif/* HAL_I2C_MODULE_ENABLED */
00202
00203 /* SPIx bus function */
00204 #if defined(HAL_SPI_MODULE_ENABLED)

```

```

00205 static void SPIx_Init(void);
00206 static void SPIx_MspInit(SPI_H
andleTypeDef *hspl);
00207 static void SPIx_DeInit(void);
00208 static void SPIx_MspDeInit(void
);
00209 static uint8_t SPIx_WriteRead(uint8_t Byte);
00210 static void SPIx_Write(uint8_t
byte);
00211 static uint8_t SPIx_Read(void);
00212 #endif
00213
00214 /***** Link functions
***** /
00215 #if defined(HAL_I2C_MODULE_ENABLED)
00216 /* Link functions for EEPROM peripheral over
I2C */
00217 void EEPROM_I2C_IO_Init(
void);
00218 HAL_StatusTypeDef EEPROM_I2C_IO_Writ
eData(uint16_t DevAddress, uint16_t MemAddress, ui
nt8_t* pBuffer, uint32_t BufferSize);
00219 HAL_StatusTypeDef EEPROM_I2C_IO_Read
Data(uint16_t DevAddress, uint16_t MemAddress, uin
t8_t* pBuffer, uint32_t BufferSize);
00220 HAL_StatusTypeDef EEPROM_I2C_IO_IsDe
viceReady(uint16_t DevAddress, uint32_t Trials);
00221
00222 /* Link functions for Audio Codec peripheral
*/
00223 void AUDIO_IO_Init(void
);
00224 void AUDIO_IO_DeInit(vo
id);
00225 void AUDIO_IO_Write(uint8_t Addr, uint8_t Reg, uint8_t Value);

```

```

00226 uint8_t          AUDIO_IO_Read(uint
8_t Addr, uint8_t Reg);
00227 void              AUDIO_IO_Delay(uint
t32_t delay);
00228 #endif/* HAL_I2C_MODULE_ENABLED */
00229
00230 #if defined(HAL_SPI_MODULE_ENABLED)
00231 /* Link function for COMPASS / ACCELERO peri
pheral */
00232 void                ACCELERO_IO_Init(v
oid);
00233 void                ACCELERO_IO_DeInit(
void);
00234 void                ACCELERO_IO_ITConf
ig(void);
00235 void                ACCELERO_IO_Write(
uint8_t RegisterAddr, uint8_t Value);
00236 uint8_t            ACCELERO_IO_Read(u
int8_t RegisterAddr);
00237
00238 void                MAGNETO_IO_Init(vo
id);
00239 void                MAGNETO_IO_DeInit(
void);
00240 void                MAGNETO_IO_ITConfig
(void);
00241 void                MAGNETO_IO_Write(u
int8_t RegisterAddr, uint8_t Value);
00242 uint8_t            MAGNETO_IO_Read(ui
nt8_t RegisterAddr);
00243
00244
00245 /* Link functions for GYRO peripheral */
00246 void                GYRO_IO_Init(void)
;
00247 void                GYRO_IO_DeInit(void
);

```

```

00248 void                                GYRO_IO_Write(uint
8_t* pBuffer, uint8_t WriteAddr, uint16_t NumByteT
oWrite);
00249 void                                GYRO_IO_Read(uint8
_t* pBuffer, uint8_t ReadAddr, uint16_t NumByteToR
ead);
00250
00251 #endif
00252
00253 #if defined(HAL_I2C_MODULE_ENABLED)
00254 /* Link functions IOExpander */
00255 void                                IOE_Init(void);
00256 void                                IOE_ITConfig(void)
;
00257 void                                IOE_Delay(uint32_t
Delay);
00258 void                                IOE_Write(uint8_t
Addr, uint8_t Reg, uint8_t Value);
00259 uint8_t                            IOE_Read(uint8_t A
ddr, uint8_t Reg);
00260 uint16_t                            IOE_ReadMultiple(u
int8_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_
t Length);
00261
00262 /* Link functions for IDD measurment */
00263 void                                MFX_IO_Init(void);
00264 void                                MFX_IO_DeInit(void
);
00265 void                                MFX_IO_ITConfig (v
oid);
00266 void                                MFX_IO_EnableWakeu
pPin(void);
00267 void                                MFX_IO_Wakeup(void
);
00268 void                                MFX_IO_Delay(uint3
2_t delay);
00269 void                                MFX_IO_Write(uint1

```

```

6_t addr, uint8_t reg, uint8_t value);
00270 uint8_t          MFX_IO_Read(uint16
_t addr, uint8_t reg);
00271 void          MFX_IO_WriteMultiple(uint16_t Addr, uint8_t Reg, uint8_t *Buffer, ui
nt16_t Length);
00272 uint16_t          MFX_IO_ReadMultiple
(uint16_t addr, uint8_t reg, uint8_t *buffer, uint
16_t length);
00273 #endif/* HAL_I2C_MODULE_ENABLED */
00274 /**
00275  * @}
00276  */
00277
00278 /** @defgroup STM32L476G_DISCOVERY_Exported_
Functions Exported Functions
00279  * @{
00280  */
00281
00282 /**
00283  * @brief This method returns the STM32L47
6 DISCOVERY BSP Driver revision
00284  * @retval version : 0xXYZR (8bits for each
decimal, R for RC)
00285  */
00286 uint32_t BSP_GetVersion(void)
00287 {
00288     return __STM32L476G_DISCOVERY_BSP_VERSION;
00289 }
00290
00291 /**
00292  * @brief This method returns the STM32L47
6 DISCOVERY supply mode
00293  * @retval Code of current supply mode
00294  *          This code can be one of followi
ng:
00295  *          @arg SUPPLY_MODE_EXTERNAL

```

```

00296      *          @arg SUPPLY_MODE_BATTERY
00297      */
00298  SupplyMode_TypeDef BSP_SupplyModeDetection(v
oid)
00299  {
00300      SupplyMode_TypeDef supplymode = SUPPLY_MOD
E_ERROR;
00301      GPIO_InitTypeDef GPIO_InitStruct;
00302
00303      BATTERY_DETECTION_GPIO_CLK_ENABLE();
00304
00305      /* COMP GPIO pin configuration */
00306      GPIO_InitStruct.Pin = BATTERY_DETECTION_PIN
;
00307      GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
00308      GPIO_InitStruct.Pull = GPIO_NOPULL;
00309      GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
00310      HAL_GPIO_Init(BATTERY_DETECTION_GPIO_PORT,
&GPIO_InitStruct);
00311
00312      HAL_Delay(400);
00313      if(HAL_GPIO_ReadPin(BATTERY_DETECTION_GPIO
_PORT, GPIO_InitStruct.Pin) != GPIO_PIN_RESET)
00314      {
00315          supplymode = SUPPLY_MODE_EXTERNAL;
00316      }
00317      else
00318      {
00319          supplymode = SUPPLY_MODE_BATTERY;
00320      }
00321
00322      HAL_GPIO_DeInit(BATTERY_DETECTION_GPIO_PORT
, GPIO_InitStruct.Pin);
00323
00324      return supplymode;
00325  }
00326

```

```

00327 #if defined (USE_STM32L476G_DISCO_REVC) || d
efined (USE_STM32L476G_DISCO_REVB)
00328 /**
00329  * @brief Configures LED GPIOs.
00330  * @param Led: Specifies the Led to be con
figured.
00331  * This parameter can be one of following
parameters:
00332  * @arg LED4
00333  * @arg LED5
00334  * @retval None
00335  */
00336 #elif defined (USE_STM32L476G_DISCO_REVA)
00337 /**
00338  * @brief Configures LED GPIOs.
00339  * @param Led: Specifies the Led to be con
figured.
00340  * This parameter can be one of following
parameters:
00341  * @arg LED3
00342  * @arg LED4
00343  * @retval None
00344  */
00345 #endif
00346 void BSP_LED_Init(Led_TypeDef Led)
00347 {
00348     GPIO_InitTypeDef GPIO_InitStructure;
00349
00350     /* Enable the GPIO_LED clock */
00351     LEDx_GPIO_CLK_ENABLE(Led);
00352
00353     /* Configure the GPIO_LED pin */
00354     GPIO_InitStructure.Pin = LED_PIN[Led];
00355     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT
_PP;
00356     GPIO_InitStructure.Pull = GPIO_NOPULL;
00357     GPIO_InitStructure.Speed = GPIO_SPEED_HIGH

```

```

;
00358
00359     HAL_GPIO_Init(LED_PORT[Led], &GPIO_InitStr
ucture);
00360
00361     HAL_GPIO_WritePin(LED_PORT[Led], GPIO_Init
Structure.Pin, GPIO_PIN_RESET);
00362 }
00363
00364 #if defined (USE_STM32L476G_DISCO_REVC) || d
efined (USE_STM32L476G_DISCO_REVB)
00365 /**
00366  * @brief Unconfigures LED GPIOs.
00367  * @param Led: Specifies the Led to be unc
onfigured.
00368  *      This parameter can be one of following
parameters:
00369  *          @arg LED4
00370  *          @arg LED5
00371  * @retval None
00372  */
00373 #elif defined (USE_STM32L476G_DISCO_REVA)
00374 /**
00375  * @brief Unconfigures LED GPIOs.
00376  * @param Led: Specifies the Led to be unc
onfigured.
00377  *      This parameter can be one of following
parameters:
00378  *          @arg LED3
00379  *          @arg LED4
00380  * @retval None
00381  */
00382 #endif
00383 void BSP_LED_DeInit(Led_TypeDef Led)
00384 {
00385     /* Enable the GPIO_LED clock */
00386     LEDx_GPIO_CLK_ENABLE(Led);

```



```

00387
00388     HAL_GPIO_DeInit(LED_PORT[Led], LED_PIN[Led
00389 ]);
00390 }
00391 #if defined (USE_STM32L476G_DISCO_REVC) || d
00392 efined (USE_STM32L476G_DISCO_REVB)
00393 /**
00394  * @brief Turns selected LED On.
00395  * @param Led: Specifies the Led to be set
00396  * on.
00397  * This parameter can be one of following
00398  * parameters:
00399  * @arg LED4
00400  * @arg LED5
00401  * @retval None
00402  */
00403 #elif defined (USE_STM32L476G_DISCO_REVA)
00404 /**
00405  * @brief Turns selected LED On.
00406  * @param Led: Specifies the Led to be set
00407  * on.
00408  * This parameter can be one of following
00409  * parameters:
00410  * @arg LED3
00411  * @arg LED4
00412  * @retval None
00413  */
00414 #endif
00415 void BSP_LED_On(Led_TypeDef Led)
00416 {
00417     HAL_GPIO_WritePin(LED_PORT[Led], LED_PIN[L
00418 ed], GPIO_PIN_SET);
00419 }
00420 }
00421 #if defined (USE_STM32L476G_DISCO_REVC) || d
00422 efined (USE_STM32L476G_DISCO_REVB)

```

```

00416 /**
00417  * @brief Turns selected LED Off.
00418  * @param Led: Specifies the Led to be set
00419  *           off.
00420  *           This parameter can be one of following
00421  *           parameters:
00422  *           @arg LED4
00423  *           @arg LED5
00424  *           @retval None
00425  */
00426 #elif defined (USE_STM32L476G_DISCO_REVA)
00427 /**
00428  * @brief Turns selected LED Off.
00429  * @param Led: Specifies the Led to be set
00430  *           off.
00431  *           This parameter can be one of following
00432  *           parameters:
00433  *           @arg LED3
00434  *           @arg LED4
00435  *           @retval None
00436  */
00437 #endif
00438 void BSP_LED_Off(Led_TypeDef Led)
00439 {
00440     HAL_GPIO_WritePin(LED_PORT[Led], LED_PIN[Led], GPIO_PIN_RESET);
00441 }
00442
00443 #if defined (USE_STM32L476G_DISCO_REVC) || defined (USE_STM32L476G_DISCO_REVB)
00444 /**
00445  * @brief Toggles the selected LED.
00446  * @param Led: Specifies the Led to be toggled.
00447  *           This parameter can be one of following
00448  *           parameters:
00449  *           @arg LED4

```

```

00445      *      @arg LED5
00446      * @retval None
00447      */
00448 #elif defined (USE_STM32L476G_DISCO_REVA)
00449 /**
00450      * @brief Toggles the selected LED.
00451      * @param Led: Specifies the Led to be tog
00452      *      This parameter can be one of following
00453      *      parameters:
00454      *      @arg LED3
00455      *      @arg LED4
00456      * @retval None
00457      */
00458 void BSP_LED_Toggle(Led_TypeDef Led)
00459 {
00460     HAL_GPIO_TogglePin(LED_PORT[Led], LED_PIN[
00461     Led]);
00462 }
00463 /**
00464      * @brief Configures all buttons of the jo
00465      *      ystick in GPIO or EXTI modes.
00466      * @param Joy_Mode: Joystick mode.
00467      *      This parameter can be one of the foll
00468      *      owing values:
00469      *      @arg JOY_MODE_GPIO: Joystick pins w
00470      *      ill be used as simple I/Os
00471      *      @arg JOY_MODE_EXTI: Joystick pins w
00472      *      ill be connected to EXTI line
00473      *      with int
00474      *      errupt generation capability
00475      * @retval HAL_OK: if all initializations a
00476      *      re OK. Other value if error.
00477      */
00478 uint8_t BSP_JOY_Init(JOYMode_TypeDef Joy_Mod

```

```

e)
00473 {
00474     JOYState_TypeDef joykey;
00475     GPIO_InitTypeDef GPIO_InitStructure;
00476
00477     /* Initialized the Joystick. */
00478     for(joykey = JOY_SEL; joykey < (JOY_SEL +
JOYn) ; joykey++)
00479     {
00480         /* Enable the JOY clock */
00481         JOYx_GPIO_CLK_ENABLE(joykey);
00482
00483         GPIO_InitStructure.Pin = JOY_PIN[joykey];
00484         GPIO_InitStructure.Pull = GPIO_PULLDOWN;
00485         GPIO_InitStructure.Speed = GPIO_SPEED_HIGH;
00486
00487         if (Joy_Mode == JOY_MODE_GPIO)
00488         {
00489             /* Configure Joy pin as input */
00490             GPIO_InitStructure.Mode = GPIO_MODE_INPUT
;
00491             HAL_GPIO_Init(JOY_PORT[joykey], &GPIO_
InitStruct);
00492         }
00493         else if (Joy_Mode == JOY_MODE_EXTI)
00494         {
00495             /* Configure Joy pin as input with Ext
ernal interrupt */
00496             GPIO_InitStructure.Mode = GPIO_MODE_IT_FA
LLING;
00497             HAL_GPIO_Init(JOY_PORT[joykey], &GPIO_
InitStruct);
00498
00499             /* Enable and set Joy EXTI Interrupt t
o the lowest priority */
00500             HAL_NVIC_SetPriority((IRQn_Type)(JOY_I
RQn[joykey]), 0x0F, 0x00);

```

```

00501         HAL_NVIC_EnableIRQ((IRQn_Type)(JOY_IRQn
[joykey]));
00502     }
00503 }
00504
00505     return HAL_OK;
00506 }
00507
00508 /**
00509  * @brief Unconfigures all GPIOs used as bu
ttons of the joystick.
00510  * @retval None.
00511  */
00512 void BSP_JOY_DeInit(void)
00513 {
00514     JOYState_TypeDef joykey;
00515
00516     /* Initialized the Joystick. */
00517     for(joykey = JOY_SEL; joykey < (JOY_SEL +
JOYn) ; joykey++)
00518     {
00519         /* Enable the JOY clock */
00520         JOYx_GPIO_CLK_ENABLE(joykey);
00521
00522         HAL_GPIO_DeInit(JOY_PORT[joykey], JOY_PIN
[joykey]);
00523     }
00524 }
00525
00526 /**
00527  * @brief Returns the current joystick statu
s.
00528  * @retval Code of the joystick key pressed
00529  *         This code can be one of the follo
wing values:
00530  *         @arg JOY_NONE
00531  *         @arg JOY_SEL

```

```
00532 * @arg JOY_DOWN  
00533 * @arg JOY_LEFT  
00534 * @arg JOY_RIGHT  
00535 * @arg JOY_UP  
00536 */  
00537 JOYState_TypeDef BSP_JOY_GetState(void)  
00538 {  
00539     JOYState_TypeDef joykey;  
00540  
00541     for (joykey = JOY_SEL; joykey < (JOY_SEL +  
JOYN); joykey++)  
00542     {  
00543         if (HAL_GPIO_ReadPin(JOY_PORT[joykey], J  
OYPIN[joykey]) == GPIO_PIN_SET)  
00544         {  
00545             /* Return Code Joystick key pressed */  
00546             return joykey;  
00547         }  
00548     }  
00549  
00550     /* No Joystick key pressed */  
00551     return JOY_NONE;  
00552 }  
00553  
00554 /**  
00555  * @}  
00556  */  
00557  
00558 /** @defgroup STM32L476G_DISCOVERY_BusOperations_Functio  
ions_Functions Bus Operations Functions  
00559  * @{  
00560  */  
00561  
00562 /*****  
*****  
  
00563 BUS OPERATIONS  
00564 *****/
```

```

*****/
00565 #if defined(HAL_SPI_MODULE_ENABLED)
00566 /***** SPI Routine
S*****/
00567 /**
00568  * @brief SPIx Bus initialization
00569  * @retval None
00570  */
00571 static void SPIx_Init(void)
00572 {
00573     if(HAL_SPI_GetState(&SpiHandle) == HAL_SPI
_STATE_RESET)
00574     {
00575         /* SPI Config */
00576         SpiHandle.Instance = DISCOVERY_SPIx;
00577         /* SPI baudrate is set to 10 MHz (PCLK2/
SPI_BaudRatePrescaler = 80/8 = 10 MHz)
00578         to verify these constraints:
00579         lsm303c SPI interface max baudrate is
10MHz for write/read
00580         PCLK2 frequency is set to 80 MHz
00581         */
00582         SpiHandle.Init.BaudRatePrescaler = SPI_B
AUDRATEPRESCALER_8;
00583         SpiHandle.Init.Direction = SPI_DIRECTION
_2LINES;
00584         SpiHandle.Init.CLKPhase = SPI_PHASE_1EDG
E;
00585         SpiHandle.Init.CLKPolarity = SPI_POLARIT
Y_LOW;
00586         SpiHandle.Init.CRCCalculation = SPI_CRCC
ALCULATION_DISABLE;
00587         SpiHandle.Init.CRCPolynomial = 7;
00588         SpiHandle.Init.DataSize = SPI_DATASIZE_8
BIT;
00589         SpiHandle.Init.FirstBit = SPI_FIRSTBIT_M
SB;

```

```

00590     SpiHandle.Init.NSS = SPI_NSS_SOFT;
00591     SpiHandle.Init.TIMode = SPI_TIMODE_DISABLE;
00592     SpiHandle.Init.Mode = SPI_MODE_MASTER;
00593
00594     SPIx_MspInit(&SpiHandle);
00595     HAL_SPI_Init(&SpiHandle);
00596 }
00597 }
00598
00599 /**
00600  * @brief SPI MSP Init
00601  * @param hspi: SPI handle
00602  * @retval None
00603  */
00604 static void SPIx_MspInit(SPI_HandleTypeDef *
hspi)
00605 {
00606     GPIO_InitTypeDef    GPIO_InitStructure;
00607
00608     /* Enable SPIx clock */
00609     DISCOVERY_SPIx_CLOCK_ENABLE();
00610
00611     /* enable SPIx gpio clock */
00612     DISCOVERY_SPIx_GPIO_CLK_ENABLE();
00613
00614     /* configure SPIx SCK, MOSI and MISO */
00615     GPIO_InitStructure.Pin = (DISCOVERY_SPIx_S
CK_PIN | DISCOVERY_SPIx_MOSI_PIN | DISCOVERY_SPIx_
MISO_PIN);
00616     GPIO_InitStructure.Mode = GPIO_MODE_AF_PP;
00617     GPIO_InitStructure.Pull = GPIO_NOPULL; //
GPIO_PULLDOWN;
00618     GPIO_InitStructure.Speed = GPIO_SPEED_FAST
;
00619     GPIO_InitStructure.Alternate = DISCOVERY_S
PIx_AF;

```



```

00620     HAL_GPIO_Init(DISCOVERY_SPIx_GPIO_PORT, &G
PIO_InitStructure);
00621 }
00622
00623 /**
00624  * @brief SPIx Bus Deinitialization
00625  * @retval None
00626  */
00627 void SPIx_DeInit(void)
00628 {
00629     if(HAL_SPI_GetState(&SpiHandle) != HAL_SPI
_STATE_RESET)
00630     {
00631         /* SPI Deinit */
00632         HAL_SPI_DeInit(&SpiHandle);
00633         SPIx_MspDeInit();
00634     }
00635 }
00636
00637 /**
00638  * @brief SPI MSP DeInit
00639  * @retval None
00640  */
00641 static void SPIx_MspDeInit(void)
00642 {
00643     /* enable SPIx gpio clock */
00644     DISCOVERY_SPIx_GPIO_CLK_ENABLE();
00645
00646     /* Unconfigure SPIx SCK, MOSI and MISO */
00647     HAL_GPIO_DeInit(DISCOVERY_SPIx_GPIO_PORT, (
DISCOVERY_SPIx_SCK_PIN | DISCOVERY_SPIx_MOSI_PIN |
DISCOVERY_SPIx_MISO_PIN));
00648
00649     DISCOVERY_SPIx_GPIO_FORCE_RESET();
00650     DISCOVERY_SPIx_GPIO_RELEASE_RESET();
00651
00652     /* Disable SPIx clock */

```

```

00653     DISCOVERY_SPIx_CLOCK_DISABLE();
00654 }
00655
00656 /**
00657  * @brief Sends a Byte through the SPI int
erface and return the Byte received
00658  *         from the SPI bus.
00659  * @param Byte : Byte send.
00660  * @retval none.
00661  */
00662 static uint8_t SPIx_WriteRead(uint8_t Byte)
00663 {
00664     uint8_t receivedbyte;
00665
00666     /* Enable the SPI */
00667     __HAL_SPI_ENABLE(&SpiHandle);
00668     /* check TXE flag */
00669     while((SpiHandle.Instance->SR & SPI_FLAG_T
XE) != SPI_FLAG_TXE);
00670
00671     /* Write the data */
00672     *((__IO uint8_t*)&SpiHandle.Instance->DR)
= Byte;
00673
00674     while((SpiHandle.Instance->SR & SPI_FLAG_R
XNE) != SPI_FLAG_RXNE);
00675     receivedbyte = *((__IO uint8_t*)&SpiHandle
.Instance->DR);
00676
00677     /* Wait BSY flag */
00678     while((SpiHandle.Instance->SR & SPI_FLAG_F
TLVL) != SPI_FTLVL_EMPTY);
00679     while((SpiHandle.Instance->SR & SPI_FLAG_B
SY) == SPI_FLAG_BSY);
00680
00681     /* disable the SPI */
00682     __HAL_SPI_DISABLE(&SpiHandle);

```

```

00683
00684     return receivedbyte;
00685 }
00686
00687 /**
00688  * @brief Sends a Byte through the SPI int
erface.
00689  * @param Byte : Byte to send.
00690  * @retval none.
00691  */
00692 static void SPIx_Write(uint8_t Byte)
00693 {
00694     /* Enable the SPI */
00695     __HAL_SPI_ENABLE(&SpiHandle);
00696     /* check TXE flag */
00697     while((SpiHandle.Instance->SR & SPI_FLAG_T
XE) != SPI_FLAG_TXE);
00698
00699     /* Write the data */
00700     *((__IO uint8_t*)&SpiHandle.Instance->DR)
= Byte;
00701
00702     /* Wait BSY flag */
00703     while((SpiHandle.Instance->SR & SPI_FLAG_B
SY) == SPI_FLAG_BSY);
00704
00705     /* disable the SPI */
00706     __HAL_SPI_DISABLE(&SpiHandle);
00707 }
00708
00709 #if defined(__ICCARM__)
00710 #pragma optimize=none
00711 #endif
00712 /**
00713  * @brief Receives a Byte from the SPI bus.
00714  * @retval The received byte value

```

```

00715     */
00716 static uint8_t SPIx_Read(void)
00717 {
00718     uint8_t receivedbyte;
00719
00720     __HAL_SPI_ENABLE(&SpiHandle);
00721     __DSB();
00722     __DSB();
00723     __DSB();
00724     __DSB();
00725     __DSB();
00726     __DSB();
00727     __DSB();
00728     __DSB();
00729     __HAL_SPI_DISABLE(&SpiHandle);
00730
00731     while((SpiHandle.Instance->SR & SPI_FLAG_RXNE) != SPI_FLAG_RXNE);
00732     /* read the received data */
00733     receivedbyte = *(__IO uint8_t *)&SpiHandle
        .Instance->DR;
00734
00735     /* Wait for the BSY flag reset */
00736     while((SpiHandle.Instance->SR & SPI_FLAG_BSY) == SPI_FLAG_BSY);
00737
00738
00739     return receivedbyte;
00740 }
00741 #endif /* HAL_SPI_MODULE_ENABLED */
00742
00743
00744 #if defined(HAL_I2C_MODULE_ENABLED)
00745 /** ***** I2C Routine
S***** */
00746 /**
00747     * @brief Discovery I2C1 Bus initialization

```

```

00748     * @retval None
00749     */
00750 static void I2C1_Init(void)
00751 {
00752     if(HAL_I2C_GetState(&I2c1Handle) == HAL_I2
C_STATE_RESET)
00753     {
00754         I2c1Handle.Instance                = DISCO
VERY_I2C1;
00755         I2c1Handle.Init.Timing              = DISCO
VERY_I2C1_TIMING;
00756         I2c1Handle.Init.OwnAddress1         = 0;
00757         I2c1Handle.Init.AddressingMode     = I2C_A
DDRESSINGMODE_7BIT;
00758         I2c1Handle.Init.DualAddressMode    = I2C_D
UALADDRESS_DISABLE;
00759         I2c1Handle.Init.OwnAddress2        = 0;
00760         I2c1Handle.Init.GeneralCallMode    = I2C_G
ENERALCALL_DISABLE;
00761         I2c1Handle.Init.NoStretchMode      = I2C_N
OSTRETCH_DISABLE;
00762
00763         /* Init the I2C */
00764         I2C1_MspInit(&I2c1Handle);
00765         HAL_I2C_Init(&I2c1Handle);
00766     }
00767 }
00768
00769 /**
00770  * @brief Discovery I2C1 MSP Initialization
00771  * @param hi2c: I2C handle
00772  * @retval None
00773  */
00774 static void I2C1_MspInit(I2C_HandleTypeDef *
hi2c)
00775 {
00776     GPIO_InitTypeDef  GPIO_InitStructure;

```

```

00777   RCC_PeriphCLKInitTypeDef  RCC_PeriphCLKIni
tStruct;
00778
00779   /* IOSV bit MUST be set to access GPIO por
t G[2:15] */
00780   __HAL_RCC_PWR_CLK_ENABLE();
00781   HAL_PWREx_EnableVddIO2();
00782
00783   if (hi2c->Instance == DISCOVERY_I2C1)
00784   {
00785       /*##-1- Configure the Discovery I2C cloc
k source. The clock is derived from the SYSClk */
00786       RCC_PeriphCLKInitStruct.PeriphClockSelec
tion = RCC_PERIPHCLK_I2C1;
00787       RCC_PeriphCLKInitStruct.I2c1ClockSelecti
on = RCC_I2C1CLKSOURCE_SYSClk;
00788       HAL_RCCEx_PeriphCLKConfig(&RCC_PeriphCLK
InitStruct);
00789
00790       /*##-2- Configure the GPIOs #####
#####*/
00791       /* Enable GPIO clock */
00792       DISCOVERY_I2C1_SDA_GPIO_CLK_ENABLE();
00793       DISCOVERY_I2C1_SCL_GPIO_CLK_ENABLE();
00794
00795       /* Configure I2C Rx/Tx as alternate func
tion */
00796       GPIO_InitStructure.Pin           = DISCOVERY
_I2C1_SCL_PIN | DISCOVERY_I2C1_SDA_PIN;
00797       GPIO_InitStructure.Mode          = GPIO_MODE
_AF_OD;
00798       GPIO_InitStructure.Pull         = GPIO_PULL
UP;
00799       GPIO_InitStructure.Speed        = GPIO_SPEE
D_HIGH;
00800       GPIO_InitStructure.Alternate = DISCOVERY
_I2C1_SCL_SDA_AF;

```

```

00801     HAL_GPIO_Init(DISCOVERY_I2C1_SCL_GPIO_PORT, &GPIO_InitStructure);
00802
00803     /*##-3- Configure the Discovery I2C1 peripheral #####*/
00804     /* Enable Discovery I2C1 clock */
00805     DISCOVERY_I2C1_CLK_ENABLE();
00806
00807     /* Force and release the I2C Peripheral Clock Reset */
00808     DISCOVERY_I2C1_FORCE_RESET();
00809     DISCOVERY_I2C1_RELEASE_RESET();
00810
00811     /* Enable and set Discovery I2C1 Interrupt to the highest priority */
00812     HAL_NVIC_SetPriority(DISCOVERY_I2C1_EV_IRQn, 0x00, 0);
00813     HAL_NVIC_EnableIRQ(DISCOVERY_I2C1_EV_IRQn);
00814
00815     /* Enable and set Discovery I2C1 Interrupt to the highest priority */
00816     HAL_NVIC_SetPriority(DISCOVERY_I2C1_ER_IRQn, 0x00, 0);
00817     HAL_NVIC_EnableIRQ(DISCOVERY_I2C1_ER_IRQn);
00818 }
00819 }
00820
00821 /**
00822  * @brief Discovery I2C1 Bus Deinitialization
00823  * @retval None
00824  */
00825 static void I2C1_DeInit(void)
00826 {
00827     if(HAL_I2C_GetState(&I2c1Handle) != HAL_I2C_STATE_RESET)

```

```

00828     {
00829         /* Deinit the I2C */
00830         HAL_I2C_DeInit(&I2c1Handle);
00831         I2C1_MspDeInit(&I2c1Handle);
00832     }
00833 }
00834
00835 /**
00836  * @brief Discovery I2C1 MSP Deinitialization
00837  * @param hi2c: I2C handle
00838  * @retval None
00839  */
00840 static void I2C1_MspDeInit(I2C_HandleTypeDef
    *hi2c)
00841 {
00842     if(hi2c->Instance == DISCOVERY_I2C1)
00843     {
00844         /*##-1- Unconfigure the GPIOs #####
00845         #####*/
00846         /* Enable GPIO clock */
00847         DISCOVERY_I2C1_SDA_GPIO_CLK_ENABLE();
00848         DISCOVERY_I2C1_SCL_GPIO_CLK_ENABLE();
00849
00850         /* Deinit Rx/Tx pins */
00851         HAL_GPIO_DeInit(DISCOVERY_I2C1_SCL_GPIO_
00852             PORT, (DISCOVERY_I2C1_SCL_PIN | DISCOVERY_I2C1_SDA
00853                 _PIN));
00854
00855         /*##-2- Unconfigure the Discovery I2C1 p
00856         eripheral #####*/
00857         /* Force & Release the I2C Peripheral Cl
00858         ock Reset */
00859         DISCOVERY_I2C1_FORCE_RESET();
00860         DISCOVERY_I2C1_RELEASE_RESET();
00861
00862         /* Disable Discovery I2C1 clock */

```



```

00858     DISCOVERY_I2C1_CLK_DISABLE();
00859
00860     /* Disable Discovery I2C1 interrupts */
00861     HAL_NVIC_DisableIRQ(DISCOVERY_I2C1_EV_IRQn);
00862     HAL_NVIC_DisableIRQ(DISCOVERY_I2C1_ER_IRQn);
00863
00864     __HAL_RCC_PWR_CLK_ENABLE();
00865     HAL_PWREx_DisableVddIO2();
00866 }
00867 }
00868
00869 /**
00870  * @brief Write a value in a register of the device through BUS.
00871  * @param Addr: Device address on BUS Bus.
00872  * @param Reg: The target register address to write
00873  * @param RegSize: The target register size (can be 8BIT or 16BIT)
00874  * @param pBuffer: The target register value to be written
00875  * @param Length: buffer size to be written
00876  * @retval None
00877  */
00878 static HAL_StatusTypeDef I2C1_WriteBuffer(uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length)
00879 {
00880     HAL_StatusTypeDef status = HAL_OK;
00881
00882     status = HAL_I2C_Mem_Write(&I2C1Handle, Addr, (uint16_t)Reg, RegSize, pBuffer, Length, I2C1TIMEOUT);

```

```

00883
00884 /* Check the communication status */
00885     if(status != HAL_OK)
00886     {
00887         /* Re-Initiaize the BUS */
00888         I2C1_Error();
00889     }
00890     return status;
00891 }
00892
00893 /**
00894  * @brief Reads multiple data on the BUS.
00895  * @param Addr: I2C Address
00896  * @param Reg: Reg Address
00897  * @param RegSize : The target register si
00898  * @param pBuffer: pointer to read data bu
00899  * @param Length: length of the data
00900  * @retval 0 if no problems to read multipl
00901  */
00902 static HAL_StatusTypeDef I2C1_ReadBuffer(uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length)
00903 {
00904     HAL_StatusTypeDef status = HAL_OK;
00905
00906     status = HAL_I2C_Mem_Read(&I2c1Handle, Addr, (uint16_t)Reg, RegSize, pBuffer, Length, I2c1Timeout);
00907
00908     /* Check the communication status */
00909     if(status != HAL_OK)
00910     {
00911         /* Re-Initiaize the BUS */
00912         I2C1_Error();

```

```

00913     }
00914     return status;
00915 }
00916
00917 /**
00918  * @brief Discovery I2C1 error treatment fu
nction
00919  * @retval None
00920  */
00921 static void I2C1_Error (void)
00922 {
00923     /* De-initialize the I2C communication BUS
*/
00924     HAL_I2C_DeInit(&I2c1Handle);
00925
00926     /* Re- Initiaize the I2C communication BUS
*/
00927     I2C1_Init();
00928 }
00929
00930 /**
00931  * @brief Discovery I2C2 Bus initialization
00932  * @retval None
00933  */
00934 static void I2C2_Init(void)
00935 {
00936     if(HAL_I2C_GetState(&I2c2Handle) == HAL_I2
C_STATE_RESET)
00937     {
00938         I2c2Handle.Instance           = DISCO
VERY_I2C2;
00939         I2c2Handle.Init.Timing        = DISCO
VERY_I2C2_TIMING;
00940         I2c2Handle.Init.OwnAddress1   = 0;
00941         I2c2Handle.Init.AddressingMode = I2C_A
DDRESSINGMODE_7BIT;
00942         I2c2Handle.Init.DualAddressMode = I2C_D

```

```

UALADDRESS_DISABLE;
00943     I2c2Handle.Init.OwnAddress2      = 0;
00944     I2c2Handle.Init.GeneralCallMode   = I2C_G
ENERALCALL_DISABLE;
00945     I2c2Handle.Init.NoStretchMode      = I2C_N
OSTRETCH_DISABLE;
00946
00947     /* Init the I2C */
00948     I2C2_MspInit(&I2c2Handle);
00949     HAL_I2C_Init(&I2c2Handle);
00950 }
00951 }
00952
00953 /**
00954  * @brief Discovery I2C2 MSP Initialization
00955  * @param hi2c: I2C2 handle
00956  * @retval None
00957  */
00958 static void I2C2_MspInit(I2C_HandleTypeDef *
hi2c)
00959 {
00960     GPIO_InitTypeDef  GPIO_InitStructure;
00961     RCC_PeriphCLKInitTypeDef  RCC_PeriphCLKIni
tStruct;
00962
00963     if (hi2c->Instance == DISCOVERY_I2C2)
00964     {
00965         /*##-1- Configure the Discovery I2C2 clo
ck source. The clock is derived from the SYSClk ##*/

00966         RCC_PeriphCLKInitStruct.PeriphClockSelec
tion = RCC_PERIPHCLK_I2C2;
00967         RCC_PeriphCLKInitStruct.I2c2ClockSelecti
on = RCC_I2C2CLKSOURCE_SYSClk;
00968         HAL_RCCEx_PeriphCLKConfig(&RCC_PeriphCLK
InitStruct);
00969

```

```

00970      /*##-2- Configure the GPIOs #####
#####*/
00971      /* Enable GPIO clock */
00972      DISCOVERY_I2C2_SDA_GPIO_CLK_ENABLE();
00973      DISCOVERY_I2C2_SCL_GPIO_CLK_ENABLE();
00974
00975      /* Configure I2C Rx/Tx as alternate func
tion */
00976      GPIO_InitStructure.Pin          = DISCOVERY
_I2C2_SCL_PIN | DISCOVERY_I2C2_SDA_PIN;
00977      GPIO_InitStructure.Mode         = GPIO_MODE
_AF_OD;
00978      GPIO_InitStructure.Pull        = GPIO_PULL
UP;
00979      GPIO_InitStructure.Speed        = GPIO_SPEE
D_HIGH;
00980      GPIO_InitStructure.Alternate = DISCOVERY
_I2C2_SCL_SDA_AF;
00981      HAL_GPIO_Init(DISCOVERY_I2C2_SCL_GPIO_PO
RT, &GPIO_InitStructure);
00982
00983      /*##-3- Configure the Discovery I2C2 per
ipheral #####*/
00984      /* Enable Discovery_I2C2 clock */
00985      DISCOVERY_I2C2_CLK_ENABLE();
00986
00987      /* Force and release the I2C Peripheral
Clock Reset */
00988      DISCOVERY_I2C2_FORCE_RESET();
00989      DISCOVERY_I2C2_RELEASE_RESET();
00990
00991      /* Enable and set Discovery I2C2 Interru
pt to the highest priority */
00992      HAL_NVIC_SetPriority(DISCOVERY_I2C2_EV_I
RQn, 0x00, 0);
00993      HAL_NVIC_EnableIRQ(DISCOVERY_I2C2_EV_IRQn
);

```

```

00994
00995     /* Enable and set Discovery I2C2 Interrupt to the highest priority */
00996     HAL_NVIC_SetPriority(DISCOVERY_I2C2_ER_IRQn, 0x00, 0);
00997     HAL_NVIC_EnableIRQ(DISCOVERY_I2C2_ER_IRQn);
00998 }
00999 }
01000
01001 /**
01002  * @brief Discovery I2C2 Bus Deinitialization
01003  * @retval None
01004  */
01005 static void I2C2_DeInit(void)
01006 {
01007     if(HAL_I2C_GetState(&I2c2Handle) != HAL_I2C_STATE_RESET)
01008     {
01009         /* DeInit the I2C */
01010         HAL_I2C_DeInit(&I2c2Handle);
01011         I2C2_MspDeInit(&I2c2Handle);
01012     }
01013 }
01014
01015 /**
01016  * @brief Discovery I2C2 MSP Deinitialization
01017  * @param hi2c: I2C2 handle
01018  * @retval None
01019  */
01020 static void I2C2_MspDeInit(I2C_HandleTypeDef *hi2c)
01021 {
01022     if (hi2c->Instance == DISCOVERY_I2C2)
01023     {

```

```

01024      /*##-1- Unconfigure the GPIOs #####
#####*/
01025      /* Enable GPIO clock */
01026      DISCOVERY_I2C2_SDA_GPIO_CLK_ENABLE();
01027      DISCOVERY_I2C2_SCL_GPIO_CLK_ENABLE();
01028
01029      /* Configure I2C Rx/Tx as alternate func
tion */
01030      HAL_GPIO_DeInit(DISCOVERY_I2C2_SCL_GPIO_
PORT, (DISCOVERY_I2C2_SCL_PIN | DISCOVERY_I2C2_SDA
_PIN));
01031
01032      /*##-2- Unconfigure the Discovery I2C2 p
eripheral #####*/
01033      /* Force and release I2C Peripheral */
01034      DISCOVERY_I2C2_FORCE_RESET();
01035      DISCOVERY_I2C2_RELEASE_RESET();
01036
01037      /* Disable Discovery I2C2 clock */
01038      DISCOVERY_I2C2_CLK_DISABLE();
01039
01040      /* Disable Discovery I2C2 interrupts */
01041      HAL_NVIC_DisableIRQ(DISCOVERY_I2C2_EV_IR
Qn);
01042      HAL_NVIC_DisableIRQ(DISCOVERY_I2C2_ER_IR
Qn);
01043  }
01044 }
01045
01046 /**
01047  * @brief Write a value in a register of t
he device through BUS.
01048  * @param Addr: Device address on BUS Bus.
01049
01049  * @param Reg: The target register address
to write
01050  * @param RegSize: The target register siz

```

```

e (can be 8BIT or 16BIT)
01051     * @param Value: The target register value
        to be written
01052     * @retval None
01053     */
01054 static void I2C2_WriteData(uint16_t Addr, ui
nt16_t Reg, uint16_t RegSize, uint8_t Value)
01055     {
01056     HAL_StatusTypeDef status = HAL_OK;
01057
01058     status = HAL_I2C_Mem_Write(&I2c2Handle, Ad
dr, (uint16_t)Reg, RegSize, &Value, 1, I2c2Timeout
);
01059
01060     /* Check the communication status */
01061     if(status != HAL_OK)
01062     {
01063         /* Re-Initiaize the BUS */
01064         I2C2_Error();
01065     }
01066 }
01067
01068 /**
01069     * @brief Write a value in a register of t
he device through BUS.
01070     * @param Addr: Device address on BUS Bus.

01071     * @param Reg: The target register address
        to write
01072     * @param RegSize: The target register siz
e (can be 8BIT or 16BIT)
01073     * @param pBuffer: The target register val
ue to be written
01074     * @param Length: buffer size to be written

01075     * @retval None
01076     */

```



```

01077 static HAL_StatusTypeDef I2C2_WriteBuffer(ui
nt16_t Addr, uint16_t Reg, uint16_t RegSize, uint8
_t *pBuffer, uint16_t Length)
01078 {
01079     HAL_StatusTypeDef status = HAL_OK;
01080
01081     status = HAL_I2C_Mem_Write(&I2c2Handle, Ad
dr, (uint16_t)Reg, RegSize, pBuffer, Length, I2c2T
imeout);
01082
01083     /* Check the communication status */
01084     if(status != HAL_OK)
01085     {
01086         /* Re-Initiaize the BUS */
01087         I2C2_Error();
01088     }
01089
01090     return status;
01091 }
01092
01093 /**
01094  * @brief Read a register of the device th
rough BUS
01095  * @param Addr: Device address on BUS
01096  * @param Reg: The target register address
to read
01097  * @param RegSize: The target register siz
e (can be 8BIT or 16BIT)
01098  * @retval read register value
01099  */
01100 static uint8_t I2C2_ReadData(uint16_t Addr,
uint16_t Reg, uint16_t RegSize)
01101 {
01102     HAL_StatusTypeDef status = HAL_OK;
01103     uint8_t value = 0x0;
01104
01105     status = HAL_I2C_Mem_Read(&I2c2Handle, Add

```

```

r, Reg, RegSize, &value, 1, I2c2Timeout);
01106
01107     /* Check the communication status */
01108     if(status != HAL_OK)
01109     {
01110         /* Re-Initiaize the BUS */
01111         I2C2_Error();
01112     }
01113
01114     return value;
01115 }
01116
01117 /**
01118  * @brief Reads multiple data on the BUS.
01119  * @param Addr: I2C Address
01120  * @param Reg: Reg Address
01121  * @param RegSize : The target register si
01122  * @param pBuffer: pointer to read data bu
01123  * @param Length: length of the data
01124  * @retval 0 if no problems to read multipl
01125  */
01126 static HAL_StatusTypeDef I2C2_ReadBuffer(uint16_t Addr, uint16_t Reg, uint16_t RegSize, uint8_t *pBuffer, uint16_t Length)
01127 {
01128     HAL_StatusTypeDef status = HAL_OK;
01129
01130     status = HAL_I2C_Mem_Read(&I2c2Handle, Addr, (uint16_t)Reg, RegSize, pBuffer, Length, I2c2Timeout);
01131
01132     /* Check the communication status */
01133     if(status != HAL_OK)
01134     {

```

```

01135     /* Re-Initiaize the BUS */
01136     I2C2_Error();
01137 }
01138
01139     return status;
01140 }
01141
01142 /**
01143  * @brief Discovery I2C2 error treatment fu
01144  * @retval None
01145  */
01146 static void I2C2_Error (void)
01147 {
01148     /* De-initialize the I2C communication BUS
01149     */
01149     HAL_I2C_DeInit(&I2c2Handle);
01150
01151     /* Re- Initiaize the I2C communication BUS
01152     */
01152     I2C2_Init();
01153 }
01154 #endif /*HAL_I2C_MODULE_ENABLED*/
01155
01156
01157 /*****
01158                                     LINK OPERATIONS
01159 *****/
01160 #if defined(HAL_SPI_MODULE_ENABLED)
01161 /***** LINK ACCELEROMETER
01162 *****/
01162 /**
01163  * @brief Configures COMPASS/ACCELEROMETER
01164  * @retval None

```

```

01165     */
01166 void ACCELER0_IO_Init(void)
01167 {
01168     GPIO_InitTypeDef GPIO_InitStructure;
01169
01170     /* Enable CS GPIO clock and Configure GPI
0 PIN for Gyroscope Chip select */
01171     ACCELER0_CS_GPIO_CLK_ENABLE();
01172     GPIO_InitStructure.Pin = ACCELER0_CS_PIN;
01173     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT
_PP;
01174     GPIO_InitStructure.Pull = GPIO_NOPULL;
01175     GPIO_InitStructure.Speed = GPIO_SPEED_HIGH
;
01176     HAL_GPIO_Init(ACCELER0_CS_GPIO_PORT, &GPIO
_InitStructure);
01177
01178     /* Deselect : Chip Select high */
01179     ACCELER0_CS_HIGH();
01180
01181     SPIx_Init();
01182 }
01183
01184 /**
01185  * @brief De-Configures COMPASS/ACCELEROME
TER io interface.
01186  * @retval None
01187  */
01188 void ACCELER0_IO_DeInit(void)
01189 {
01190     GPIO_InitTypeDef GPIO_InitStructure;
01191
01192     /* Enable CS GPIO clock and Configure GPI
0 PIN for Gyroscope Chip select */
01193     ACCELER0_CS_GPIO_CLK_ENABLE();
01194     GPIO_InitStructure.Pin = ACCELER0_CS_PIN;
01195     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT

```

```

_PP;
01196     GPIO_InitStructure.Pull    = GPIO_NOPULL;
01197     GPIO_InitStructure.Speed    = GPIO_SPEED_HIGH
;
01198     HAL_GPIO_Init(ACCELER0_CS_GPIO_PORT, &GPIO
_InitStructure);
01199
01200     /* Deselect : Chip Select high */
01201     ACCELER0_CS_HIGH();
01202
01203     /* Uninitialize SPI bus */
01204     SPIx_DeInit();
01205 }
01206
01207 /**
01208  * @brief Configures COMPASS / ACCELER0 cl
01209  * @retval None
01210  */
01211 void ACCELER0_IO_ITConfig(void)
01212 {
01213 }
01214
01215 /**
01216  * @brief Writes one byte to the COMPASS /
01217  * ACCELEROMETER.
01218  * @param RegisterAddr specifies the COMPA
01219  * SS / ACCELEROMETER register to be written.
01220  * @param Value : Data to be written
01221  * @retval None
01222  */
01223 void ACCELER0_IO_Write(uint8_t RegisterAddr,
uint8_t Value)
01224 {
01225     ACCELER0_CS_LOW();
01226     __SPI_DIRECTION_1LINE_TX(&SpiHandle);
01227     /* call SPI Read data bus function */

```

```

01226     SPIx_Write(RegisterAddr);
01227     SPIx_Write(Value);
01228     ACCELERO_CS_HIGH();
01229 }
01230
01231 /**
01232  * @brief Reads a block of data from the C
01233  * @param RegisterAddr : specifies the COM
01234  * @retval ACCELEROMETER register value
01235  */
01236 uint8_t ACCELERO_IO_Read(uint8_t RegisterAddr)
01237 {
01238     RegisterAddr = RegisterAddr | ((uint8_t)0x80);
01239     ACCELERO_CS_LOW();
01240     __SPI_DIRECTION_1LINE_TX(&SpiHandle);
01241     SPIx_Write(RegisterAddr);
01242     __SPI_DIRECTION_1LINE_RX(&SpiHandle);
01243     uint8_t val = SPIx_Read();
01244     ACCELERO_CS_HIGH();
01245     return val;
01246 }
01247
01248 /***** LINK MAGNETO *****/
01249 /**
01250  * @brief Configures COMPASS/MAGNETO SPI interface.
01251  * @retval None
01252  */
01253 void MAGNETO_IO_Init(void)
01254 {
01255     GPIO_InitTypeDef GPIO_InitStructure;

```

```

01256
01257  /* Enable CS GPIO clock and Configure GPI
0 PIN for Gyroscope Chip select */
01258  MAGNETO_CS_GPIO_CLK_ENABLE();
01259  GPIO_InitStructure.Pin = MAGNETO_CS_PIN;
01260  GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT
_PP;
01261  GPIO_InitStructure.Pull  = GPIO_NOPULL;
01262  GPIO_InitStructure.Speed = GPIO_SPEED_HIGH
;
01263  HAL_GPIO_Init(MAGNETO_CS_GPIO_PORT, &GPIO_
InitStructure);
01264
01265  /* Deselect : Chip Select high */
01266  MAGNETO_CS_HIGH();
01267
01268  SPIx_Init();
01269 }
01270
01271 /**
01272  * @brief de-Configures COMPASS/MAGNETO SP
I interface.
01273  * @retval None
01274  */
01275 void MAGNETO_IO_DeInit(void)
01276 {
01277  GPIO_InitTypeDef GPIO_InitStructure;
01278
01279  /* Enable CS GPIO clock and Configure GPI
0 PIN for Gyroscope Chip select */
01280  MAGNETO_CS_GPIO_CLK_ENABLE();
01281  GPIO_InitStructure.Pin = MAGNETO_CS_PIN;
01282  GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT
_PP;
01283  GPIO_InitStructure.Pull  = GPIO_NOPULL;
01284  GPIO_InitStructure.Speed = GPIO_SPEED_HIGH
;

```

```

01285     HAL_GPIO_Init(MAGNETO_CS_GPIO_PORT, &GPIO_
InitStructure);
01286
01287     /* Deselect : Chip Select high */
01288     MAGNETO_CS_HIGH();
01289
01290     HAL_GPIO_DeInit(MAGNETO_CS_GPIO_PORT, MAG
NETO_INT1_PIN|MAGNETO_DRDY_PIN);
01291
01292
01293     /* Uninitialize SPI bus */
01294     SPIx_DeInit();
01295 }
01296
01297 /**
01298  * @brief Writes one byte to the COMPASS/M
AGNETO.
01299  * @param RegisterAddr specifies the COMPA
SS/MAGNETO register to be written.
01300  * @param Value : Data to be written
01301  * @retval None
01302  */
01303 void MAGNETO_IO_Write(uint8_t RegisterAddr,
uint8_t Value)
01304 {
01305     MAGNETO_CS_LOW();
01306     __SPI_DIRECTION_1LINE_TX(&SpiHandle);
01307     /* call SPI Read data bus function */
01308     SPIx_Write(RegisterAddr);
01309     SPIx_Write(Value);
01310     MAGNETO_CS_HIGH();
01311 }
01312
01313 /**
01314  * @brief Reads a block of data from the C
OMPASS/MAGNETO.
01315  * @param RegisterAddr : specifies the COM

```



PASS/MAGNETO internal address register to read from

```
01316     * @retval ACCELEROMETER register value
01317     */
01318 uint8_t MAGNETO_IO_Read(uint8_t RegisterAddr
01319 )
01319 {
01320     MAGNETO_CS_LOW();
01321     __SPI_DIRECTION_1LINE_TX(&SpiHandle);
01322     SPIx_Write(RegisterAddr | 0x80);
01323     __SPI_DIRECTION_1LINE_RX(&SpiHandle);
01324     uint8_t val = SPIx_Read();
01325     MAGNETO_CS_HIGH();
01326     return val;
01327 }
01328
01329 /***** LINK GYRO
01330 *****/
01330 /**
01331     * @brief Configures the GYRO SPI interface.
01332     * @retval None
01333     */
01334 void GYRO_IO_Init(void)
01335 {
01336     GPIO_InitTypeDef GPIO_InitStructure;
01337
01338
01339     /* Case GYRO not used in the demonstration
01340        software except being set in
01341        low power mode.
01342        To avoid access conflicts with accelerometer and magnetometer,
01343        initialize XL_CS and MAG_CS pins then
01344        deselect these I/O */
01343     ACCELERO_CS_GPIO_CLK_ENABLE();
01344     GPIO_InitStructure.Pin = ACCELERO_CS_PIN;
```

```

01345     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT
_PP;
01346     GPIO_InitStructure.Pull  = GPIO_NOPULL;
01347     GPIO_InitStructure.Speed = GPIO_SPEED_HIGH
;
01348     HAL_GPIO_Init(ACCELERO_CS_GPIO_PORT, &GPIO
_InitStructure);
01349
01350     /* Deselect : Chip Select high */
01351     ACCELERO_CS_HIGH();
01352
01353     /* Enable CS GPIO clock and Configure GP
IO PIN for Gyroscope Chip select */
01354     MAGNETO_CS_GPIO_CLK_ENABLE();
01355     GPIO_InitStructure.Pin = MAGNETO_CS_PIN;
01356     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT
_PP;
01357     GPIO_InitStructure.Pull  = GPIO_NOPULL;
01358     GPIO_InitStructure.Speed = GPIO_SPEED_HIGH
;
01359     HAL_GPIO_Init(MAGNETO_CS_GPIO_PORT, &GPIO_
InitStructure);
01360
01361     /* Deselect : Chip Select high */
01362     MAGNETO_CS_HIGH();
01363
01364
01365     /* Configure the Gyroscope Control pins --
-----*/
01366     /* Enable CS GPIO clock and Configure GPI
O PIN for Gyroscope Chip select */
01367     GYRO_CS_GPIO_CLK_ENABLE();
01368     GPIO_InitStructure.Pin = GYRO_CS_PIN;
01369     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT
_PP;
01370     GPIO_InitStructure.Pull  = GPIO_NOPULL;
01371     GPIO_InitStructure.Speed = GPIO_SPEED_HIGH

```

```

;
01372  HAL_GPIO_Init(GYRO_CS_GPIO_PORT, &GPIO_InitStructure);
01373
01374  /* Deselect : Chip Select high */
01375  GYRO_CS_HIGH();
01376
01377  /* Enable INT1, INT2 GPIO clock and Configure GPIO PINs to detect Interrupts */
01378  GYRO_INT1_GPIO_CLK_ENABLE();
01379  GPIO_InitStructure.Pin = GYRO_INT1_PIN;
01380  GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
01381  GPIO_InitStructure.Speed = GPIO_SPEED_HIGH
;
01382  GPIO_InitStructure.Pull= GPIO_NOPULL;
01383  HAL_GPIO_Init(GYRO_INT1_GPIO_PORT, &GPIO_InitStructure);
01384
01385  GYRO_INT2_GPIO_CLK_ENABLE();
01386  GPIO_InitStructure.Pin = GYRO_INT2_PIN;
01387  HAL_GPIO_Init(GYRO_INT2_GPIO_PORT, &GPIO_InitStructure);
01388
01389  SPIx_Init();
01390
01391 }
01392
01393
01394 /**
01395  * @brief de-Configures GYRO SPI interface.
01396
01397  * @retval None
01398  */
01398 void GYRO_IO_DeInit(void)
01399 {
01400  GPIO_InitTypeDef GPIO_InitStructure;
01401  /* Enable CS GPIO clock */

```

```

01402     GYRO_CS_GPIO_CLK_ENABLE();
01403
01404     GPIO_InitStructure.Pin = GYRO_CS_PIN;
01405     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT
_PP;
01406     GPIO_InitStructure.Pull  = GPIO_NOPULL;
01407     GPIO_InitStructure.Speed = GPIO_SPEED_HIGH
;
01408     HAL_GPIO_Init(GYRO_CS_GPIO_PORT, &GPIO_Ini
tStructure);
01409
01410     /* Deselect : Chip Select high */
01411     GYRO_CS_HIGH();
01412
01413     GYRO_INT1_GPIO_CLK_ENABLE();
01414     GYRO_INT2_GPIO_CLK_ENABLE();
01415
01416     /* Uninitialize the INT1/INT2 Pins */
01417     HAL_GPIO_DeInit(GYRO_INT1_GPIO_PORT, GYRO_
INT1_PIN);
01418     HAL_GPIO_DeInit(GYRO_INT2_GPIO_PORT, GYRO_
INT2_PIN);
01419
01420     /* Uninitialize SPI bus */
01421     SPIx_DeInit();
01422 }
01423
01424 /**
01425  * @brief Writes one byte to the GYRO.
01426  * @param pBuffer : pointer to the buffer
    containing the data to be written to the GYRO.
01427  * @param WriteAddr : GYRO's internal addr
    ess to write to.
01428  * @param NumByteToWrite: Number of bytes
    to write.
01429  * @retval None
01430  */

```

```

01431 void GYRO_IO_Write(uint8_t* pBuffer, uint8_t
    WriteAddr, uint16_t NumByteToWrite)
01432 {
01433     /* Configure the MS bit:
01434         - When 0, the address will remain unc
hanged in multiple read/write commands.
01435         - When 1, the address will be auto in
cremented in multiple read/write commands.
01436     */
01437     if(NumByteToWrite > 0x01)
01438     {
01439         WriteAddr |= (uint8_t)MULTIPLEBYTE_CMD;
01440     }
01441     /* Set chip select Low at the start of the
transmission */
01442     GYRO_CS_LOW();
01443     __SPI_DIRECTION_2LINES(&SpiHandle);
01444
01445     /* Send the Address of the indexed registe
r */
01446     SPIx_WriteRead(WriteAddr);
01447
01448     /* Send the data that will be written into
the device (MSB First) */
01449     while(NumByteToWrite >= 0x01)
01450     {
01451         SPIx_WriteRead(*pBuffer);
01452         NumByteToWrite--;
01453         pBuffer++;
01454     }
01455
01456     /* Set chip select High at the end of the
transmission */
01457     GYRO_CS_HIGH();
01458 }
01459
01460 /**

```

```

01461  * @brief Reads a block of data from the G
GYROSCOPE.
01462  * @param pBuffer : pointer to the buffer
that receives the data read from the GYROSCOPE.
01463  * @param ReadAddr : GYROSCOPE's internal
address to read from.
01464  * @param NumByteToRead : number of bytes
to read from the GYROSCOPE.
01465  * @retval None
01466  */
01467 void GYRO_IO_Read(uint8_t* pBuffer, uint8_t
ReadAddr, uint16_t NumByteToRead)
01468 {
01469     if(NumByteToRead > 0x01)
01470     {
01471         ReadAddr |= (uint8_t)(READWRITE_CMD | MU
LTIPLEBYTE_CMD);
01472     }
01473     else
01474     {
01475         ReadAddr |= (uint8_t)READWRITE_CMD;
01476     }
01477     /* Set chip select Low at the start of the
transmission */
01478     GYRO_CS_LOW();
01479     __SPI_DIRECTION_2LINES(&SpiHandle);
01480     /* Send the Address of the indexed registe
r */
01481     SPIx_WriteRead(ReadAddr);
01482
01483     /* Receive the data that will be read from
the device (MSB First) */
01484     while(NumByteToRead > 0x00)
01485     {
01486         /* Send dummy byte (0x00) to generate th
e SPI clock to GYROSCOPE (Slave device) */
01487         *pBuffer = SPIx_WriteRead(0x00);

```

```

01488     NumByteToRead--;
01489     pBuffer++;
01490 }
01491
01492 /* Set chip select High at the end of the
transmission */
01493     GYRO_CS_HIGH();
01494 }
01495 #endif /* HAL_SPI_MODULE_ENABLED */
01496
01497 #if defined(HAL_I2C_MODULE_ENABLED)
01498 /***** LINK MFX
*****/
01499 /**
01500  * @brief Initializes MFX low level.
01501  * @retval None
01502  */
01503 void MFX_IO_Init(void)
01504 {
01505     /* I2C2 init */
01506     I2C2_Init();
01507 }
01508 /**
01509  * @brief Deinitializes MFX low level.
01510  * @retval None
01511  */
01512 void MFX_IO_DeInit(void)
01513 {
01514     GPIO_InitTypeDef  GPIO_InitStructure;
01515
01516     /* Enable wakeup gpio clock */
01517     IDD_WAKEUP_GPIO_CLK_ENABLE();
01518
01519     /* MFX wakeup pin configuration */
01520     GPIO_InitStructure.Pin    = IDD_WAKEUP_PIN;
01521     GPIO_InitStructure.Mode   = GPIO_MODE_OUTPUT_P
P;

```

```

01522     GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
01523     GPIO_InitStruct.Pull  = GPIO_PULLDOWN;
01524     HAL_GPIO_Init(IDD_WAKEUP_GPIO_PORT, &GPIO_
InitStruct);
01525
01526     /* DeInit interrupt pin : disable IRQ befo
re to avoid spurious interrupt */
01527     HAL_NVIC_DisableIRQ((IRQn_Type)(IDD_INT_EX
TI_IRQn));
01528     IDD_INT_GPIO_CLK_ENABLE();
01529     HAL_GPIO_DeInit(IDD_INT_GPIO_PORT, IDD_INT
_PIN);
01530
01531     /* I2C2 Deinit */
01532     I2C2_DeInit();
01533 }
01534
01535 /**
01536  * @brief Configures MFX low level interrui
pt.
01537  * @retval None
01538  */
01539 void MFX_IO_ITConfig(void)
01540 {
01541     GPIO_InitTypeDef  GPIO_InitStruct;
01542
01543     /* Enable the GPIO clock */
01544     IDD_INT_GPIO_CLK_ENABLE();
01545
01546     /* MFX_OUT_IRQ (normally used for EXTI_WKU
P) */
01547     GPIO_InitStruct.Pin    = IDD_INT_PIN;
01548     GPIO_InitStruct.Pull   = GPIO_PULLDOWN;
01549     GPIO_InitStruct.Speed  = GPIO_SPEED_HIGH;
01550     GPIO_InitStruct.Mode   = GPIO_MODE_IT_RISIN
G;
01551     HAL_GPIO_Init(IDD_INT_GPIO_PORT, &GPIO_Ini

```



```

tStruct);
01552
01553  /* Enable and set GPIO EXTI Interrupt to t
he lowest priority */
01554  HAL_NVIC_SetPriority((IRQn_Type)(IDD_INT_E
XTI_IRQn), 0x0F, 0x0F);
01555  HAL_NVIC_EnableIRQ((IRQn_Type)(IDD_INT_EXT
I_IRQn));
01556 }
01557
01558 /**
01559  * @brief Configures MFX wke up pin.
01560  * @retval None
01561  */
01562 void MFX_IO_EnableWakeupPin(void)
01563 {
01564     GPIO_InitTypeDef  GPIO_InitStructure;
01565
01566     /* Enable wakeup gpio clock */
01567     IDD_WAKEUP_GPIO_CLK_ENABLE();
01568
01569     /* MFX wakeup pin configuration */
01570     GPIO_InitStructure.Pin    = IDD_WAKEUP_PIN;
01571     GPIO_InitStructure.Mode   = GPIO_MODE_OUTPUT_P
P;
01572     GPIO_InitStructure.Speed  = GPIO_SPEED_FAST;
01573     GPIO_InitStructure.Pull   = GPIO_NOPULL;
01574     HAL_GPIO_Init(IDD_WAKEUP_GPIO_PORT, &GPIO_
InitStruct);
01575 }
01576
01577 /**
01578  * @brief Wakeup MFX.
01579  * @retval None
01580  */
01581 void MFX_IO_Wakeup(void)
01582 {

```

```

01583  /* Set Wakeup pin to high to wakeup Idd me
asurement component from standby mode */
01584  HAL_GPIO_WritePin(IDD_WAKEUP_GPIO_PORT, ID
D_WAKEUP_PIN, GPIO_PIN_SET);
01585
01586  /* Wait */
01587  HAL_Delay(1);
01588
01589  /* Set gpio pin basck to low */
01590  HAL_GPIO_WritePin(IDD_WAKEUP_GPIO_PORT, ID
D_WAKEUP_PIN, GPIO_PIN_RESET);
01591 }
01592
01593 /**
01594  * @brief MFX writes single data.
01595  * @param Addr: I2C address
01596  * @param Reg: Register address
01597  * @param Value: Data to be written
01598  * @retval None
01599  */
01600 void MFX_IO_Write(uint16_t Addr, uint8_t Reg
, uint8_t Value)
01601 {
01602     I2C2_WriteData(Addr, Reg, I2C_MEMADD_SIZE_
8BIT, Value);
01603 }
01604
01605 /**
01606  * @brief MFX reads single data.
01607  * @param Addr: I2C address
01608  * @param Reg: Register address
01609  * @retval Read data
01610  */
01611 uint8_t MFX_IO_Read(uint16_t Addr, uint8_t R
eg)
01612 {
01613     return I2C2_ReadData(Addr, Reg, I2C_MEMADD

```

```

_SIZE_8BIT);
01614 }
01615
01616 /**
01617  * @brief MFX reads multiple data.
01618  * @param Addr: I2C address
01619  * @param Reg: Register address
01620  * @param Buffer: Pointer to data buffer
01621  * @param Length: Length of the data
01622  * @retval Number of read data
01623  */
01624 uint16_t MFX_IO_ReadMultiple(uint16_t Addr,
uint8_t Reg, uint8_t *Buffer, uint16_t Length)
01625 {
01626     return I2C2_ReadBuffer(Addr, (uint16_t)Reg,
I2C_MEMADD_SIZE_8BIT, Buffer, Length);
01627 }
01628
01629 /**
01630  * @brief MFX writes multiple data.
01631  * @param Addr: I2C address
01632  * @param Reg: Register address
01633  * @param Buffer: Pointer to data buffer
01634  * @param Length: Length of the data
01635  * @retval None
01636  */
01637 void MFX_IO_WriteMultiple(uint16_t Addr, uint8_t Reg, uint8_t *Buffer, uint16_t Length)
01638 {
01639     I2C2_WriteBuffer(Addr, (uint16_t)Reg, I2C_MEMADD_SIZE_8BIT, Buffer, Length);
01640 }
01641
01642 /**
01643  * @brief MFX delay
01644  * @param Delay: Delay in ms
01645  * @retval None

```

```

01646     */
01647 void MFX_IO_Delay(uint32_t Delay)
01648 {
01649     HAL_Delay(Delay);
01650 }
01651
01652
01653 /***** LINK AUDIO
0 *****/
01654 /**
01655  * @brief Initializes Audio low level.
01656  * @retval None
01657  */
01658 void AUDIO_IO_Init(void)
01659 {
01660     GPIO_InitTypeDef  GPIO_InitStructure;
01661
01662     /* Enable Reset GPIO Clock */
01663     AUDIO_RESET_GPIO_CLK_ENABLE();
01664
01665     /* Audio reset pin configuration */
01666     GPIO_InitStructure.Pin = AUDIO_RESET_PIN;
01667     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP
;
01668     GPIO_InitStructure.Speed = GPIO_SPEED_FAST;
01669     GPIO_InitStructure.Pull = GPIO_NOPULL;
01670     HAL_GPIO_Init(AUDIO_RESET_GPIO, &GPIO_Init
Struct);
01671
01672     /* I2C bus init */
01673     I2C1_Init();
01674
01675     /* Power Down the codec */
01676     CODEC_AUDIO_POWER_OFF();
01677
01678     /* wait for a delay to insure registers er
asing */

```

```

01679     HAL_Delay(5);
01680
01681     /* Power on the codec */
01682     CODEC_AUDIO_POWER_ON();
01683
01684     /* wait for a delay to insure registers er
asing */
01685     HAL_Delay(5);
01686 }
01687
01688 /**
01689  * @brief Deinitializes Audio low level.
01690  * @retval None
01691  */
01692 void AUDIO_IO_DeInit(void)
    /* TO DO */
01693 {
01694     GPIO_InitTypeDef  GPIO_InitStructure;
01695
01696     /*****
    *****/
01697     /* In case of battery-supplied powered, th
ere is no audio codec-based
01698         features available. Set audio codec I/O
default setting */
01699     /*****
    *****/
01700     __HAL_RCC_GPIOE_CLK_ENABLE();
01701     GPIO_InitStructure.Mode      = GPIO_MODE_OUTP
UT_PP ;
01702     GPIO_InitStructure.Pin       = (GPIO_PIN_2 |
GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6)
;
01703     GPIO_InitStructure.Pull      = GPIO_PULLDOWN;
01704     GPIO_InitStructure.Speed     = GPIO_SPEED_HIG
H;
01705     HAL_GPIO_Init(GPIOE, &GPIO_InitStructure);

```

```

01706     HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_
PIN_RESET);
01707     HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_
PIN_RESET);
01708     HAL_GPIO_WritePin(GPIOE, GPIO_PIN_4, GPIO_
PIN_RESET);
01709     HAL_GPIO_WritePin(GPIOE, GPIO_PIN_5, GPIO_
PIN_RESET);
01710     HAL_GPIO_WritePin(GPIOE, GPIO_PIN_6, GPIO_
PIN_RESET);
01711
01712     /* I2C bus Deinit */
01713     I2C1_DeInit();
01714 }
01715
01716 /**
01717  * @brief Writes a single data.
01718  * @param Addr: I2C address
01719  * @param Reg: Reg address
01720  * @param Value: Data to be written
01721  * @retval None
01722  */
01723 void AUDIO_IO_Write(uint8_t Addr, uint8_t Re
g, uint8_t Value)
01724 {
01725     I2C1_WriteBuffer(Addr, (uint16_t)Reg, I2C_
MEMADD_SIZE_8BIT, &Value, 1);
01726 }
01727
01728 /**
01729  * @brief Reads a single data.
01730  * @param Addr: I2C address
01731  * @param Reg: Reg address
01732  * @retval Data to be read
01733  */
01734 uint8_t AUDIO_IO_Read(uint8_t Addr, uint8_t
Reg)

```

```

01735 {
01736     uint8_t Read_Value = 0;
01737
01738     I2C1_ReadBuffer((uint16_t) Addr, (uint16_t
) Reg, I2C_MEMADD_SIZE_8BIT, &Read_Value, 1);
01739
01740     return Read_Value;
01741 }
01742
01743 /**
01744  * @brief AUDIO Codec delay
01745  * @param Delay: Delay in ms
01746  * @retval None
01747  */
01748 void AUDIO_IO_Delay(uint32_t Delay)
01749 {
01750     HAL_Delay(Delay);
01751 }
01752 #endif /* HAL_I2C_MODULE_ENABLED */
01753
01754 /**
01755  * @}
01756  */
01757
01758 /**
01759  * @}
01760  */
01761
01762 /**
01763  * @}
01764  */
01765
01766 /**
01767  * @}
01768  */
01769
01770 /** ***** (C) COPYRIGHT STMi

```

```
croelectronics *****END OF FILE*****/
```



---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_accelerometer.h

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      ****
00003      * @file      stm32l476g_discovery_accelerome
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file contains definitions
00008      *            for stm32l476g_discovery_accelerometer.c
00009      *            firmware driver.
00010      *            ****
00011      *            ****
00012      * @attention
00013      *
00014      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00015      * icroelectronics</center></h2>
00016      *
00017      * Redistribution and use in source and bin
00018      * ary forms, with or without modification,
00019      * are permitted provided that the followin
00020      * g conditions are met:
```

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

```

00035      *
00036      ****
00037      */
00038
00039 /* Define to prevent recursive inclusion ---
-----*/
00040 #ifndef __STM32L476G_DISCOVERY_ACCELEROMETER
_H
00041 #define __STM32L476G_DISCOVERY_ACCELEROMETER
_H
00042
00043 #ifdef __cplusplus
00044     extern "C" {
00045 #endif
00046
00047 /* Includes -----
-----*/
00048 #include "stm32l476g_discovery.h"
00049 /* Include Gyroscope component driver */
00050 #include "../Components/lsm303dlhc/lsm303dlh
c.h"
00051
00052 /** @addtogroup BSP
00053     * @{
00054     */
00055
00056 /** @addtogroup STM32L476G_DISCOVERY
00057     * @{
00058     */
00059
00060 /** @addtogroup STM32L476G_DISCOVERY_ACCELER
OMETER
00061     * @{
00062     */
00063
00064 /** @defgroup STM32L476G_DISCOVERY_ACCELEROM

```

## ETER\_Exported\_Types Exported Types

```
00065      * @{
00066      */
00067
00068 /**
00069      * @}
00070      */
00071
00072 /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
```

## ETER\_Exported\_Constants Exported Constants

```
00073      * @{
00074      */
00075 typedef enum
00076 {
00077     ACCELERO_OK = 0,
00078     ACCELERO_ERROR = 1,
00079     ACCELERO_TIMEOUT = 2
00080 }
00081 ACCELERO_StatusTypeDef;
```

```
00082
00083 /**
00084      * @}
00085      */
00086
00087 /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
```

## ETER\_Exported\_Macros Exported Macros

```
00088      * @{
00089      */
00090
00091 /**
00092      * @}
00093      */
00094
00095 /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
```

## ETER\_Exported\_Functions Exported Functions

```
00096      * @{
00097      */
```

```

00098 /* Accelerometer functions */
00099 uint8_t   BSP_ACCELER0_Init(void);
00100 void      BSP_ACCELER0_Reset(void);
00101 void      BSP_ACCELER0_GetXYZ(int16_t *pData
XYZ);
00102
00103 /**
00104  * @}
00105  */
00106
00107 /**
00108  * @}
00109  */
00110
00111 /**
00112  * @}
00113  */
00114
00115 /**
00116  * @}
00117  */
00118
00119 #ifdef __cplusplus
00120 }
00121 #endif
00122
00123 #endif /* __STM32L476G_DISCOVERY_ACCELEROMET
ER_H */
00124
00125 /***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE*****/

```

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_accelerometer.c

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      ****
00003  * @file    stm32l476g_discovery_accelerometer.c
00004  * @author  MCD Application Team
00005  * @version $VERSION$
00006  * @date    $DATE$
00007  * @brief   This file provides a set of functions needed to manage the ACCELEROMETER
00008  *          MEMS available on STM32L476G-Discovery Kit.
00009      ****
00009      ****
00010  * @attention
00011  *
00012  * <h2><center>&copy; COPYRIGHT(c) 2015 STMicroelectronics</center></h2>
00013  *
00014  * Redistribution and use in source and binary forms, with or without modification,
00015  * are permitted provided that the following
```

g conditions are met:

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,

00017 \* this list of conditions and the fol  
lowing disclaimer.

00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,

00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation

00020 \* and/or other materials provided wit  
h the distribution.

00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors

00022 \* may be used to endorse or promote p  
roducts derived from this software

00023 \* without specific prior written perm  
ission.

00024 \*

00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"

00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE

00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE

00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE

00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL

00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR

00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER

00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,

00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE

00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE

## POSSIBILITY OF SUCH DAMAGE.

```
00035      *
00036      ****
*****
*****
00037      */
00038
00039 /* Includes -----
----- */
00040 #include "stm32l476g_discovery_accelerometer
.h"
00041
00042 /** @addtogroup BSP
00043     * @{
00044     */
00045
00046 /** @addtogroup STM32L476G_DISCOVERY
00047     * @{
00048     */
00049
00050 /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
ETER STM32L476G-DISCOVERY ACCELEROMETER
00051     * @{
00052     */
00053
00054 /* Private typedef -----
----- */
00055 /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
ETER_Private_Types Private Types
00056     * @{
00057     */
00058 /**
00059     * @}
00060     */
00061
00062 /* Private defines -----
----- */
00063 /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
```



```

ETER_Private_Constants Private Constants
00064      * @{
00065      */
00066  /**
00067      * @}
00068      */
00069
00070  /* Private macros -----
-----*/
00071  /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
ETER_Private_Macros Private Macros
00072      * @{
00073      */
00074  /**
00075      * @}
00076      */
00077
00078  /* Private variables -----
-----*/
00079  /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
ETER_Private_Variables Private Variables
00080      * @{
00081      */
00082  static ACCELER0_DrvTypeDef *AccelerometerDrv
;
00083
00084  /**
00085      * @}
00086      */
00087
00088  /* Private function prototypes -----
-----*/
00089  /** @defgroup STM32L476G_DISCOVERY_ACCELEROM
ETER_Private_FunctionPrototypes Private Functions
00090      * @{
00091      */
00092  /**

```

```

00093     * @}
00094     */
00095
00096 /* Exported functions -----
-----*/
00097 /** @addtogroup STM32L476G_DISCOVERY_ACCELER
OMETER_Exported_Functions
00098     * @{
00099     */
00100
00101 /**
00102     * @brief Initialize Accelerometer.
00103     * @retval ACCELER0_OK or ACCELER0_ERROR
00104     */
00105 uint8_t BSP_ACCELER0_Init(void)
00106 {
00107     uint8_t ret = ACCELER0_ERROR;
00108     uint16_t ctrl = 0x0000;
00109     ACCELER0_InitTypeDef LSM303DLHC_InitStruct
ure;
00110     ACCELER0_FilterConfigTypeDef LSM303DLHC_Fi
lterStructure;
00111
00112     if(Lsm303dlhcDrv.ReadID() == I_AM_LMS303DL
HC)
00113     {
00114         /* Initialize the gyroscope driver struc
ture */
00115         AccelerometerDrv = &Lsm303dlhcDrv;
00116
00117         /* MEMS configuration -----
-----*/
00118         /* Fill the accelerometer structure */
00119         LSM303DLHC_InitStructure.Power_Mode = LS
M303DLHC_NORMAL_MODE;
00120         LSM303DLHC_InitStructure.AccOutput_DataR
ate = LSM303DLHC_ODR_50_HZ;

```

```

00121     LSM303DLHC_InitStructure.Axes_Enable= LSM303DLHC_AXES_ENABLE;
00122     LSM303DLHC_InitStructure.AccFull_Scale = LSM303DLHC_FULLSCALE_2G;
00123     LSM303DLHC_InitStructure.BlockData_Update = LSM303DLHC_BlockUpdate_Continuous;
00124     LSM303DLHC_InitStructure.Endianness=LSM303DLHC_BLE_LSB;
00125     LSM303DLHC_InitStructure.High_Resolution=LSM303DLHC_HR_ENABLE;
00126
00127     /* Configure MEMS: data rate, power mode, full scale and axes */
00128     ctrl |= (LSM303DLHC_InitStructure.Power_Mode | LSM303DLHC_InitStructure.AccOutput_DataRate | \
00129             LSM303DLHC_InitStructure.Axes_Enable);
00130
00131     ctrl |= ((LSM303DLHC_InitStructure.BlockData_Update | LSM303DLHC_InitStructure.Endianness | \
00132             LSM303DLHC_InitStructure.AccFull_Scale | LSM303DLHC_InitStructure.High_Resolution) << 8);
00133
00134     /* Configure the accelerometer main parameters */
00135     AccelerometerDrv->Init(ctrl);
00136
00137     /* Fill the accelerometer LPF structure */
00138     LSM303DLHC_FilterStructure.HighPassFilter_Mode_Selection =LSM303DLHC_HPM_NORMAL_MODE;
00139     LSM303DLHC_FilterStructure.HighPassFilter_CutOff_Frequency = LSM303DLHC_HPFCF_16;
00140     LSM303DLHC_FilterStructure.HighPassFilter

```

```

r_AOI1 = LSM303DLHC_HPF_AOI1_DISABLE;
00141     LSM303DLHC_FilterStructure.HighPassFilter
r_AOI2 = LSM303DLHC_HPF_AOI2_DISABLE;
00142
00143     /* Configure MEMS: mode, cutoff frequency
, Filter status, Click, AOI1 and AOI2 */
00144     ctrl = (uint8_t) (LSM303DLHC_FilterStructure.HighPassFilter_Mode_Selection | \
00145                     LSM303DLHC_FilterStructure.HighPassFilter_CutOff_Frequency | \
00146                     LSM303DLHC_FilterStructure.HighPassFilter_AOI1 | \
00147                     LSM303DLHC_FilterStructure.HighPassFilter_AOI2);
00148
00149     /* Configure the accelerometer LPF main
parameters */
00150     AccelerometerDrv->FilterConfig(ctrl);
00151
00152     ret = ACCELER0_OK;
00153 }
00154 else
00155 {
00156     ret = ACCELER0_ERROR;
00157 }
00158
00159     return ret;
00160 }
00161
00162 /**
00163  * @brief Reboot memory content of Accelerometer.
00164  * @retval None
00165  */
00166 void BSP_ACCELER0_Reset(void)
00167 {
00168     if(AccelerometerDrv->Reset != NULL)

```

```

00169     {
00170         AccelerometerDrv->Reset();
00171     }
00172 }
00173
00174 /**
00175  * @brief Get XYZ angular accelerations fr
00176  * @param pDataXYZ Pointer on 3 angular ac
00177  *          pDataXYZ[0] = X axis, p
00178  *          DataXYZ[1] = Y axis, pDataXYZ[2] = Z axis
00179  * @retval None
00180  */
00180 void BSP_ACCELERO_GetXYZ(int16_t *pDataXYZ)
00181 {
00182     if(AccelerometerDrv->GetXYZ!= NULL)
00183     {
00184         AccelerometerDrv->GetXYZ(pDataXYZ);
00185     }
00186 }
00187
00188 /**
00189  * @}
00190  */
00191
00192 /**
00193  * @}
00194  */
00195
00196 /**
00197  * @}
00198  */
00199
00200 /**
00201  * @}
00202  */

```

00203

00204 /\*\*\*\*\* (C) COPYRIGHT STMicroelectronics \*\*\*\*\*/

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_audio.h

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery_audio.h
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file contains the common d
00008      *             efines and functions prototypes for
00009      *             the stm32l476g_discovery_audio.
00010      *             c driver.
00011      *             ****
00012      * @attention
00013      *
00014      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00015      * icroelectronics</center></h2>
00016      *
00017      * Redistribution and use in source and bin
00018      * ary forms, with or without modification,
00019      * are permitted provided that the followin
00020      * g conditions are met:
```

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.



```

00035      *
00036      ****
00037      */
00038
00039  /* Define to prevent recursive inclusion ---
-----*/
00040  #ifndef __STM32L476G_DISCOVERY_AUDIO_H
00041  #define __STM32L476G_DISCOVERY_AUDIO_H
00042
00043  #ifdef __cplusplus
00044      extern "C" {
00045  #endif
00046
00047  /* Includes -----
-----*/
00048  #if defined(BSP_AUDIO_USE_RTOS)
00049  #include "k_mem.h"
00050  #else
00051  #include <stdlib.h>
00052  #endif
00053  /* Include audio component Driver */
00054  #include "../Components/cs43l22/cs43l22.h"
00055  #include "stm32l476g_discovery.h"
00056
00057  /** @addtogroup BSP
00058      * @{
00059      */
00060
00061  /** @addtogroup STM32L476G_DISCOVERY
00062      * @{
00063      */
00064
00065  /** @addtogroup STM32L476G_DISCOVERY_AUDIO
00066      * @{
00067      */
00068

```

```

00069 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Exp
orted_Types Exported Types
00070     * @{
00071     */
00072 typedef void (*Audio_CallbackTypeDef)(void);
00073
00074 /**
00075     * @}
00076     */
00077
00078 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Exp
orted_Constants Exported Constants
00079     * @{
00080     */
00081 /** @defgroup BSP_Audio_Out_Option BSP Audio
Out Option
00082     * @{
00083     */
00084 #define BSP_AUDIO_OUT_CIRCULARMODE ((ui
nt32_t)0x00000001) /* BUFFER CIRCULAR MODE */
00085 #define BSP_AUDIO_OUT_NORMALMODE ((ui
nt32_t)0x00000002) /* BUFFER NORMAL MODE */
00086 #define BSP_AUDIO_OUT_STEREO MODE ((ui
nt32_t)0x00000004) /* STEREO MODE */
00087 #define BSP_AUDIO_OUT_MONOMODE ((ui
nt32_t)0x00000008) /* MONO MODE */
00088 /**
00089     * @}
00090     */
00091
00092 /** @defgroup BSP_Audio_Sample_Rate BSP Audi
o Sample Rate
00093     * @{
00094     */
00095 #define BSP_AUDIO_FREQUENCY_96K SAI_
AUDIO_FREQUENCY_96K
00096 #define BSP_AUDIO_FREQUENCY_48K SAI_

```

```

AUDIO_FREQUENCY_48K
00097 #define BSP_AUDIO_FREQUENCY_44K          SAI_
AUDIO_FREQUENCY_44K
00098 #define BSP_AUDIO_FREQUENCY_32K          SAI_
AUDIO_FREQUENCY_32K
00099 #define BSP_AUDIO_FREQUENCY_22K          SAI_
AUDIO_FREQUENCY_22K
00100 #define BSP_AUDIO_FREQUENCY_16K          SAI_
AUDIO_FREQUENCY_16K
00101 #define BSP_AUDIO_FREQUENCY_11K          SAI_
AUDIO_FREQUENCY_11K
00102 #define BSP_AUDIO_FREQUENCY_8K           SAI_
AUDIO_FREQUENCY_8K
00103 /**
00104     * @}
00105     */
00106 /*-----
-----
00107                                USER SAI defines p
arameters
00108 -----
-----*/
00109 /* SAI peripheral configuration defines */
00110 #define AUDIO_SAIx
        SAI1_Block_A
00111 #define AUDIO_SAIx_CLK_ENABLE()
        __HAL_RCC_SAI1_CLK_ENABLE()
00112 #define AUDIO_SAIx_CLK_DISABLE()
        __HAL_RCC_SAI1_CLK_DISABLE()
00113 #define AUDIO_SAIx_MCK_SCK_SD_FS_AF
        GPIO_AF13_SAI1
00114
00115 #define AUDIO_SAIx_MCK_SCK_SD_FS_ENABLE()
        __HAL_RCC_GPIOE_CLK_ENABLE()
00116 #define AUDIO_SAIx_MCK_SCK_SD_FS_DISABLE()
        __HAL_RCC_GPIOE_CLK_DISABLE()
00117 #define AUDIO_SAIx_FS_PIN

```

```

    GPIO_PIN_4
00118 #define AUDIO_SAIx_SCK_PIN
    GPIO_PIN_5
00119 #define AUDIO_SAIx_SD_PIN
    GPIO_PIN_6
00120 #define AUDIO_SAIx_MCK_PIN
    GPIO_PIN_2
00121 #define AUDIO_SAIx_MCK_SCK_SD_FS_GPIO_PORT
    GPIOE
00122
00123 /* SAI DMA Channel definitions */
00124 #define AUDIO_SAIx_DMAX_CLK_ENABLE()
    __HAL_RCC_DMA2_CLK_ENABLE()
00125 #define AUDIO_SAIx_DMAX_CLK_DISABLE()
    __HAL_RCC_DMA2_CLK_DISABLE()
00126 #define AUDIO_SAIx_DMAX_CHANNEL
    DMA2_Channel1
00127 #define AUDIO_SAIx_DMAX_IRQ
    DMA2_Channel1_IRQn
00128 #define AUDIO_SAIx_DMAX_PERIPH_DATA_SIZE
    DMA_PDATAALIGN_HALFWORD
00129 #define AUDIO_SAIx_DMAX_MEM_DATA_SIZE
    DMA_MDATAALIGN_HALFWORD
00130 #define DMA_MAX_SZE
    (uint32_t)0xFFFF
00131
00132 #define AUDIO_SAIx_DMAX_IRQHandler
    DMA2_Channel1_IRQHandler
00133
00134 /* Select the interrupt preemption priority
for the DMA interrupt */
00135 #define AUDIO_OUT_IRQ_PREPRIO                    5
/* Select the preemption priority level(0 is the h
ighest) */
00136
00137 /* Disable SAIx PLL */
00138 #define AUDIO_SAIx_PLL_DISABLE()

```

```

    HAL_RCCEx_DisablePLLSAI1()
00139
00140 /*-----
-----
00141                                AUDIO IN CONFIGURATI
ON
00142 -----
-----*/
00143 /* DFSDM Configuration defines */
00144 #define AUDIO_DFSDMx_LEFT_CHANNEL
        DFSDM_Channel12
00145 #define AUDIO_DFSDMx_LEFT_FILTER
        DFSDM_Filter0
00146 #define AUDIO_DFSDMx_CLK_ENABLE()
        __HAL_RCC_DFSDM_CLK_ENABLE()
00147 #define AUDIO_DFSDMx_CLK_DISABLE()
        __HAL_RCC_DFSDM_CLK_DISABLE()
00148 #define AUDIO_DFSDMx_CKOUT_PIN
        GPIO_PIN_9
00149 #define AUDIO_DFSDMx_DMIC_DATIN_PIN
        GPIO_PIN_7
00150 #define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_P
ORT        GPIOE
00151 #define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_C
LK_ENABLE() __HAL_RCC_GPIOE_CLK_ENABLE()
00152 #define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_GPIO_C
LK_DISABLE() __HAL_RCC_GPIOE_CLK_DISABLE()
00153 #define AUDIO_DFSDMx_CKOUT_DMIC_DATIN_AF
        GPIO_AF6_DFSDM
00154
00155 /* DFSDM DMA Right and Left channels definit
ions */
00156 #define AUDIO_DFSDMx_DMax_CLK_ENABLE()
        __HAL_RCC_DMA1_CLK_ENABLE()
00157 #define AUDIO_DFSDMx_DMax_CLK_DISABLE()
        __HAL_RCC_DMA1_CLK_DISABLE()
00158 #define AUDIO_DFSDMx_DMax_LEFT_CHANNEL

```

```

DMA1_Channel4
00159 #define AUDIO_DFSDMx_DMAX_LEFT_IRQ
DMA1_Channel4_IRQn
00160 #define AUDIO_DFSDMx_DMAX_PERIPH_DATA_SIZE
DMA_PDATAALIGN_WORD
00161 #define AUDIO_DFSDMx_DMAX_MEM_DATA_SIZE
DMA_MDATAALIGN_WORD
00162
00163 #define AUDIO_DFSDM_DMAX_LEFT_IRQHandler
DMA1_Channel4_IRQHandler
00164
00165 /* Select the interrupt preemption priority
and subpriority for the IT/DMA interrupt */
00166 #define AUDIO_IN_IRQ_PREPRIO
6 /* Select the preemption priority level(0 is t
he highest) */
00167
00168 /*-----
-----
00169 CONFIGURATION: Audio Driver Con
figuration parameters
00170 -----
-----*/
00171
00172 #define AUDIODATA_SIZE
2 /* 16-bits audio data size */
00173
00174 /* Audio status definition */
00175 #define AUDIO_OK 0
00176 #define AUDIO_ERROR 1
00177 #define AUDIO_TIMEOUT 2
00178
00179 /* AudioFreq * DataSize (2 bytes) * NumChann
els (Stereo: 2) */

```

```

00180 #define DEFAULT_AUDIO_IN_FREQ
BSP_AUDIO_FREQUENCY_16K
00181 #define DEFAULT_AUDIO_IN_BIT_RESOLUTION
16
00182 #define DEFAULT_AUDIO_IN_CHANNEL_NBR
1 /* Mono = 1, Stereo = 2 */
00183 #define DEFAULT_AUDIO_IN_VOLUME
64
00184
00185 /*-----
-----
00186                      OPTIONAL Configuration d
efines parameters
00187 -----
-----*/
00188
00189 /* Delay for the Codec to be correctly reset
*/
00190 #define CODEC_RESET_DELAY          5
00191
00192 /**
00193  * @}
00194  */
00195
00196 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Exp
orted_Variables Exported Variables
00197  * @{
00198  */
00199 extern SAI_HandleTypeDef          BSP_AUDIO_
hSai;
00200 extern DFSDM_Filter_HandleTypeDef BSP_AUDIO_
hDfsdmLeftFilter;
00201
00202 /**
00203  * @}
00204  */
00205

```

```

00206 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Exp
orted_Macros Exported Macros
00207     * @{
00208     */
00209 #define DMA_MAX(_X_)                (((_X_)
<= DMA_MAX_SIZE)? (_X_):DMA_MAX_SIZE)
00210
00211 /**
00212     * @}
00213     */
00214
00215 /* Exported functions -----
-----*/
00216 /** @defgroup STM32L476G_DISCOVERY_AUDIO_Exp
orted_Functions Exported Functions
00217     * @{
00218     */
00219 uint8_t BSP_AUDIO_OUT_Init(uint16_t OutputDe
vice, uint8_t Volume, uint32_t AudioFreq);
00220 uint8_t BSP_AUDIO_OUT_DeInit(void);
00221 uint8_t BSP_AUDIO_OUT_Play(uint16_t* pData,
uint32_t Size);
00222 uint8_t BSP_AUDIO_OUT_ChangeBuffer(uint16_t
*pData, uint16_t Size);
00223 uint8_t BSP_AUDIO_OUT_Pause(void);
00224 uint8_t BSP_AUDIO_OUT_Resume(void);
00225 uint8_t BSP_AUDIO_OUT_Stop(uint32_t Option);
00226 uint8_t BSP_AUDIO_OUT_SetVolume(uint8_t Volu
me);
00227 uint8_t BSP_AUDIO_OUT_SetFrequency(uint32_t
AudioFreq);
00228 void BSP_AUDIO_OUT_ChangeAudioConfig(uint
32_t AudioOutOption);
00229 uint8_t BSP_AUDIO_OUT_SetMute(uint32_t Cmd);
00230 uint8_t BSP_AUDIO_OUT_SetOutputMode(uint8_t
Output);
00231 void BSP_AUDIO_OUT_RegisterCallbacks(Audi

```



```

o_CallbackTypeDef ErrorCallback,
00232
o_CallbackTypeDef HalfTransferCallback,
00233
o_CallbackTypeDef TransferCompleteCallback);
00234
00235 uint8_t BSP_AUDIO_IN_Init(uint32_t AudioFreq
, uint32_t BitRes, uint32_t ChnlNbr);
00236 uint8_t BSP_AUDIO_IN_DeInit(void);
00237 uint8_t BSP_AUDIO_IN_Record(uint16_t *pData,
uint32_t Size);
00238 uint8_t BSP_AUDIO_IN_SetFrequency(uint32_t A
udioFreq);
00239 uint8_t BSP_AUDIO_IN_Stop(void);
00240 uint8_t BSP_AUDIO_IN_Pause(void);
00241 uint8_t BSP_AUDIO_IN_Resume(void);
00242 void BSP_AUDIO_IN_RegisterCallbacks(Audio
_CallbackTypeDef ErrorCallback,
00243
_CallbackTypeDef HalfTransferCallback,
00244
_CallbackTypeDef TransferCompleteCallback);
00245 /**
00246  * @}
00247  */
00248
00249 /**
00250  * @}
00251  */
00252
00253 /**
00254  * @}
00255  */
00256
00257 /**
00258  * @}
00259  */

```

```
00260
00261 #ifdef __cplusplus
00262 }
00263 #endif
00264
00265 #endif /* __STM32L476G_DISCOVERY_AUDIO_H */
00266
00267 /***** (C) COPYRIGHT STMicroelectronics *****/
00268 *****/
```

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_compass.h

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery_compass.h
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file contains definitions
00008      *            for stm32l476g_discovery_compass.c
00009      *            firmware driver.
00010      *            ****
00011      * @attention
00012      *
00013      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00014      * icroelectronics</center></h2>
00015      *
00016      * Redistribution and use in source and bin
00017      * ary forms, with or without modification,
00018      * are permitted provided that the followin
00019      * g conditions are met:
00020      * 1. Redistributions of source code must
```

retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.  
00035 \*

```

00036      ****
00037      */
00038
00039 /* Define to prevent recursive inclusion ---
-----*/
00040 #ifndef __STM32L476G_DISCOVERY_COMPASS_H
00041 #define __STM32L476G_DISCOVERY_COMPASS_H
00042
00043 #ifdef __cplusplus
00044     extern "C" {
00045 #endif
00046
00047 /* Includes -----
-----*/
00048 #include "stm32l476g_discovery.h"
00049
00050 /** @addtogroup BSP
00051     * @{
00052     */
00053
00054 /** @addtogroup STM32L476G_DISCOVERY
00055     * @{
00056     */
00057
00058 /** @addtogroup STM32L476G_DISCOVERY_COMPASS
00059     * @{
00060     */
00061
00062 /** @defgroup STM32L476G_DISCOVERY_COMPASS_E
xported_Types Exported Types
00063     * @{
00064     */
00065
00066 /**
00067     * @}
00068     */

```

```

00069
00070 /** @defgroup STM32L476G_DISCOVERY_COMPASS_E
Exported_Constants Exported Constants
00071 * @{
00072 */
00073 typedef enum
00074 {
00075     COMPASS_OK = 0,
00076     COMPASS_ERROR = 1,
00077     COMPASS_TIMEOUT = 2
00078 }
00079 COMPASS_StatusTypeDef;
00080
00081 /**
00082 * @}
00083 */
00084
00085 /** @defgroup STM32L476G_DISCOVERY_COMPASS_E
Exported_Macros Exported Macros
00086 * @{
00087 */
00088
00089 /**
00090 * @}
00091 */
00092
00093 /* Exported functions -----
-----*/
00094 /** @defgroup STM32L476G_DISCOVERY_COMPASS_E
Exported_Functions Exported Functions
00095 * @{
00096 */
00097 /* COMPASS functions */
00098 COMPASS_StatusTypeDef BSP_COMPASS_Init(void
);
00099 void BSP_COMPASS_DeInit(v
oid);

```

```

00100 void                                BSP_COMPASS_LowPower(
void);
00101 void                                BSP_COMPASS_MagGetXYZ
(int16_t *pDataXYZ);
00102 void                                BSP_COMPASS_AccGetXYZ
(int16_t *pDataXYZ);
00103
00104 /**
00105  * @}
00106  */
00107
00108 /**
00109  * @}
00110  */
00111
00112 /**
00113  * @}
00114  */
00115
00116 /**
00117  * @}
00118  */
00119
00120 #ifdef __cplusplus
00121 }
00122 #endif
00123
00124 #endif /* __STM32L476G_DISCOVERY_COMPASS_H */

00125
00126 /***** (C) COPYRIGHT STMicroelectronics *****/

```

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		

## stm32l476g\_discovery\_compass.c

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery_compass.c
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file provides a set of functions needed to manage the E-Compass
00008      *              (ACCELEROMETER + MAGNETOMETER)
00009      *              MEMS LSM303C available on STM32L476G-Discovery
00010      *              board.
00011      ****
00012      * @attention
00013      *
00014      * <h2><center>&copy; COPYRIGHT(c) 2015 STMicroelectronics</center></h2>
00015      *
00016      * Redistribution and use in source and binary forms, with or without modification,
00017      * are permitted provided that the following
```



g conditions are met:

00017 \* 1. Redistributions of source code must  
retain the above copyright notice,

00018 \* this list of conditions and the fol  
lowing disclaimer.

00019 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,

00020 \* this list of conditions and the fol  
lowing disclaimer in the documentation

00021 \* and/or other materials provided wit  
h the distribution.

00022 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors

00023 \* may be used to endorse or promote p  
roducts derived from this software

00024 \* without specific prior written perm  
ission.

00025 \*

00026 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"

00027 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE

00028 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE

00029 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE

00030 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL

00031 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR

00032 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER

00033 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,

00034 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE

00035 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE

## POSSIBILITY OF SUCH DAMAGE.

```
00036      *
00037      ****
*****
00038      */
00039
00040 /* Includes -----
----- */
00041 #include "stm32l476g_discovery.h"
00042 #include "stm32l476g_discovery_compass.h"
00043 #include "../Components/lsm303c/lsm303c.h"
00044 #include <math.h>
00045
00046 /** @addtogroup BSP
00047     * @{
00048     */
00049
00050 /** @addtogroup STM32L476G_DISCOVERY
00051     * @{
00052     */
00053
00054 /** @defgroup STM32L476G_DISCOVERY_COMPASS S
TM32L476G-DISCOVERY COMPASS
00055     * @{
00056     */
00057
00058 /* Private typedef -----
----- */
00059 /** @defgroup STM32L476G_DISCOVERY_COMPASS_P
rivate_Types Private Types
00060     * @{
00061     */
00062 /**
00063     * @}
00064     */
00065
00066 /* Private defines -----
```

```

-----*/
00067 /** @defgroup STM32L476G_DISCOVERY_COMPASS_P
private_Constants Private Constants
00068     * @{
00069     */
00070 /**
00071     * @}
00072     */
00073
00074 /* Private macros -----
-----*/
00075 /** @defgroup STM32L476G_DISCOVERY_COMPASS_P
private_Macros Private Macros
00076     * @{
00077     */
00078 /**
00079     * @}
00080     */
00081
00082 /* Private variables -----
-----*/
00083 /** @defgroup STM32L476G_DISCOVERY_COMPASS_P
private_Variables Private Variables
00084     * @{
00085     */
00086 static ACCELERO_DrvTypeDef *AccelerometerDrv
;
00087 static MAGNETO_DrvTypeDef *MagnettoDrv;
00088
00089 /**
00090     * @}
00091     */
00092
00093 /* Private function prototypes -----
-----*/
00094 /** @addtogroup STM32L476G_DISCOVERY_COMPASS
_Private_FunctionPrototypes Private Functions

```

```

00095     * @{
00096     */
00097 /**
00098     * @}
00099     */
00100
00101 /* Exported functions -----
-----*/
00102 /** @addtogroup STM32L476G_DISCOVERY_COMPASS
    _Exported_Functions
00103     * @{
00104     */
00105 extern void  ACCELERO_IO_DeInit(void);
00106 extern void  MAGNETO_IO_DeInit(void);
00107
00108 /**
00109     * @brief Initialize the COMPASS.
00110     * @retval COMPASS_OK or COMPASS_ERROR
00111     */
00112 COMPASS_StatusTypeDef BSP_COMPASS_Init(void)
00113 {
00114     COMPASS_StatusTypeDef ret = COMPASS_OK;
00115     uint16_t ctrl = 0x0000;
00116     ACCELERO_InitTypeDef LSM303C_InitStructure
;
00117     ACCELERO_FilterConfigTypeDef LSM303C_Filter
Structure;
00118     MAGNETO_InitTypeDef LSM303C_InitStructureM
ag;
00119
00120     if(Lsm303cDrv_accelero.ReadID() != LMS303C
_ACC_ID)
00121     {
00122         ret = COMPASS_ERROR;
00123     }
00124     else
00125     {

```

```

00126      /* Initialize the COMPASS accelerometer
driver structure */
00127      AccelerometerDrv = &Lsm303cDrv_accelero;
00128
00129      /* MEMS configuration -----
-----*/
00130      /* Fill the COMPASS accelerometer struct
ure */
00131      LSM303C_InitStructure.AccOutput_DataRate
= LSM303C_ACC_ODR_50_HZ;
00132      LSM303C_InitStructure.Axes_Enable= LSM30
3C_ACC_AXES_ENABLE;
00133      LSM303C_InitStructure.AccFull_Scale = LS
M303C_ACC_FULLSCALE_2G;
00134      LSM303C_InitStructure.BlockData_Update =
LSM303C_ACC_BDU_CONTINUOUS;
00135      LSM303C_InitStructure.High_Resolution =
LSM303C_ACC_HR_DISABLE;
00136      LSM303C_InitStructure.Communication_Mode
= LSM303C_ACC_SPI_MODE;
00137
00138      /* Configure MEMS: data rate, power mode
, full scale and axes */
00139      ctrl = (LSM303C_InitStructure.High_Reso
lution | LSM303C_InitStructure.AccOutput_DataRate
| \
00140                  LSM303C_InitStructure
.Axes_Enable | LSM303C_InitStructure.BlockData_Upd
ate);
00141
00142      ctrl |= (LSM303C_InitStructure.AccFull_S
cale | LSM303C_InitStructure.Communication_Mode) <
< 8;
00143
00144      /* Configure the COMPASS accelerometer m
ain parameters */
00145      AccelerometerDrv->Init(ctrl);

```

```

00146
00147     /* Fill the COMPASS accelerometer HPF st
ructure */
00148     LSM303C_FilterStructure.HighPassFilter_M
ode_Selection = LSM303C_ACC_HPM_NORMAL_MODE;
00149     LSM303C_FilterStructure.HighPassFilter_C
utoff_Frequency = LSM303C_ACC_DFC1_ODRDIV50;
00150     LSM303C_FilterStructure.HighPassFilter_S
tat = LSM303C_ACC_HPI2S_INT1_DISABLE | LSM303C_ACC
_HPI2S_INT2_DISABLE;
00151
00152     /* Configure MEMS: mode, cutoff frequenc
y, Filter status, Click, AOI1 and AOI2 */
00153     ctrl = (uint8_t) (LSM303C_FilterStructur
e.HighPassFilter_Mode_Selection |\
00154                     LSM303C_FilterStructur
e.HighPassFilter_Cutoff_Frequency|\
00155                     LSM303C_FilterStructur
e.HighPassFilter_Stat);
00156
00157     /* Configure the COMPASS accelerometer L
PF main parameters */
00158     AccelerometerDrv->FilterConfig(ctrl);
00159 }
00160
00161 if(Lsm303cDrv_magneto.ReadID() != LMS303C_
MAG_ID)
00162 {
00163     ret = COMPASS_ERROR;
00164 }
00165 else
00166 {
00167     /* Initialize the COMPASS magnetometer d
river structure */
00168     MagnetoDrv = &Lsm303cDrv_magneto;
00169
00170     /* MEMS configuration -----

```

```

-----*/
00171      /* Fill the COMPASS magnetometer structu
re */
00172      LSM303C_InitStructureMag.Register1 = LSM
303C_MAG_TEMPSENSOR_DISABLE | LSM303C_MAG_OM_XY_UL
TRAHIGH | LSM303C_MAG_ODR_40_HZ;
00173      LSM303C_InitStructureMag.Register2 = LSM
303C_MAG_FS_16_GA | LSM303C_MAG_REBOOT_DEFAULT | L
SM303C_MAG_SOFT_RESET_DEFAULT;
00174      LSM303C_InitStructureMag.Register3 = LSM
303C_MAG_SPI_MODE | LSM303C_MAG_CONFIG_NORMAL_MODE
| LSM303C_MAG_CONTINUOUS_MODE;
00175      LSM303C_InitStructureMag.Register4 = LSM
303C_MAG_OM_Z_ULTRAHIGH | LSM303C_MAG_BLE_LSB;
00176      LSM303C_InitStructureMag.Register5 = LSM
303C_MAG_BDU_CONTINUOUS;
00177      /* Configure the COMPASS magnetometer ma
in parameters */
00178      MagnetoDrv->Init(LSM303C_InitStructureMa
g);
00179  }
00180
00181  return ret;
00182 }
00183
00184 /**
00185  * @brief DeInitialize the COMPASS.
00186  * @retval None.
00187  */
00188 void BSP_COMPASS_DeInit(void)
00189 {
00190     /* DeInitialize the COMPASS accelerometer
& magnetometer IO interfaces */
00191     ACCELERO_IO_DeInit();
00192     MAGNETO_IO_DeInit();
00193 }
00194

```

```

00195 /**
00196  * @brief Put the COMPASS in low power mod
00197  * @retval None
00198  */
00199 void BSP_COMPASS_LowPower(void)
00200 {
00201     /* Put the COMPASS accelerometer in low po
00202     wer mode */
00203     if(AccelerometerDrv->LowPower != NULL)
00204     {
00205         AccelerometerDrv->LowPower();
00206     }
00207     /* Put the COMPASS magnetometer in low pow
00208     er mode */
00209     if(MagnetoDrv->LowPower != NULL)
00210     {
00211         MagnetoDrv->LowPower();
00212     }
00213 }
00214 /**
00215  * @brief Get XYZ acceleration values.
00216  * @param pDataXYZ Pointer on 3 angular ac
00217  * celerations table with
00218  *          pDataXYZ[0] = X axis, p
00219  *          DataXYZ[1] = Y axis, pDataXYZ[2] = Z axis
00220  * @retval None
00221  */
00222 void BSP_COMPASS_AccGetXYZ(int16_t *pDataXYZ
00223 )
00224 {
00225     if(AccelerometerDrv->GetXYZ!= NULL)
00226     {
00227         AccelerometerDrv->GetXYZ(pDataXYZ);
00228     }
00229 }

```



```

00226
00227 /**
00228  * @brief Get XYZ magnetometer values.
00229  * @param pDataXYZ Pointer on 3 magnetomet
er values table with
00230  *                      pDataXYZ[0] = X axis, p
DataXYZ[1] = Y axis, pDataXYZ[2] = Z axis
00231  * @retval None
00232  */
00233 void BSP_COMPASS_MagGetXYZ(int16_t *pDataXYZ
)
00234 {
00235     if(MagnetoDrv->GetXYZ!= NULL)
00236     {
00237         MagnetoDrv->GetXYZ(pDataXYZ);
00238     }
00239 }
00240
00241 /**
00242  * @}
00243  */
00244
00245 /**
00246  * @}
00247  */
00248
00249 /**
00250  * @}
00251  */
00252
00253 /**
00254  * @}
00255  */
00256
00257 /***** (C) COPYRIGHT STMi
croelectronics *****/

```

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_glass\_lcd.h

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      ****
00003      * @file      stm32l476g_discovery_glass_lcd.h
00004
00004      * @author    MCD Application Team
00005      * @version  $VERSION$
00006      * @date      $DATE$
00007      * @brief    Header file for stm32l476g_discovery_glass_lcd.c module.
00008      ****
00008      ****
00009      * @attention
00010      *
00011      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00011      icroelectronics</center></h2>
00012      *
00013      * Redistribution and use in source and binary forms, with or without modification,
00014      * are permitted provided that the following conditions are met:
00015      * 1. Redistributions of source code must
```

retain the above copyright notice,  
00016 \* this list of conditions and the fol  
lowing disclaimer.  
00017 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00018 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00019 \* and/or other materials provided wit  
h the distribution.  
00020 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00021 \* may be used to endorse or promote p  
roducts derived from this software  
00022 \* without specific prior written perm  
ission.  
00023 \*  
00024 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00025 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00026 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00027 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00028 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00029 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00030 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00031 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00032 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00033 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.  
00034 \*

```

00035      ****
00036      */
00037
00038 /* Define to prevent recursive inclusion ---
-----*/
00039 #ifndef __STM32L476G_DISCOVERY_GLASS_LCD_H
00040 #define __STM32L476G_DISCOVERY_GLASS_LCD_H
00041
00042 #ifdef __cplusplus
00043     extern "C" {
00044 #endif
00045
00046 /* Includes -----
-----*/
00047 #include "stm32l476g_discovery.h"
00048
00049 /** @addtogroup BSP
00050     * @{
00051     */
00052
00053 /** @addtogroup STM32L476G_DISCOVERY
00054     * @{
00055     */
00056
00057 /** @addtogroup STM32L476G_DISCOVERY_GLASS_L
CD
00058     * @{
00059     */
00060
00061 /* Exported types -----
-----*/
00062
00063 /** @defgroup STM32L476G_DISCOVERY_GLASS_LCD
_Exported_Types Exported Types
00064     * @{
00065     */

```

```

00066 /**
00067  * @brief LCD Glass digit position
00068  */
00069 typedef enum
00070 {
00071     LCD_DIGIT_POSITION_1 = 0,
00072     LCD_DIGIT_POSITION_2 = 1,
00073     LCD_DIGIT_POSITION_3 = 2,
00074     LCD_DIGIT_POSITION_4 = 3,
00075     LCD_DIGIT_POSITION_5 = 4,
00076     LCD_DIGIT_POSITION_6 = 5,
00077     LCD_DIGIT_MAX_NUMBER = 6,
00078 }DigitPosition_Typedef;
00079 /**
00080  * @brief LCD Glass point
00081  * Warning: element values correspond to LCD Glass point.
00082  */
00083
00084 typedef enum
00085 {
00086     POINT_OFF = 0,
00087     POINT_ON = 1
00088 }Point_Typedef;
00089
00090 /**
00091  * @brief LCD Glass Double point
00092  * Warning: element values correspond to LCD Glass Double point.
00093  */
00094 typedef enum
00095 {
00096     DOUBLEPOINT_OFF = 0,
00097     DOUBLEPOINT_ON = 1
00098 }DoublePoint_Typedef;
00099
00100 /**

```

```

00101     * @brief LCD Glass Battery Level
00102     * element values correspond to different L
CD Glass battery levels
00103     */
00104 typedef enum
00105 {
00106     BATTERYLEVEL_OFF = 0,
00107     BATTERYLEVEL_1_4 = 1,
00108     BATTERYLEVEL_1_2 = 2,
00109     BATTERYLEVEL_3_4 = 3,
00110     BATTERYLEVEL_FULL = 4
00111 }BatteryLevel_Typedef;
00112
00113 /**
00114     * @brief LCD Glass Bar Id
00115     */
00116 typedef enum
00117 {
00118     LCD_BAR_NONE    = 0,
00119     LCD_BAR_0       = (1 << 0),
00120     LCD_BAR_1       = (1 << 1),
00121     LCD_BAR_2       = (1 << 2),
00122     LCD_BAR_3       = (1 << 3)
00123 }BarId_Typedef;
00124 /**
00125     * @}
00126     */
00127
00128 /** @defgroup STM32L476G_DISCOVERY_GLASS_LCD
_Exported_Constants Exported Constants
00129     * @{
00130     */
00131 /**
00132     * @brief LCD digit defintion
00133     */
00134 #define COM_PER_DIGIT_NB          4/*!< Spec
ifies number of COM to address a digit */

```

```

00135 #define SEG_PER_DIGIT_NB          4/*!< Specifies number of SEG to address a digit */
00136
00137 #define LCD_MAP_CHAR_COM0_SEG_1ST_POS (1 <
< LCD_MAP_CHAR_COM0_SEG_1ST_SHIFT)
00138 #define LCD_MAP_CHAR_COM0_SEG_2ND_POS (1 <
< LCD_MAP_CHAR_COM0_SEG_2ND_SHIFT)
00139 #define LCD_MAP_CHAR_COM0_SEG_3RD_POS (1 <
< LCD_MAP_CHAR_COM0_SEG_3RD_SHIFT)
00140 #define LCD_MAP_CHAR_COM0_SEG_4TH_POS (1 <
< LCD_MAP_CHAR_COM0_SEG_4TH_SHIFT)
00141 #define LCD_MAP_CHAR_COM1_SEG_1ST_POS (1 <
< LCD_MAP_CHAR_COM1_SEG_1ST_SHIFT)
00142 #define LCD_MAP_CHAR_COM1_SEG_2ND_POS (1 <
< LCD_MAP_CHAR_COM1_SEG_2ND_SHIFT)
00143 #define LCD_MAP_CHAR_COM1_SEG_3RD_POS (1 <
< LCD_MAP_CHAR_COM1_SEG_3RD_SHIFT)
00144 #define LCD_MAP_CHAR_COM1_SEG_4TH_POS (1 <
< LCD_MAP_CHAR_COM1_SEG_4TH_SHIFT)
00145 #define LCD_MAP_CHAR_COM2_SEG_1ST_POS (1 <
< LCD_MAP_CHAR_COM2_SEG_1ST_SHIFT)
00146 #define LCD_MAP_CHAR_COM2_SEG_2ND_POS (1 <
< LCD_MAP_CHAR_COM2_SEG_2ND_SHIFT)
00147 #define LCD_MAP_CHAR_COM2_SEG_3RD_POS (1 <
< LCD_MAP_CHAR_COM2_SEG_3RD_SHIFT)
00148 #define LCD_MAP_CHAR_COM2_SEG_4TH_POS (1 <
< LCD_MAP_CHAR_COM2_SEG_4TH_SHIFT)
00149 #define LCD_MAP_CHAR_COM3_SEG_1ST_POS (1 <
< LCD_MAP_CHAR_COM3_SEG_1ST_SHIFT)
00150 #define LCD_MAP_CHAR_COM3_SEG_2ND_POS (1 <
< LCD_MAP_CHAR_COM3_SEG_2ND_SHIFT)
00151 #define LCD_MAP_CHAR_COM3_SEG_3RD_POS (1 <
< LCD_MAP_CHAR_COM3_SEG_3RD_SHIFT)
00152 #define LCD_MAP_CHAR_COM3_SEG_4TH_POS (1 <
< LCD_MAP_CHAR_COM3_SEG_4TH_SHIFT)
00153 #define LCD_MAP_CHAR_COM0_SEG_1ST_SHIFT 0x00
000000

```



```

00154 #define LCD_MAP_CHAR_COM0_SEG_2ND_SHIFT 0x00
000001
00155 #define LCD_MAP_CHAR_COM0_SEG_3RD_SHIFT 0x00
000002
00156 #define LCD_MAP_CHAR_COM0_SEG_4TH_SHIFT 0x00
000003
00157 #define LCD_MAP_CHAR_COM1_SEG_1ST_SHIFT 0x00
000004
00158 #define LCD_MAP_CHAR_COM1_SEG_2ND_SHIFT 0x00
000005
00159 #define LCD_MAP_CHAR_COM1_SEG_3RD_SHIFT 0x00
000006
00160 #define LCD_MAP_CHAR_COM1_SEG_4TH_SHIFT 0x00
000007
00161 #define LCD_MAP_CHAR_COM2_SEG_1ST_SHIFT 0x00
000008
00162 #define LCD_MAP_CHAR_COM2_SEG_2ND_SHIFT 0x00
000009
00163 #define LCD_MAP_CHAR_COM2_SEG_3RD_SHIFT 0x00
000010
00164 #define LCD_MAP_CHAR_COM2_SEG_4TH_SHIFT 0x00
000011
00165 #define LCD_MAP_CHAR_COM3_SEG_1ST_SHIFT 0x00
000012
00166 #define LCD_MAP_CHAR_COM3_SEG_2ND_SHIFT 0x00
000013
00167 #define LCD_MAP_CHAR_COM3_SEG_3RD_SHIFT 0x00
000014
00168 #define LCD_MAP_CHAR_COM3_SEG_4TH_SHIFT 0x00
000015
00169
00170 /**
00171  * @brief LCD Digit defines
00172  */
00173 #define LCD_DIGIT1_COM0 LCD_CO
M0
00174 #define LCD_DIGIT1_COM0_SEG_MASK ~(LCD_

```

```

SEG0 | LCD_SEG1 | LCD_SEG22 | LCD_SEG23)
00175 #define LCD_DIGIT1_COM1 LCD_CO
M1
00176 #define LCD_DIGIT1_COM1_SEG_MASK ~(LCD_
SEG0 | LCD_SEG1 | LCD_SEG22 | LCD_SEG23)
00177 #define LCD_DIGIT1_COM2 LCD_CO
M2
00178 #define LCD_DIGIT1_COM2_SEG_MASK ~(LCD_
SEG0 | LCD_SEG1 | LCD_SEG22 | LCD_SEG23)
00179 #define LCD_DIGIT1_COM3 LCD_CO
M3
00180 #define LCD_DIGIT1_COM3_SEG_MASK ~(LCD_
SEG0 | LCD_SEG1 | LCD_SEG22 | LCD_SEG23)
00181
00182 #define LCD_DIGIT2_COM0 LCD_CO
M0
00183 #define LCD_DIGIT2_COM0_SEG_MASK ~(LCD_
SEG2 | LCD_SEG3 | LCD_SEG20 | LCD_SEG21)
00184 #define LCD_DIGIT2_COM1 LCD_CO
M1
00185 #define LCD_DIGIT2_COM1_SEG_MASK ~(LCD_
SEG2 | LCD_SEG3 | LCD_SEG20 | LCD_SEG21)
00186 #define LCD_DIGIT2_COM2 LCD_CO
M2
00187 #define LCD_DIGIT2_COM2_SEG_MASK ~(LCD_
SEG2 | LCD_SEG3 | LCD_SEG20 | LCD_SEG21)
00188 #define LCD_DIGIT2_COM3 LCD_CO
M3
00189 #define LCD_DIGIT2_COM3_SEG_MASK ~(LCD_
SEG2 | LCD_SEG3 | LCD_SEG20 | LCD_SEG21)
00190
00191 #define LCD_DIGIT3_COM0 LCD_CO
M0
00192 #define LCD_DIGIT3_COM0_SEG_MASK ~(LCD_
SEG4 | LCD_SEG5 | LCD_SEG18 | LCD_SEG19)
00193 #define LCD_DIGIT3_COM1 LCD_CO
M1

```

```

00194 #define LCD_DIGIT3_COM1_SEG_MASK ~(LCD_
SEG4 | LCD_SEG5 | LCD_SEG18 | LCD_SEG19)
00195 #define LCD_DIGIT3_COM2 LCD_CO
M2
00196 #define LCD_DIGIT3_COM2_SEG_MASK ~(LCD_
SEG4 | LCD_SEG5 | LCD_SEG18 | LCD_SEG19)
00197 #define LCD_DIGIT3_COM3 LCD_CO
M3
00198 #define LCD_DIGIT3_COM3_SEG_MASK ~(LCD_
SEG4 | LCD_SEG5 | LCD_SEG18 | LCD_SEG19)
00199
00200 #define LCD_DIGIT4_COM0 LCD_CO
M0
00201 #define LCD_DIGIT4_COM0_SEG_MASK ~(LCD_
SEG6 | LCD_SEG17)
00202 #define LCD_DIGIT4_COM0_1 LCD_CO
M0_1
00203 #define LCD_DIGIT4_COM0_1_SEG_MASK ~(LCD_
SEG7 | LCD_SEG16)
00204 #define LCD_DIGIT4_COM1 LCD_CO
M1
00205 #define LCD_DIGIT4_COM1_SEG_MASK ~(LCD_
SEG6 | LCD_SEG17)
00206 #define LCD_DIGIT4_COM1_1 LCD_CO
M1_1
00207 #define LCD_DIGIT4_COM1_1_SEG_MASK ~(LCD_
SEG7 | LCD_SEG16)
00208 #define LCD_DIGIT4_COM2 LCD_CO
M2
00209 #define LCD_DIGIT4_COM2_SEG_MASK ~(LCD_
SEG6 | LCD_SEG17)
00210 #define LCD_DIGIT4_COM2_1 LCD_CO
M2_1
00211 #define LCD_DIGIT4_COM2_1_SEG_MASK ~(LCD_
SEG7 | LCD_SEG16)
00212 #define LCD_DIGIT4_COM3 LCD_CO
M3

```

```

00213 #define LCD_DIGIT4_COM3_SEG_MASK      ~(LCD_
SEG6 | LCD_SEG17)
00214 #define LCD_DIGIT4_COM3_1             LCD_CO
M3_1
00215 #define LCD_DIGIT4_COM3_1_SEG_MASK    ~(LCD_
SEG7 | LCD_SEG16)
00216
00217 #define LCD_DIGIT5_COM0               LCD_CO
M0
00218 #define LCD_DIGIT5_COM0_SEG_MASK      ~(LCD_
SEG9 | LCD_SEG14)
00219 #define LCD_DIGIT5_COM0_1             LCD_CO
M0_1
00220 #define LCD_DIGIT5_COM0_1_SEG_MASK    ~(LCD_
SEG8 | LCD_SEG15)
00221 #define LCD_DIGIT5_COM1               LCD_CO
M1
00222 #define LCD_DIGIT5_COM1_SEG_MASK      ~(LCD_
SEG9 | LCD_SEG14)
00223 #define LCD_DIGIT5_COM1_1             LCD_CO
M1_1
00224 #define LCD_DIGIT5_COM1_1_SEG_MASK    ~(LCD_
SEG8 | LCD_SEG15)
00225 #define LCD_DIGIT5_COM2               LCD_CO
M2
00226 #define LCD_DIGIT5_COM2_SEG_MASK      ~(LCD_
SEG9 | LCD_SEG14)
00227 #define LCD_DIGIT5_COM2_1             LCD_CO
M2_1
00228 #define LCD_DIGIT5_COM2_1_SEG_MASK    ~(LCD_
SEG8 | LCD_SEG15)
00229 #define LCD_DIGIT5_COM3               LCD_CO
M3
00230 #define LCD_DIGIT5_COM3_SEG_MASK      ~(LCD_
SEG9 | LCD_SEG14)
00231 #define LCD_DIGIT5_COM3_1             LCD_CO
M3_1

```

```

00232 #define LCD_DIGIT5_COM3_1_SEG_MASK    ~(LCD_
SEG8 | LCD_SEG15)
00233
00234 #define LCD_DIGIT6_COM0                LCD_CO
M0
00235 #define LCD_DIGIT6_COM0_SEG_MASK      ~(LCD_
SEG10 | LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
00236 #define LCD_DIGIT6_COM1                LCD_CO
M1
00237 #define LCD_DIGIT6_COM1_SEG_MASK      ~(LCD_
SEG10 | LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
00238 #define LCD_DIGIT6_COM2                LCD_CO
M2
00239 #define LCD_DIGIT6_COM2_SEG_MASK      ~(LCD_
SEG10 | LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
00240 #define LCD_DIGIT6_COM3                LCD_CO
M3
00241 #define LCD_DIGIT6_COM3_SEG_MASK      ~(LCD_
SEG10 | LCD_SEG11 | LCD_SEG12 | LCD_SEG13)
00242
00243 /**
00244  * @brief LCD Bar location
00245  */
00246 #define LCD_BAR0_2_COM                  LCD_COM3
00247 #define LCD_BAR1_3_COM                  LCD_COM2
00248 #define LCD_BAR0_SEG                    LCD_SEG11
00249 #define LCD_BAR1_SEG                    LCD_SEG11
00250 #define LCD_BAR2_SEG                    LCD_SEG9
00251 #define LCD_BAR3_SEG                    LCD_SEG9
00252 #define LCD_BAR0_2_SEG_MASK             ~(LCD_BAR0
_SEG | LCD_BAR2_SEG)
00253 #define LCD_BAR1_3_SEG_MASK             ~(LCD_BAR1
_SEG | LCD_BAR3_SEG)
00254
00255 /**
00256  * @brief LCD segments & coms redefinition.
00257  * LCD component segments & coms are not ne

```

cessarily link to MCU segments & coms output.

```
00258    */
```

```
00259 #if defined (USE_STM32L476G_DISCO_REVC) || d  
efined (USE_STM32L476G_DISCO_REVB)
```

```
00260 #define LCD_COM0          MCU_LCD_COM0
```

```
00261 #define LCD_COM0_1        MCU_LCD_COM0_1
```

```
00262 #define LCD_COM1          MCU_LCD_COM1
```

```
00263 #define LCD_COM1_1        MCU_LCD_COM1_1
```

```
00264 #define LCD_COM2          MCU_LCD_COM2
```

```
00265 #define LCD_COM2_1        MCU_LCD_COM2_1
```

```
00266 #define LCD_COM3          MCU_LCD_COM3
```

```
00267 #define LCD_COM3_1        MCU_LCD_COM3_1
```

```
00268 #elif defined (USE_STM32L476G_DISCO_REVA)
```

```
00269 #define LCD_COM0          MCU_LCD_COM5
```

```
00270 #define LCD_COM0_1        MCU_LCD_COM5_1
```

```
00271 #define LCD_COM1          MCU_LCD_COM7
```

```
00272 #define LCD_COM1_1        MCU_LCD_COM7_1
```

```
00273 #define LCD_COM2          MCU_LCD_COM6
```

```
00274 #define LCD_COM2_1        MCU_LCD_COM6_1
```

```
00275 #define LCD_COM3          MCU_LCD_COM4
```

```
00276 #define LCD_COM3_1        MCU_LCD_COM4_1
```

```
00277 #endif
```

```
00278 #define LCD_SEG0          MCU_LCD_SEG4
```

```
00279 #define LCD_SEG1          MCU_LCD_SEG23
```

```
00280 #define LCD_SEG2          MCU_LCD_SEG6
```

```
00281 #define LCD_SEG3          MCU_LCD_SEG13
```

```
00282 #define LCD_SEG4          MCU_LCD_SEG15
```

```
00283 #define LCD_SEG5          MCU_LCD_SEG29
```

```
00284 #define LCD_SEG6          MCU_LCD_SEG31
```

```
00285 #define LCD_SEG7          MCU_LCD_SEG33
```

```
00286 #define LCD_SEG8          MCU_LCD_SEG35
```

```
00287 #define LCD_SEG9          MCU_LCD_SEG25
```

```
00288 #define LCD_SEG10         MCU_LCD_SEG17
```

```
00289 #define LCD_SEG11         MCU_LCD_SEG8
```

```
00290 #define LCD_SEG12         MCU_LCD_SEG9
```

```
00291 #define LCD_SEG13         MCU_LCD_SEG26
```

```
00292 #define LCD_SEG14         MCU_LCD_SEG24
```

00293	#define LCD_SEG15	MCU_LCD_SEG34
00294	#define LCD_SEG16	MCU_LCD_SEG32
00295	#define LCD_SEG17	MCU_LCD_SEG30
00296	#define LCD_SEG18	MCU_LCD_SEG28
00297	#define LCD_SEG19	MCU_LCD_SEG14
00298	#define LCD_SEG20	MCU_LCD_SEG12
00299	#define LCD_SEG21	MCU_LCD_SEG5
00300	#define LCD_SEG22	MCU_LCD_SEG22
00301	#define LCD_SEG23	MCU_LCD_SEG3
00302	#define LCD_SEG0_SHIFT	MCU_LCD_SEG4
	_SHIFT	
00303	#define LCD_SEG1_SHIFT	MCU_LCD_SEG2
	3_SHIFT	
00304	#define LCD_SEG2_SHIFT	MCU_LCD_SEG6
	_SHIFT	
00305	#define LCD_SEG3_SHIFT	MCU_LCD_SEG1
	3_SHIFT	
00306	#define LCD_SEG4_SHIFT	MCU_LCD_SEG1
	5_SHIFT	
00307	#define LCD_SEG5_SHIFT	MCU_LCD_SEG2
	9_SHIFT	
00308	#define LCD_SEG6_SHIFT	MCU_LCD_SEG3
	1_SHIFT	
00309	#define LCD_SEG7_SHIFT	MCU_LCD_SEG3
	3_SHIFT	
00310	#define LCD_SEG8_SHIFT	MCU_LCD_SEG3
	5_SHIFT	
00311	#define LCD_SEG9_SHIFT	MCU_LCD_SEG2
	5_SHIFT	
00312	#define LCD_SEG10_SHIFT	MCU_LCD_SEG1
	7_SHIFT	
00313	#define LCD_SEG11_SHIFT	MCU_LCD_SEG8
	_SHIFT	
00314	#define LCD_SEG12_SHIFT	MCU_LCD_SEG9
	_SHIFT	
00315	#define LCD_SEG13_SHIFT	MCU_LCD_SEG2
	6_SHIFT	

```

00316 #define LCD_SEG14_SHIFT          MCU_LCD_SEG2
4_SHIFT
00317 #define LCD_SEG15_SHIFT          MCU_LCD_SEG3
4_SHIFT
00318 #define LCD_SEG16_SHIFT          MCU_LCD_SEG3
2_SHIFT
00319 #define LCD_SEG17_SHIFT          MCU_LCD_SEG3
0_SHIFT
00320 #define LCD_SEG18_SHIFT          MCU_LCD_SEG2
8_SHIFT
00321 #define LCD_SEG19_SHIFT          MCU_LCD_SEG1
4_SHIFT
00322 #define LCD_SEG20_SHIFT          MCU_LCD_SEG1
2_SHIFT
00323 #define LCD_SEG21_SHIFT          MCU_LCD_SEG5
_SHIFT
00324 #define LCD_SEG22_SHIFT          MCU_LCD_SEG2
2_SHIFT
00325 #define LCD_SEG23_SHIFT          MCU_LCD_SEG3
_SHIFT
00326
00327 /**
00328  * @brief STM32 LCD segments & coms definit
ions.
00329  */
00330 #define MCU_LCD_COM0                LCD_RAM_REGIST
ER0
00331 #define MCU_LCD_COM0_1              LCD_RAM_REGIST
ER1
00332 #define MCU_LCD_COM1                LCD_RAM_REGIST
ER2
00333 #define MCU_LCD_COM1_1              LCD_RAM_REGIST
ER3
00334 #define MCU_LCD_COM2                LCD_RAM_REGIST
ER4
00335 #define MCU_LCD_COM2_1              LCD_RAM_REGIST
ER5

```



00336 #define MCU_LCD_COM3 ER6	LCD_RAM_REGIST
00337 #define MCU_LCD_COM3_1 ER7	LCD_RAM_REGIST
00338 #define MCU_LCD_COM4 ER8	LCD_RAM_REGIST
00339 #define MCU_LCD_COM4_1 ER9	LCD_RAM_REGIST
00340 #define MCU_LCD_COM5 ER10	LCD_RAM_REGIST
00341 #define MCU_LCD_COM5_1 ER11	LCD_RAM_REGIST
00342 #define MCU_LCD_COM6 ER12	LCD_RAM_REGIST
00343 #define MCU_LCD_COM6_1 ER13	LCD_RAM_REGIST
00344 #define MCU_LCD_COM7 ER14	LCD_RAM_REGIST
00345 #define MCU_LCD_COM7_1 ER15	LCD_RAM_REGIST
00346 #define MCU_LCD_SEG0 _SEG0_SHIFT)	(1U << MCU_LCD
00347 #define MCU_LCD_SEG1 _SEG1_SHIFT)	(1U << MCU_LCD
00348 #define MCU_LCD_SEG2 _SEG2_SHIFT)	(1U << MCU_LCD
00349 #define MCU_LCD_SEG3 _SEG3_SHIFT)	(1U << MCU_LCD
00350 #define MCU_LCD_SEG4 _SEG4_SHIFT)	(1U << MCU_LCD
00351 #define MCU_LCD_SEG5 _SEG5_SHIFT)	(1U << MCU_LCD
00352 #define MCU_LCD_SEG6 _SEG6_SHIFT)	(1U << MCU_LCD
00353 #define MCU_LCD_SEG7 _SEG7_SHIFT)	(1U << MCU_LCD
00354 #define MCU_LCD_SEG8	(1U << MCU_LCD

```
_SEG8_SHIFT)
00355 #define MCU_LCD_SEG9          (1U << MCU_LCD
_SEG9_SHIFT)
00356 #define MCU_LCD_SEG10         (1U << MCU_LCD
_SEG10_SHIFT)
00357 #define MCU_LCD_SEG11         (1U << MCU_LCD
_SEG11_SHIFT)
00358 #define MCU_LCD_SEG12         (1U << MCU_LCD
_SEG12_SHIFT)
00359 #define MCU_LCD_SEG13         (1U << MCU_LCD
_SEG13_SHIFT)
00360 #define MCU_LCD_SEG14         (1U << MCU_LCD
_SEG14_SHIFT)
00361 #define MCU_LCD_SEG15         (1U << MCU_LCD
_SEG15_SHIFT)
00362 #define MCU_LCD_SEG16         (1U << MCU_LCD
_SEG16_SHIFT)
00363 #define MCU_LCD_SEG17         (1U << MCU_LCD
_SEG17_SHIFT)
00364 #define MCU_LCD_SEG18         (1U << MCU_LCD
_SEG18_SHIFT)
00365 #define MCU_LCD_SEG19         (1U << MCU_LCD
_SEG19_SHIFT)
00366 #define MCU_LCD_SEG20         (1U << MCU_LCD
_SEG20_SHIFT)
00367 #define MCU_LCD_SEG21         (1U << MCU_LCD
_SEG21_SHIFT)
00368 #define MCU_LCD_SEG22         (1U << MCU_LCD
_SEG22_SHIFT)
00369 #define MCU_LCD_SEG23         (1U << MCU_LCD
_SEG23_SHIFT)
00370 #define MCU_LCD_SEG24         (1U << MCU_LCD
_SEG24_SHIFT)
00371 #define MCU_LCD_SEG25         (1U << MCU_LCD
_SEG25_SHIFT)
00372 #define MCU_LCD_SEG26         (1U << MCU_LCD
_SEG26_SHIFT)
```

```
00373 #define MCU_LCD_SEG27          (1U << MCU_LCD
_SEG27_SHIFT)
00374 #define MCU_LCD_SEG28          (1U << MCU_LCD
_SEG28_SHIFT)
00375 #define MCU_LCD_SEG29          (1U << MCU_LCD
_SEG29_SHIFT)
00376 #define MCU_LCD_SEG30          (1U << MCU_LCD
_SEG30_SHIFT)
00377 #define MCU_LCD_SEG31          (1U << MCU_LCD
_SEG31_SHIFT)
00378 #define MCU_LCD_SEG32          (1U << MCU_LCD
_SEG32_SHIFT)
00379 #define MCU_LCD_SEG33          (1U << MCU_LCD
_SEG33_SHIFT)
00380 #define MCU_LCD_SEG34          (1U << MCU_LCD
_SEG34_SHIFT)
00381 #define MCU_LCD_SEG35          (1U << MCU_LCD
_SEG35_SHIFT)
00382 #define MCU_LCD_SEG36          (1U << MCU_LCD
_SEG36_SHIFT)
00383 #define MCU_LCD_SEG37          (1U << MCU_LCD
_SEG37_SHIFT)
00384 #define MCU_LCD_SEG38          (1U << MCU_LCD
_SEG38_SHIFT)
00385 #define MCU_LCD_SEG0_SHIFT      0
00386 #define MCU_LCD_SEG1_SHIFT      1
00387 #define MCU_LCD_SEG2_SHIFT      2
00388 #define MCU_LCD_SEG3_SHIFT      3
00389 #define MCU_LCD_SEG4_SHIFT      4
00390 #define MCU_LCD_SEG5_SHIFT      5
00391 #define MCU_LCD_SEG6_SHIFT      6
00392 #define MCU_LCD_SEG7_SHIFT      7
00393 #define MCU_LCD_SEG8_SHIFT      8
00394 #define MCU_LCD_SEG9_SHIFT      9
00395 #define MCU_LCD_SEG10_SHIFT     10
00396 #define MCU_LCD_SEG11_SHIFT     11
00397 #define MCU_LCD_SEG12_SHIFT     12
```

```

00398 #define MCU_LCD_SEG13_SHIFT 13
00399 #define MCU_LCD_SEG14_SHIFT 14
00400 #define MCU_LCD_SEG15_SHIFT 15
00401 #define MCU_LCD_SEG16_SHIFT 16
00402 #define MCU_LCD_SEG17_SHIFT 17
00403 #define MCU_LCD_SEG18_SHIFT 18
00404 #define MCU_LCD_SEG19_SHIFT 19
00405 #define MCU_LCD_SEG20_SHIFT 20
00406 #define MCU_LCD_SEG21_SHIFT 21
00407 #define MCU_LCD_SEG22_SHIFT 22
00408 #define MCU_LCD_SEG23_SHIFT 23
00409 #define MCU_LCD_SEG24_SHIFT 24
00410 #define MCU_LCD_SEG25_SHIFT 25
00411 #define MCU_LCD_SEG26_SHIFT 26
00412 #define MCU_LCD_SEG27_SHIFT 27
00413 #define MCU_LCD_SEG28_SHIFT 28
00414 #define MCU_LCD_SEG29_SHIFT 29
00415 #define MCU_LCD_SEG30_SHIFT 30
00416 #define MCU_LCD_SEG31_SHIFT 31
00417 #define MCU_LCD_SEG32_SHIFT 0
00418 #define MCU_LCD_SEG33_SHIFT 1
00419 #define MCU_LCD_SEG34_SHIFT 2
00420 #define MCU_LCD_SEG35_SHIFT 3
00421 #define MCU_LCD_SEG36_SHIFT 4
00422 #define MCU_LCD_SEG37_SHIFT 5
00423 #define MCU_LCD_SEG38_SHIFT 6
00424 #define MCU_LCD_SEG39_SHIFT 7
00425 #define MCU_LCD_SEG40_SHIFT 8
00426 #define MCU_LCD_SEG41_SHIFT 9
00427 #define MCU_LCD_SEG42_SHIFT 10
00428 #define MCU_LCD_SEG43_SHIFT 11
00429
00430 /**
00431  * @brief LCD Pins definition.
00432  */
00433 #if defined (USE_STM32L476G_DISCO_REVC) || d
efined (USE_STM32L476G_DISCO_REVB)

```

```

00434 #define LCD_GPIO_BANKA_PINS (GPIO_PIN_6 | G
PIO_PIN_7 | GPIO_PIN_8 | \
00435 GPIO_PIN_9 | G
PIO_PIN_10 | GPIO_PIN_15)
00436 #define LCD_GPIO_BANKB_PINS (GPIO_PIN_0 | G
PIO_PIN_1 | GPIO_PIN_4 | \
00437 GPIO_PIN_5 | G
PIO_PIN_9 | GPIO_PIN_12 | \
00438 GPIO_PIN_13 |
GPIO_PIN_14 | GPIO_PIN_15)
00439 #define LCD_GPIO_BANKC_PINS (GPIO_PIN_3 | G
PIO_PIN_4 | GPIO_PIN_5 | \
00440 GPIO_PIN_6 | G
PIO_PIN_7 | GPIO_PIN_8)
00441 #define LCD_GPIO_BANKD_PINS (GPIO_PIN_8 | G
PIO_PIN_9 | GPIO_PIN_10 | \
00442 GPIO_PIN_11 |
GPIO_PIN_12 | GPIO_PIN_13 | \
00443 GPIO_PIN_14 |
GPIO_PIN_15)
00444 #elif defined (USE_STM32L476G_DISCO_REVA)
00445 #define LCD_GPIO_BANKA_PINS (GPIO_PIN_6 | G
PIO_PIN_7 | GPIO_PIN_15)
00446 #define LCD_GPIO_BANKB_PINS (GPIO_PIN_0 | G
PIO_PIN_1 | GPIO_PIN_4 | \
00447 GPIO_PIN_5 | G
PIO_PIN_12 | GPIO_PIN_13 | \
00448 GPIO_PIN_14 |
GPIO_PIN_15)
00449 #define LCD_GPIO_BANKC_PINS (GPIO_PIN_3 | G
PIO_PIN_4 | GPIO_PIN_5 | \
00450 GPIO_PIN_6 | G
PIO_PIN_7 | GPIO_PIN_8 | \
00451 GPIO_PIN_10 |
GPIO_PIN_11 | GPIO_PIN_12)
00452 #define LCD_GPIO_BANKD_PINS (GPIO_PIN_2 | G
PIO_PIN_8 | GPIO_PIN_9 | \

```

```

00453                                     GPIO_PIN_10 |
GPIO_PIN_11 | GPIO_PIN_12 | \
00454                                     GPIO_PIN_13 |
GPIO_PIN_14 | GPIO_PIN_15)
00455 #endif
00456
00457 /* Define for scrolling sentences*/
00458 #define SCROLL_SPEED_HIGH      150
00459 #define SCROLL_SPEED_MEDIUM    300
00460 #define SCROLL_SPEED_LOW       450
00461
00462 #define DOT                     ((uint16_t) 0x
8000 ) /* for add decimal point in string */
00463 #define DOUBLE_DOT              ((uint16_t) 0x
4000) /* for add decimal point in string */
00464
00465 /* code for '(' character */
00466 #define C_OPENPARMAP            ((uint16_t) 0x
0028)
00467
00468 /* code for ')' character */
00469 #define C_CLOSEPARMAP           ((uint16_t) 0x
0011)
00470
00471 /* code for 'd' character */
00472 #define C_DMAP                  ((uint16_t) 0x
f300)
00473
00474 /* code for 'm' character */
00475 #define C_MMAP                  ((uint16_t) 0x
b210)
00476
00477 /* code for 'n' character */
00478 #define C_NMAP                  ((uint16_t) 0x
2210)
00479
00480 /* code for '0' character */

```

```

00481 #define C_UMAP                ((uint16_t) 0x
6084)
00482
00483 /* constant code for '*' character */
00484 #define C_STAR                ((uint16_t) 0x
A0DD)
00485
00486 /* constant code for '-' character */
00487 #define C_MINUS                ((uint16_t) 0x
A000)
00488
00489 /* constant code for '+' character */
00490 #define C_PLUS                ((uint16_t) 0x
A014)
00491
00492 /* constant code for '/' */
00493 #define C_SLATCH                ((uint16_t) 0x
00c0)
00494
00495 /* constant code for % */
00496 #define C_PERCENT_1            ((uint16_t) 0x
ec00)
00497
00498 /* constant code for small o */
00499 #define C_PERCENT_2            ((uint16_t) 0x
b300)
00500
00501 #define C_FULL                ((uint16_t) 0x
ffdd)
00502
00503 /**
00504  * @}
00505  */
00506
00507 /* Exported functions -----
----- */
00508

```

```

00509 /** @defgroup STM32L476G_DISCOVERY_LCD_Export
ted_Functions Exported Functions
00510     * @{
00511     */
00512 void BSP_LCD_GLASS_Init(void);
00513 void BSP_LCD_GLASS_DeInit(void);
00514 void BSP_LCD_GLASS_BlinkConfig(uint32_t Blink
kMode, uint32_t BlinkFrequency);
00515 void BSP_LCD_GLASS_Contrast(uint32_t Contrast);
00516 void BSP_LCD_GLASS_DisplayChar(uint8_t* ch,
Point_Typedef Point, DoublePoint_Typedef Column, D
igitPosition_Typedef Position);
00517 void BSP_LCD_GLASS_DisplayString(uint8_t* ptr);
00518 void BSP_LCD_GLASS_DisplayStrDeci(uint16_t*
ptr);
00519 void BSP_LCD_GLASS_ScrollSentence(uint8_t* p
tr, uint16_t nScroll, uint16_t ScrollSpeed);
00520 void BSP_LCD_GLASS_DisplayBar(uint32_t BarId
);
00521 void BSP_LCD_GLASS_ClearBar(uint32_t BarId);
00522 void BSP_LCD_GLASS_BarLevelConfig(uint8_t Ba
rLevel);
00523 void BSP_LCD_GLASS_Clear(void);
00524 /**
00525     * @}
00526     */
00527
00528 /**
00529     * @}
00530     */
00531
00532 /**
00533     * @}
00534     */
00535

```



```
00536 /**
00537  * @}
00538 */
00539
00540 #ifdef __cplusplus
00541 }
00542 #endif
00543
00544 #endif /* __STM32L476G_DISCOVERY_GLASS_LCD_H
    */
00545
00546 /***** (C) COPYRIGHT STMicroelectronics *****/
00547 *****/
```

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		

## stm32l476g\_discovery\_glass\_lcd.c

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      ****
00003      * @file      stm32l476g_discovery_glass_lcd.c
00004
00004      * @author    MCD Application Team
00005      * @version  $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file provides a set of fun
00008                  ctions needed to manage the
00008      *           LCD Glass driver for the STM32L
00009                  476G-Discovery board.
00009      ****
00009      ****
00010      * @attention
00011      *
00012      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00013                  icroelectronics</center></h2>
00013      *
00014      * Redistribution and use in source and bin
00015                  ary forms, with or without modification,
00015      * are permitted provided that the followin
```

g conditions are met:

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,

00017 \* this list of conditions and the fol  
lowing disclaimer.

00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,

00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation

00020 \* and/or other materials provided wit  
h the distribution.

00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors

00022 \* may be used to endorse or promote p  
roducts derived from this software

00023 \* without specific prior written perm  
ission.

00024 \*

00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"

00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE

00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE

00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE

00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL

00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR

00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER

00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,

00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE

00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE

## POSSIBILITY OF SUCH DAMAGE.

```
00035      *
00036      ****
00037      */
00038
00039 /* Includes -----
-----*/
00040 #include "stm32l476g_discovery_glass_lcd.h"
00041
00042 /** @addtogroup BSP
00043     * @{
00044     */
00045
00046 /** @addtogroup STM32L476G_DISCOVERY
00047     * @{
00048     */
00049
00050 /** @defgroup STM32L476G_DISCOVERY_GLASS_LCD
00051     STM32L476G-DISCOVERY GLASS LCD
00052     * @brief This file includes the LCD Glass
00053     driver for LCD Module of
00054     STM32L476G-DISCOVERY board.
00055     * @{
00056     */
00057
00058 /** @defgroup STM32L476G_DISCOVERY_GLASS_LCD
00059     _Private_Constants Private Constants
00060     * @{
00061     */
00062 #define ASCII_CHAR_0                                0x30
00063 /* 0 */
00064 #define ASCII_CHAR_AT_SYMBOL                        0x40
00065 /* @ */
```

```

00063 #define ASCII_CHAR_LEFT_OPEN_BRACKET  0x5B
/* [ */
00064 #define ASCII_CHAR_APOSTROPHE           0x60
/* ` */
00065 #define ASCII_CHAR_LEFT_OPEN_BRACE       0x7B
/* ( */
00066 /**
00067  * @}
00068  */
00069
00070 /* Private variables -----
-----*/
00071
00072 /** @defgroup STM32L476G_DISCOVERY_GLASS_LCD
_Private_Variables Private Variables
00073  * @{
00074  */
00075
00076 /* this variable can be used for accelerate
the scrolling exit when push user button */
00077 __IO uint8_t bLCDGlass_KeyPressed = 0;
00078
00079 /**
00080  @verbatim
00081  =====
=====
00082                                GLASS LCD MAPP
ING
00083  =====
=====
00084 LCD allows to display informations on six 14
-segment digits and 4 bars:
00085
00086      1          2          3          4          5          6
00087  -----  -----  -----  -----  -----  -----
-
00088  |\|/| o |\|/| o |\|/| o |\|/| o |\|/|  |\|/

```

```

| BAR3
00089  --  --  --  --  --  --  --  --  --  --
- BAR2
00090  | / | \ | o | / | \ | o | / | \ | o | / | \ | o | / | \ |
| BAR1
00091  ----- * ----- * ----- * ----- * -----
- BAR0
00092
00093 LCD segment mapping:
00094 -----
00095  -----A-----
00096  | \   |   / |   COL  | _ |
00097  F H   J   K B
00098  |   \ |   /   |
00099  --G-- --M--   COL  | _ |
00100  |   / |   \   |
00101  E Q   P   N C
00102  | /   |   \ |
00103  -----D-----   DP   | _ |
00104
00105 An LCD character coding is based on the following matrix:
00106 COM           0     1     2     3
00107 SEG(n)        { E , D , P ,   N   }
00108 SEG(n+1)      { M , C , COL , DP }
00109 SEG(23-n-1)   { B , A , K ,   J   }
00110 SEG(23-n)     { G , F , Q ,   H   }
00111 with n positive odd number.
00112
00113 The character 'A' for example is:
00114 -----
00115 LSB   { 1 , 0 , 0 , 0 }
00116       { 1 , 1 , 0 , 0 }
00117       { 1 , 1 , 0 , 0 }
00118 MSB   { 1 , 1 , 0 , 0 }
00119 -----
00120 'A' = F     E     0     0 hexa

```

```

00121
00122     @endverbatim
00123 */
00124
00125 LCD_HandleTypeDef LCDHandle;
00126
00127 /* Constant table for cap characters 'A' -->
    'Z' */
00128 const uint16_t CapLetterMap[26]=
00129     {
00130         /* A      B      C      D      E
           F      G      H      I */
00131         0xFE00, 0x6714, 0x1D00, 0x4714, 0x9D
00, 0x9C00, 0x3F00, 0xFA00, 0x0014,
00132         /* J      K      L      M      N
           O      P      Q      R */
00133         0x5300, 0x9841, 0x1900, 0x5A48, 0x5A
09, 0x5F00, 0xFC00, 0x5F01, 0xFC01,
00134         /* S      T      U      V      W
           X      Y      Z */
00135         0xAF00, 0x0414, 0x5b00, 0x18C0, 0x5A
81, 0x00C9, 0x0058, 0x05C0
00136     };
00137
00138 /* Constant table for number '0' --> '9' */
00139 const uint16_t NumberMap[10]=
00140     {
00141         /* 0      1      2      3      4
           5      6      7      8      9 */
00142         0x5F00, 0x4200, 0xF500, 0x6700, 0xEa00, 0
xAF00, 0xBF00, 0x04600, 0xFF00, 0xEF00
00143     };
00144
00145 uint32_t Digit[4];        /* Digit frame buffer
    */
00146
00147 /* LCD BAR status: To save the bar setting a

```

```

fter writing in LCD RAM memory */
00148 uint8_t LCDBar = BATTERYLEVEL_FULL;
00149
00150 /**
00151  * @}
00152  */
00153
00154 /** @defgroup STM32L476G_DISCOVERY_LCD_Private_Func
te_Functions Private Functions
00155  * @{
00156  */
00157 static void Convert(uint8_t* Char, Point_Typ
edef Point, DoublePoint_Typedef Colon);
00158 static void WriteChar(uint8_t* ch, Point_Typ
edef Point, DoublePoint_Typedef Colon, DigitPositi
on_Typedef Position);
00159 static void LCD_MspInit(LCD_HandleTypeDef *h
lcd);
00160 static void LCD_MspDeInit(LCD_HandleTypeDef
*hlcd);
00161
00162 /**
00163  * @}
00164  */
00165
00166 /** @addtogroup STM32L476G_DISCOVERY_LCD_Exp
orted_Functions
00167  * @{
00168  */
00169
00170 /**
00171  * @brief Initialize the LCD GLASS relativ
e GPIO port IOs and LCD peripheral.
00172  * @retval None
00173  */
00174 void BSP_LCD_GLASS_Init(void)
00175 {

```



```

00176     LCDHandle.Instance                = LCD;
00177     LCDHandle.Init.Prescaler           = LCD_PRESCALER_1;
00178     LCDHandle.Init.Divider             = LCD_DIVIDER_31;
00179     #if defined (USE_STM32L476G_DISCO_REVC) || defined (USE_STM32L476G_DISCO_REVB)
00180     LCDHandle.Init.Duty                = LCD_DUTY_1_4;
00181     #elif defined (USE_STM32L476G_DISCO_REVA)
00182     LCDHandle.Init.Duty                = LCD_DUTY_1_8;
00183     #endif
00184     LCDHandle.Init.Bias                 = LCD_BIAS_1_3;
00185     LCDHandle.Init.VoltageSource        = LCD_VOLTAGESOURCE_INTERNAL;
00186     LCDHandle.Init.Contrast             = LCD_CONTRAST_LEVEL_5;
00187     LCDHandle.Init.DeadTime             = LCD_DEADTIME_0;
00188     LCDHandle.Init.PulseOnDuration      = LCD_PULSEON_DURATION_4;
00189     LCDHandle.Init.HighDrive            = LCD_HIGHDRIVE_DISABLE;
00190     LCDHandle.Init.BlinkMode            = LCD_BLINKMODE_OFF;
00191     LCDHandle.Init.BlinkFrequency       = LCD_BLINKFREQUENCY_DIV32;
00192     LCDHandle.Init.MuxSegment           = LCD_MUXSEGMENT_DISABLE;
00193
00194     /* Initialize the LCD */
00195     LCD_MspInit(&LCDHandle);
00196     HAL_LCD_Init(&LCDHandle);
00197
00198     BSP_LCD_GLASS_Clear();

```

```

00199 }
00200
00201 /**
00202  * @brief DeInitialize the LCD GLASS relative GPIO port IOs and LCD peripheral.
00203  * @retval None
00204  */
00205 void BSP_LCD_GLASS_DeInit(void)
00206 {
00207     /* De-Initialize the LCD */
00208     LCD_MspDeInit(&LCDHandle);
00209     HAL_LCD_DeInit(&LCDHandle);
00210 }
00211
00212
00213 /**
00214  * @brief Configure the LCD Blink mode and Blink frequency.
00215  * @param BlinkMode: specifies the LCD blink mode.
00216  * This parameter can be one of the following values:
00217  * @arg LCD_BLINKMODE_OFF: Blink disabled
00218  * @arg LCD_BLINKMODE_SEG0_COM0: Blink enabled on SEG[0], COM[0] (1 pixel)
00219  * @arg LCD_BLINKMODE_SEG0_ALLCOM: Blink enabled on SEG[0], all COM (up to 8
00220  * pixels according to the programmed duty)
00221  * @arg LCD_BLINKMODE_ALLSEG_ALLCOM: Blink enabled on all SEG and all COM
00222  * (all pixels)
00223  * @param BlinkFrequency: specifies the LCD blink frequency.
00224  * @arg LCD_BLINKFREQUENCY_DIV8: The

```

```

    Blink frequency = fLcd/8
00225     *      @arg LCD_BLINKFREQUENCY_DIV16:    The
    Blink frequency = fLcd/16
00226     *      @arg LCD_BLINKFREQUENCY_DIV32:    The
    Blink frequency = fLcd/32
00227     *      @arg LCD_BLINKFREQUENCY_DIV64:    The
    Blink frequency = fLcd/64
00228     *      @arg LCD_BLINKFREQUENCY_DIV128:   The
    Blink frequency = fLcd/128
00229     *      @arg LCD_BLINKFREQUENCY_DIV256:   The
    Blink frequency = fLcd/256
00230     *      @arg LCD_BLINKFREQUENCY_DIV512:   The
    Blink frequency = fLcd/512
00231     *      @arg LCD_BLINKFREQUENCY_DIV1024:  The
    Blink frequency = fLcd/1024
00232     * @retval None
00233     */
00234 void BSP_LCD_GLASS_BlinkConfig(uint32_t Blink
kMode, uint32_t BlinkFrequency)
00235 {
00236     __HAL_LCD_BLINK_CONFIG(&LCDHandle, BlinkMo
de, BlinkFrequency);
00237 }
00238
00239 /**
00240     * @brief Configure the LCD contrast.
00241     * @param Contrast: specifies the LCD cont
rast value.
00242     * This parameter can be one of the follo
wing values:
00243     *      @arg LCD_CONTRASTLEVEL_0: Maximum Vo
ltage = 2.60V
00244     *      @arg LCD_CONTRASTLEVEL_1: Maximum Vo
ltage = 2.73V
00245     *      @arg LCD_CONTRASTLEVEL_2: Maximum Vo
ltage = 2.86V
00246     *      @arg LCD_CONTRASTLEVEL_3: Maximum Vo

```

```

ltage = 2.99V
00247      *      @arg LCD_CONTRASTLEVEL_4: Maximum Vo
ltage = 3.12V
00248      *      @arg LCD_CONTRASTLEVEL_5: Maximum Vo
ltage = 3.25V
00249      *      @arg LCD_CONTRASTLEVEL_6: Maximum Vo
ltage = 3.38V
00250      *      @arg LCD_CONTRASTLEVEL_7: Maximum Vo
ltage = 3.51V
00251      * @retval None
00252      */
00253 void BSP_LCD_GLASS_Contrast(uint32_t Contrast)
00254 {
00255     __HAL_LCD_CONTRAST_CONFIG(&LCDHandle, Contrast);
00256 }
00257
00258 /**
00259  * @brief Display one or several bar in LCD
    frame buffer.
00260  * @param BarId: specifies the LCD GLASS Bar to display
00261  *      This parameter can be one of the following values:
00262  *      @arg BAR0: LCD GLASS Bar 0
00263  *      @arg BAR1: LCD GLASS Bar 1
00264  *      @arg BAR2: LCD GLASS Bar 2
00265  *      @arg BAR3: LCD GLASS Bar 3
00266  * @retval None
00267  */
00268 void BSP_LCD_GLASS_DisplayBar(uint32_t BarId)
00269 {
00270     uint32_t position = 0;
00271
00272     /* Check which bar is selected */

```

```

00273     while ((BarId) >> position)
00274     {
00275         /* Check if current bar is selected */
00276         switch(BarId & (1 << position))
00277         {
00278             /* Bar 0 */
00279             case LCD_BAR_0:
00280                 /* Set BAR0 */
00281                 HAL_LCD_Write(&LCDHandle, LCD_BAR0_2
00282                               _COM, ~(LCD_BAR0_SEG), LCD_BAR0_SEG);
00283                 break;
00284             /* Bar 1 */
00285             case LCD_BAR_1:
00286                 /* Set BAR1 */
00287                 HAL_LCD_Write(&LCDHandle, LCD_BAR1_3
00288                               _COM, ~(LCD_BAR1_SEG), LCD_BAR1_SEG);
00289                 break;
00290             /* Bar 2 */
00291             case LCD_BAR_2:
00292                 /* Set BAR2 */
00293                 HAL_LCD_Write(&LCDHandle, LCD_BAR0_2
00294                               _COM, ~(LCD_BAR2_SEG), LCD_BAR2_SEG);
00295                 break;
00296             /* Bar 3 */
00297             case LCD_BAR_3:
00298                 /* Set BAR3 */
00299                 HAL_LCD_Write(&LCDHandle, LCD_BAR1_3
00300                               _COM, ~(LCD_BAR3_SEG), LCD_BAR3_SEG);
00301                 break;
00302             default:
00303                 break;
00304         }
00305         position++;

```

```

00306     }
00307
00308     /* Update the LCD display */
00309     HAL_LCD_UpdateDisplayRequest(&LCDHandle);
00310 }
00311
00312 /**
00313  * @brief Clear one or several bar in LCD frame buffer.
00314  * @param BarId: specifies the LCD GLASS Bar to display
00315  *        This parameter can be combination of one of the following values:
00316  *        @arg LCD_BAR_0: LCD GLASS Bar 0
00317  *        @arg LCD_BAR_1: LCD GLASS Bar 1
00318  *        @arg LCD_BAR_2: LCD GLASS Bar 2
00319  *        @arg LCD_BAR_3: LCD GLASS Bar 3
00320  * @retval None
00321  */
00322 void BSP_LCD_GLASS_ClearBar(uint32_t BarId)
00323 {
00324     uint32_t position = 0;
00325
00326     /* Check which bar is selected */
00327     while ((BarId) >> position)
00328     {
00329         /* Check if current bar is selected */
00330         switch(BarId & (1 << position))
00331         {
00332             /* Bar 0 */
00333             case LCD_BAR_0:
00334                 /* Set BAR0 */
00335                 HAL_LCD_Write(&LCDHandle, LCD_BAR0_2_COM, ~(LCD_BAR0_SEG) , 0);
00336                 break;
00337
00338             /* Bar 1 */

```

```

00339         case LCD_BAR_1:
00340             /* Set BAR1 */
00341             HAL_LCD_Write(&LCDHandle, LCD_BAR1_3
00342                 _COM, ~(LCD_BAR1_SEG), 0);
00343             break;
00344         /* Bar 2 */
00345         case LCD_BAR_2:
00346             /* Set BAR2 */
00347             HAL_LCD_Write(&LCDHandle, LCD_BAR0_2
00348                 _COM, ~(LCD_BAR2_SEG), 0);
00349             break;
00350         /* Bar 3 */
00351         case LCD_BAR_3:
00352             /* Set BAR3 */
00353             HAL_LCD_Write(&LCDHandle, LCD_BAR1_3
00354                 _COM, ~(LCD_BAR3_SEG), 0);
00355             break;
00356         default:
00357             break;
00358     }
00359     position++;
00360 }
00361
00362 /* Update the LCD display */
00363 HAL_LCD_UpdateDisplayRequest(&LCDHandle);
00364 }
00365
00366 /**
00367  * @brief Configure the bar level on LCD by
00368  *        writing bar value in LCD frame buffer.
00369  * @param BarLevel: specifies the LCD GLASS
00370  *        Battery Level.
00371  *
00372  * This parameter can be one of the fol
00373  * lowing values:

```

```

00370      *      @arg BATTERYLEVEL_OFF: LCD GLASS Bat
tery Empty
00371      *      @arg BATTERYLEVEL_1_4: LCD GLASS Bat
tery 1/4 Full
00372      *      @arg BATTERYLEVEL_1_2: LCD GLASS Bat
tery 1/2 Full
00373      *      @arg BATTERYLEVEL_3_4: LCD GLASS Bat
tery 3/4 Full
00374      *      @arg BATTERYLEVEL_FULL: LCD GLASS Ba
ttery Full
00375      * @retval None
00376      */
00377 void BSP_LCD_GLASS_BarLevelConfig(uint8_t Ba
rLevel)
00378 {
00379     switch (BarLevel)
00380     {
00381         /* BATTERYLEVEL_OFF */
00382         case BATTERYLEVEL_OFF:
00383             /* Set BAR0 & BAR2 off */
00384             HAL_LCD_Write(&LCDHandle, LCD_BAR0_2_COM
, ~(LCD_BAR0_SEG | LCD_BAR2_SEG), 0);
00385             /* Set BAR1 & BAR3 off */
00386             HAL_LCD_Write(&LCDHandle, LCD_BAR1_3_COM
, ~(LCD_BAR1_SEG | LCD_BAR3_SEG), 0);
00387             LCDBar = BATTERYLEVEL_OFF;
00388             break;
00389
00390         /* BARLEVEL 1/4 */
00391         case BATTERYLEVEL_1_4:
00392             /* Set BAR0 on & BAR2 off */
00393             HAL_LCD_Write(&LCDHandle, LCD_BAR0_2_COM
, ~(LCD_BAR0_SEG | LCD_BAR2_SEG), LCD_BAR0_SEG);
00394             /* Set BAR1 & BAR3 off */
00395             HAL_LCD_Write(&LCDHandle, LCD_BAR1_3_COM
, ~(LCD_BAR1_SEG | LCD_BAR3_SEG), 0);
00396             LCDBar = BATTERYLEVEL_1_4;

```



```
00397         break;
00398
00399     /* BARLEVEL 1/2 */
00400     case BATTERYLEVEL_1_2:
00401         /* Set BAR0 on & BAR2 off */
00402         HAL_LCD_Write(&LCDHandle, LCD_BAR0_2_COM
, ~(LCD_BAR0_SEG | LCD_BAR2_SEG), LCD_BAR0_SEG);
00403         /* Set BAR1 on & BAR3 off */
00404         HAL_LCD_Write(&LCDHandle, LCD_BAR1_3_COM
, ~(LCD_BAR1_SEG | LCD_BAR3_SEG), LCD_BAR1_SEG);
00405         LCDBar = BATTERYLEVEL_1_2;
00406         break;
00407
00408     /* Battery Level 3/4 */
00409     case BATTERYLEVEL_3_4:
00410         /* Set BAR0 & BAR2 on */
00411         HAL_LCD_Write(&LCDHandle, LCD_BAR0_2_COM
, ~(LCD_BAR0_SEG | LCD_BAR2_SEG), (LCD_BAR0_SEG |
LCD_BAR2_SEG));
00412         /* Set BAR1 on & BAR3 off */
00413         HAL_LCD_Write(&LCDHandle, LCD_BAR1_3_COM
, ~(LCD_BAR1_SEG | LCD_BAR3_SEG), LCD_BAR1_SEG);
00414         LCDBar = BATTERYLEVEL_3_4;
00415         break;
00416
00417     /* BATTERYLEVEL_FULL */
00418     case BATTERYLEVEL_FULL:
00419         /* Set BAR0 & BAR2 on */
00420         HAL_LCD_Write(&LCDHandle, LCD_BAR0_2_COM
, ~(LCD_BAR0_SEG | LCD_BAR2_SEG), (LCD_BAR0_SEG |
LCD_BAR2_SEG));
00421         /* Set BAR1 on & BAR3 on */
00422         HAL_LCD_Write(&LCDHandle, LCD_BAR1_3_COM
, ~(LCD_BAR1_SEG | LCD_BAR3_SEG), (LCD_BAR1_SEG |
LCD_BAR3_SEG));
00423         LCDBar = BATTERYLEVEL_FULL;
00424         break;
```

```

00425
00426     default:
00427         break;
00428     }
00429
00430     /* Update the LCD display */
00431     HAL_LCD_UpdateDisplayRequest(&LCDHandle);
00432 }
00433
00434 /**
00435  * @brief Write a character in the LCD RAM
00436  *        buffer.
00437  * @param ch: The character to display.
00438  * @param Point: A point to add in front of
00439  *        char.
00440  *
00441  *          This parameter can be one of the
00442  *          following values:
00443  *
00444  *          @arg POINT_OFF: No point to
00445  *          add in front of char.
00446  *
00447  *          @arg POINT_ON: Add a point
00448  *          in front of char.
00449  * @param Colon: Flag indicating if a colon
00450  *        character has to be added in front
00451  *        of displayed character.
00452  *
00453  *          This parameter can be one of the
00454  *          following values:
00455  *
00456  *          @arg DOUBLEPOINT_OFF: No colon
00457  *          to add in back of char.
00458  *
00459  *          @arg DOUBLEPOINT_ON: Add an
00460  *          colon in back of char.
00461  * @param Position: Position in the LCD of
00462  *        the character to write.
00463  *
00464  *          This parameter can be
00465  *          any value in range [1:6].
00466  * @retval None
00467  * @note Required preconditions: The LCD

```

```

should be cleared before to start the
00450      *           write operation.
00451      */
00452 void BSP_LCD_GLASS_DisplayChar(uint8_t* ch,
Point_Typedef Point, DoublePoint_Typedef Colon, Di
gitPosition_Typedef Position)
00453 {
00454     WriteChar(ch, Point, Colon, Position);
00455
00456     /* Update the LCD display */
00457     HAL_LCD_UpdateDisplayRequest(&LCDHandle);
00458 }
00459
00460 /**
00461  * @brief Write a character string in the
LCD RAM buffer.
00462  * @param ptr: Pointer to string to displa
y on the LCD Glass.
00463  * @retval None
00464  */
00465 void BSP_LCD_GLASS_DisplayString(uint8_t* pt
r)
00466 {
00467     DigitPosition_Typedef position = LCD_DIGIT
_POSITION_1;
00468
00469     /* Send the string character by character
on LCD */
00470     while ((*ptr != 0) & (position <= LCD_DIGI
T_POSITION_6))
00471     {
00472         /* Write one character on LCD */
00473         WriteChar(ptr, POINT_OFF, DOUBLEPOINT_OFF
, position);
00474
00475         /* Point on the next character */
00476         ptr++;

```

```

00477
00478     /* Increment the character counter */
00479     position++;
00480 }
00481 /* Update the LCD display */
00482 HAL_LCD_UpdateDisplayRequest(&LCDHandle);
00483 }
00484
00485 /**
00486  * @brief Write a character string with de
00487  * cimal point in the LCD RAM buffer.
00488  * @param ptr: Pointer to string to displa
00489  * y on the LCD Glass.
00490  * @retval None
00491  * @note Required preconditions: Char is AS
00492  * CCI value "0Red" with decimal point or Colon flag
00493  */
00494 void BSP_LCD_GLASS_DisplayStrDeci(uint16_t*
00495 ptr)
00496 {
00497     DigitPosition_Typedef index = LCD_DIGIT_PO
00498     SITION_1;
00499     uint8_t tmpchar = 0;
00500
00501     /* Send the string character by character
00502     on LCD */
00503     while((*ptr != 0) & (index <= LCD_DIGIT_PO
00504     SITION_6))
00505     {
00506         tmpchar = (*ptr) & 0x00FF;
00507
00508         switch((*ptr) & 0xF000)
00509         {
00510             case DOT:
00511                 /* Write one character on LCD with dec
00512                 imal point */
00513                 WriteChar(&tmpchar, POINT_ON, DOUBLEPO

```

```

INT_OFF, index);
00506         break;
00507         case DOUBLE_DOT:
00508             /* Write one character on LCD with dec
imal point */
00509             WriteChar(&tmpchar, POINT_OFF, DOUBLEP
OINT_ON, index);
00510             break;
00511         default:
00512             WriteChar(&tmpchar, POINT_OFF, DOUBLEP
OINT_OFF, index);
00513             break;
00514     }/* Point on the next character */
00515     ptr++;
00516
00517     /* Increment the character counter */
00518     index++;
00519 }
00520 /* Update the LCD display */
00521 HAL_LCD_UpdateDisplayRequest(&LCDHandle);
00522 }
00523
00524 /**
00525  * @brief Clear the whole LCD RAM buffer.
00526  * @retval None
00527  */
00528 void BSP_LCD_GLASS_Clear(void)
00529 {
00530     HAL_LCD_Clear(&LCDHandle);
00531 }
00532
00533 /**
00534  * @brief Display a string in scrolling mo
de
00535  * @param ptr: Pointer to string to displa
y on the LCD Glass.
00536  * @param nScroll: Specifies how many time

```

```

    the message will be scrolled
00537     * @param ScrollSpeed : Specifies the speed of the scroll, low value gives
00538     *             higher speed
00539     * @retval None
00540     * @note Required preconditions: The LCD should be cleared before to start the
00541     *             write operation.
00542     */
00543 void BSP_LCD_GLASS_ScrollSentence(uint8_t* ptr, uint16_t nScroll, uint16_t ScrollSpeed)
00544 {
00545     uint8_t repetition = 0, nbrchar = 0, sizestr = 0;
00546     uint8_t* ptr1;
00547     uint8_t str[6] = "";
00548
00549     /* Reset interrupt variable in case key was press before entering function */
00550     bLCDGlass_KeyPressed = 0;
00551
00552     if(ptr == 0)
00553     {
00554         return;
00555     }
00556
00557     /* To calculate end of string */
00558     for(ptr1 = ptr, sizestr = 0; *ptr1 != 0; sizestr++, ptr1++);
00559
00560     ptr1 = ptr;
00561
00562     BSP_LCD_GLASS_DisplayString(str);
00563     HAL_Delay(ScrollSpeed);
00564
00565     /* To shift the string for scrolling display*/

```

```

00566     for (repetition = 0; repetition < nScroll;
00567         repetition++)
00567     {
00568         for(nbrchar = 0; nbrchar < sizestr; nbrchar++)
00569         {
00570             *(str) =* (ptr1+((nbrchar+1)%sizestr))
00571             ;
00571             *(str+1) =* (ptr1+((nbrchar+2)%sizestr
00572             ));
00572             *(str+2) =* (ptr1+((nbrchar+3)%sizestr
00573             ));
00573             *(str+3) =* (ptr1+((nbrchar+4)%sizestr
00574             ));
00574             *(str+4) =* (ptr1+((nbrchar+5)%sizestr
00575             ));
00575             *(str+5) =* (ptr1+((nbrchar+6)%sizestr
00576             ));
00576             BSP_LCD_GLASS_Clear();
00577             BSP_LCD_GLASS_DisplayString(str);
00578
00579             /* user button pressed stop the scrolling sentence */
00580             if(bLCDGlass_KeyPressed)
00581             {
00582                 bLCDGlass_KeyPressed = 0;
00583                 return;
00584             }
00585             HAL_Delay(ScrollSpeed);
00586         }
00587     }
00588 }
00589
00590 /**
00591  * @}
00592  */
00593

```

```

00594 /** @addtogroup STM32L476G_DISCOVERY_LCD_Private_Functions
00595     * @{
00596     */
00597
00598 /**
00599     * @brief Initialize the LCD MSP.
00600     * @param hlcd: LCD handle
00601     * @retval None
00602     */
00603 static void LCD_MspInit(LCD_HandleTypeDef *h
lcd)
00604 {
00605     GPIO_InitTypeDef gpioinitstruct = {0};
00606     RCC_OscInitTypeDef oscinitstruct = {0};
00607     RCC_PeriphCLKInitTypeDef periphclkstruct =
{0};
00608
00609     /*##-1- Enable PWR peripheral Clock #####
#####*/
00610     __HAL_RCC_PWR_CLK_ENABLE();
00611
00612     /*##-2- Configure LSE as RTC clock soucre
#####*/
00613     oscinitstruct.OscillatorType = RCC_OSCILL
ATORTYPE_LSE;
00614     oscinitstruct.PLL.PLLState = RCC_PLL_NO
NE;
00615     oscinitstruct.LSEState = RCC_LSE_ON
;
00616     if(HAL_RCC_OscConfig(&oscinitstruct) != HA
L_OK)
00617     {
00618         while(1);
00619     }
00620
00621     /*##-3- Select LSE as RTC clock source.###

```



```

#####*/
00622  /* Backup domain management is done in RCC
function */
00623  periphclkstruct.PeriphClockSelection = RCC
_PERIPHCLK_RTC;
00624  periphclkstruct.RTCClockSelection = RCC_RT
CCLKSOURCE_LSE;
00625  HAL_RCCEx_PeriphCLKConfig(&periphclkstruct
);
00626
00627  /*##-4- Enable LCD GPIO Clocks #####
#####*/
00628  __HAL_RCC_GPIOA_CLK_ENABLE();
00629  __HAL_RCC_GPIOB_CLK_ENABLE();
00630  __HAL_RCC_GPIOC_CLK_ENABLE();
00631  __HAL_RCC_GPIOD_CLK_ENABLE();
00632
00633
00634  /*##-5- Configure peripheral GPIO #####
#####*/
00635  /* Configure Output for LCD */
00636  /* Port A */
00637  gpioinitstruct.Pin          = LCD_GPIO_BANKA
_PINS;
00638  gpioinitstruct.Mode        = GPIO_MODE_AF_P
P;
00639  gpioinitstruct.Pull        = GPIO_NOPULL;
00640  gpioinitstruct.Speed        = GPIO_SPEED_HIG
H;
00641  gpioinitstruct.Alternate    = GPIO_AF11_LCD;
00642  HAL_GPIO_Init(GPIOA, &gpioinitstruct);
00643
00644  /* Port B */
00645  gpioinitstruct.Pin          = LCD_GPIO_BANKB
_PINS;
00646  HAL_GPIO_Init(GPIOB, &gpioinitstruct);
00647

```

```

00648      /* Port C */
00649      gpioinitstruct.Pin          = LCD_GPIO_BANKC
_PINS;
00650      HAL_GPIO_Init(GPIOC, &gpioinitstruct);
00651
00652      /* Port D */
00653      gpioinitstruct.Pin          = LCD_GPIO_BANKD
_PINS;
00654      HAL_GPIO_Init(GPIOD, &gpioinitstruct);
00655
00656      /* Wait for the external capacitor Cext wh
ich is connected to the VLCD pin is charged
00657      (approximately 2ms for Cext=1uF) */
00658      HAL_Delay(2);
00659
00660      /*##-6- Enable LCD peripheral Clock #####
#####*/
00661      __HAL_RCC_LCD_CLK_ENABLE();
00662  }
00663
00664  /**
00665   * @brief DeInitialize the LCD MSP.
00666   * @param hlcd: LCD handle
00667   * @retval None
00668   */
00669  static void LCD_MspDeInit(LCD_HandleTypeDef
*hlcd)
00670  {
00671      uint32_t gpiopin = 0;
00672
00673      /*##-1- Enable LCD GPIO Clocks #####
#####*/
00674      __HAL_RCC_GPIOA_CLK_ENABLE();
00675      __HAL_RCC_GPIOB_CLK_ENABLE();
00676      __HAL_RCC_GPIOC_CLK_ENABLE();
00677      __HAL_RCC_GPIOD_CLK_ENABLE();
00678

```

```

00679  /*##-1- Configure peripheral GPIO #####
#####*/
00680  /* Configure Output for LCD */
00681  /* Port A */
00682  gpiopin = LCD_GPIO_BANKA_PINS;
00683  HAL_GPIO_DeInit(GPIOA, gpiopin);
00684
00685  /* Port B */
00686  gpiopin = LCD_GPIO_BANKB_PINS;
00687  HAL_GPIO_DeInit(GPIOB, gpiopin);
00688
00689  /* Port C*/
00690  gpiopin = LCD_GPIO_BANKC_PINS;
00691  HAL_GPIO_DeInit(GPIOC, gpiopin);
00692
00693  /* Port D */
00694  gpiopin = LCD_GPIO_BANKD_PINS;
00695  HAL_GPIO_DeInit(GPIOD, gpiopin);
00696
00697  /*##-5- Enable LCD peripheral Clock #####
#####*/
00698  __HAL_RCC_LCD_CLK_DISABLE();
00699  }
00700
00701  /**
00702   * @brief Convert an ascii char to the a L
CD digit.
00703   * @param Char: a char to display.
00704   * @param Point: a point to add in front o
f char
00705   *           This parameter can be: POINT_OFF
or POINT_ON
00706   * @param Colon : flag indicating if a col
on character has to be added in front
00707   *           of displayed character.
00708   *           This parameter can be: DOUBLEPOI
NT_OFF or DOUBLEPOINT_ON.

```

```
00709     * @retval None
00710     */
00711 static void Convert(uint8_t* Char, Point_Type
00712 #define Point, DoublePoint_Typedef Colon)
00712 {
00713     uint16_t ch = 0 ;
00714     uint8_t loop = 0, index = 0;
00715
00716     switch (*Char)
00717     {
00718         case ' ' :
00719             ch = 0x00;
00720             break;
00721
00722         case '*':
00723             ch = C_STAR;
00724             break;
00725
00726         case '(':
00727             ch = C_OPENPARAMAP;
00728             break;
00729
00730         case ')':
00731             ch = C_CLOSEPARAMAP;
00732             break;
00733
00734         case 'd':
00735             ch = C_DMAP;
00736             break;
00737
00738         case 'm':
00739             ch = C_MMAP;
00740             break;
00741
00742         case 'n':
00743             ch = C_NMAP;
00744             break;
```

```
00745
00746     case '?' :
00747         ch = C_UMAP;
00748         break;
00749
00750     case '-' :
00751         ch = C_MINUS;
00752         break;
00753
00754     case '+' :
00755         ch = C_PLUS;
00756         break;
00757
00758     case '/' :
00759         ch = C_SLATCH;
00760         break;
00761
00762     case '?' :
00763         ch = C_PERCENT_1;
00764         break;
00765     case '%' :
00766         ch = C_PERCENT_2;
00767         break;
00768     case 255 :
00769         ch = C_FULL;
00770         break ;
00771
00772     case '0' :
00773     case '1' :
00774     case '2' :
00775     case '3' :
00776     case '4' :
00777     case '5' :
00778     case '6' :
00779     case '7' :
00780     case '8' :
00781     case '9' :
```

```

00782         ch = NumberMap[*Char - ASCII_CHAR_0];

00783         break;
00784
00785         default:
00786             /* The character Char is one letter in
upper case*/
00787             if ( (*Char < ASCII_CHAR_LEFT_OPEN_BRA
CKET) && (*Char > ASCII_CHAR_AT_SYMBOL) )
00788             {
00789                 ch = CapLetterMap[*Char - 'A'];
00790             }
00791             /* The character Char is one letter in
lower case*/
00792             if ( (*Char < ASCII_CHAR_LEFT_OPEN_BRA
CE) && ( *Char > ASCII_CHAR_APOSTROPHE) )
00793             {
00794                 ch = CapLetterMap[*Char - 'a'];
00795             }
00796             break;
00797     }
00798
00799     /* Set the digital point can be displayed
if the point is on */
00800     if (Point == POINT_ON)
00801     {
00802         ch |= 0x0002;
00803     }
00804
00805     /* Set the "COL" segment in the character
that can be displayed if the colon is on */
00806     if (Colon == DOUBLEPOINT_ON)
00807     {
00808         ch |= 0x0020;
00809     }
00810
00811     for (loop = 12, index=0 ;index < 4; loop -=

```

```

    4, index++)
00812     {
00813         Digit[index] = (ch >> loop) & 0x0f; /*To
    isolate the less significant digit */
00814     }
00815 }
00816
00817 /**
00818  * @brief Write a character in the LCD fra
    me buffer.
00819  * @param ch: the character to display.
00820  * @param Point: a point to add in front o
    f char
00821  *          This parameter can be: POINT_OFF
    or POINT_ON
00822  * @param Colon: flag indicating if a colo
    n character has to be added in front
00823  *          of displayed character.
00824  *          This parameter can be: DOUBLEPOI
    NT_OFF or DOUBLEPOINT_ON.
00825  * @param Position: position in the LCD of
    the character to write [1:6]
00826  * @retval None
00827  */
00828 static void WriteChar(uint8_t* ch, Point_Typ
    edef Point, DoublePoint_Typedef Colon, DigitPositi
    on_Typedef Position)
00829 {
00830     uint32_t data = 0x00;
00831     /* To convert displayed character in segme
    nt in array digit */
00832     Convert(ch, (Point_Typedef)Point, (DoubleP
    oint_Typedef)Colon);
00833
00834     switch (Position)
00835     {
00836         /* Position 1 on LCD (Digit1)*/

```

```

00837         case LCD_DIGIT_POSITION_1:
00838             data = ((Digit[0] & 0x1) << LCD_SEG0_S
HIFT) | (((Digit[0] & 0x2) >> 1) << LCD_SEG1_SHIFT
)
00839             | (((Digit[0] & 0x4) >> 2) << LCD_
SEG22_SHIFT) | (((Digit[0] & 0x8) >> 3) << LCD_SEG
23_SHIFT);
00840             HAL_LCD_Write(&LCDHandle, LCD_DIGIT1_C
OM0, LCD_DIGIT1_COM0_SEG_MASK, data); /* 1G 1B 1M
1E */
00841
00842             data = ((Digit[1] & 0x1) << LCD_SEG0_S
HIFT) | (((Digit[1] & 0x2) >> 1) << LCD_SEG1_SHIFT
)
00843             | (((Digit[1] & 0x4) >> 2) << LCD_
SEG22_SHIFT) | (((Digit[1] & 0x8) >> 3) << LCD_SEG
23_SHIFT);
00844             HAL_LCD_Write(&LCDHandle, LCD_DIGIT1_C
OM1, LCD_DIGIT1_COM1_SEG_MASK, data) ; /* 1F 1A 1C
1D */
00845
00846             data = ((Digit[2] & 0x1) << LCD_SEG0_S
HIFT) | (((Digit[2] & 0x2) >> 1) << LCD_SEG1_SHIFT
)
00847             | (((Digit[2] & 0x4) >> 2) << LCD_
SEG22_SHIFT) | (((Digit[2] & 0x8) >> 3) << LCD_SEG
23_SHIFT);
00848             HAL_LCD_Write(&LCDHandle, LCD_DIGIT1_C
OM2, LCD_DIGIT1_COM2_SEG_MASK, data) ; /* 1Q 1K 1C
0l 1P */
00849
00850             data = ((Digit[3] & 0x1) << LCD_SEG0_S
HIFT) | (((Digit[3] & 0x2) >> 1) << LCD_SEG1_SHIFT
)
00851             | (((Digit[3] & 0x4) >> 2) << LCD_
SEG22_SHIFT) | (((Digit[3] & 0x8) >> 3) << LCD_SEG
23_SHIFT);

```



```

00852         HAL_LCD_Write(&LCDHandle, LCD_DIGIT1_COM3, LCD_DIGIT1_COM3_SEG_MASK, data) ; /* 1H 1J 1D
P 1N */
00853         break;
00854
00855         /* Position 2 on LCD (Digit2)*/
00856         case LCD_DIGIT_POSITION_2:
00857             data = ((Digit[0] & 0x1) << LCD_SEG2_SHIFT) | (((Digit[0] & 0x2) >> 1) << LCD_SEG3_SHIFT
)
00858                 | (((Digit[0] & 0x4) >> 2) << LCD_SEG20_SHIFT) | (((Digit[0] & 0x8) >> 3) << LCD_SEG
21_SHIFT);
00859             HAL_LCD_Write(&LCDHandle, LCD_DIGIT2_COM0, LCD_DIGIT2_COM0_SEG_MASK, data); /* 1G 1B 1M
1E */
00860
00861             data = ((Digit[1] & 0x1) << LCD_SEG2_SHIFT) | (((Digit[1] & 0x2) >> 1) << LCD_SEG3_SHIFT
)
00862                 | (((Digit[1] & 0x4) >> 2) << LCD_SEG20_SHIFT) | (((Digit[1] & 0x8) >> 3) << LCD_SEG
21_SHIFT);
00863             HAL_LCD_Write(&LCDHandle, LCD_DIGIT2_COM1, LCD_DIGIT2_COM1_SEG_MASK, data) ; /* 1F 1A 1C
1D */
00864
00865             data = ((Digit[2] & 0x1) << LCD_SEG2_SHIFT) | (((Digit[2] & 0x2) >> 1) << LCD_SEG3_SHIFT
)
00866                 | (((Digit[2] & 0x4) >> 2) << LCD_SEG20_SHIFT) | (((Digit[2] & 0x8) >> 3) << LCD_SEG
21_SHIFT);
00867             HAL_LCD_Write(&LCDHandle, LCD_DIGIT2_COM2, LCD_DIGIT2_COM2_SEG_MASK, data) ; /* 1Q 1K 1C
o1 1P */
00868

```

```

00869         data = ((Digit[3] & 0x1) << LCD_SEG2_S
HIFT) | (((Digit[3] & 0x2) >> 1) << LCD_SEG3_SHIFT
)
00870         | (((Digit[3] & 0x4) >> 2) << LCD_
SEG20_SHIFT) | (((Digit[3] & 0x8) >> 3) << LCD_SEG
21_SHIFT);
00871         HAL_LCD_Write(&LCDHandle, LCD_DIGIT2_C
OM3, LCD_DIGIT2_COM3_SEG_MASK, data) ; /* 1H 1J 1D
P 1N */
00872         break;
00873
00874         /* Position 3 on LCD (Digit3)*/
00875         case LCD_DIGIT_POSITION_3:
00876         data = ((Digit[0] & 0x1) << LCD_SEG4_S
HIFT) | (((Digit[0] & 0x2) >> 1) << LCD_SEG5_SHIFT
)
00877         | (((Digit[0] & 0x4) >> 2) << LCD_
SEG18_SHIFT) | (((Digit[0] & 0x8) >> 3) << LCD_SEG
19_SHIFT);
00878         HAL_LCD_Write(&LCDHandle, LCD_DIGIT3_C
OM0, LCD_DIGIT3_COM0_SEG_MASK, data); /* 1G 1B 1M
1E */
00879
00880         data = ((Digit[1] & 0x1) << LCD_SEG4_S
HIFT) | (((Digit[1] & 0x2) >> 1) << LCD_SEG5_SHIFT
)
00881         | (((Digit[1] & 0x4) >> 2) << LCD_
SEG18_SHIFT) | (((Digit[1] & 0x8) >> 3) << LCD_SEG
19_SHIFT);
00882         HAL_LCD_Write(&LCDHandle, LCD_DIGIT3_C
OM1, LCD_DIGIT3_COM1_SEG_MASK, data) ; /* 1F 1A 1C
1D */
00883
00884         data = ((Digit[2] & 0x1) << LCD_SEG4_S
HIFT) | (((Digit[2] & 0x2) >> 1) << LCD_SEG5_SHIFT
)
00885         | (((Digit[2] & 0x4) >> 2) << LCD_

```

```

SEG18_SHIFT) | (((Digit[2] & 0x8) >> 3) << LCD_SEG
19_SHIFT);
00886      HAL_LCD_Write(&LCDHandle, LCD_DIGIT3_C
OM2, LCD_DIGIT3_COM2_SEG_MASK, data) ; /* 1Q 1K 1C
0l 1P */
00887
00888      data = ((Digit[3] & 0x1) << LCD_SEG4_S
HIFT) | (((Digit[3] & 0x2) >> 1) << LCD_SEG5_SHIFT
)
00889      | (((Digit[3] & 0x4) >> 2) << LCD_
SEG18_SHIFT) | (((Digit[3] & 0x8) >> 3) << LCD_SEG
19_SHIFT);
00890      HAL_LCD_Write(&LCDHandle, LCD_DIGIT3_C
OM3, LCD_DIGIT3_COM3_SEG_MASK, data) ; /* 1H 1J 1D
P 1N */
00891      break;
00892
00893      /* Position 4 on LCD (Digit4)*/
00894      case LCD_DIGIT_POSITION_4:
00895      data = ((Digit[0] & 0x1) << LCD_SEG6_S
HIFT) | (((Digit[0] & 0x8) >> 3) << LCD_SEG17_SHIFT
);
00896      HAL_LCD_Write(&LCDHandle, LCD_DIGIT4_C
OM0, LCD_DIGIT4_COM0_SEG_MASK, data); /* 1G 1B 1M
1E */
00897
00898      data = (((Digit[0] & 0x2) >> 1) << LCD
_SEG7_SHIFT) | (((Digit[0] & 0x4) >> 2) << LCD_SEG
16_SHIFT);
00899      HAL_LCD_Write(&LCDHandle, LCD_DIGIT4_C
OM0_1, LCD_DIGIT4_COM0_1_SEG_MASK, data); /* 1G 1B
1M 1E */
00900
00901      data = ((Digit[1] & 0x1) << LCD_SEG6_S
HIFT) | (((Digit[1] & 0x8) >> 3) << LCD_SEG17_SHIFT
);
00902      HAL_LCD_Write(&LCDHandle, LCD_DIGIT4_C

```

```

OM1, LCD_DIGIT4_COM1_SEG_MASK, data) ; /* 1F 1A 1C
1D */
00903
00904     data = (((Digit[1] & 0x2) >> 1) << LCD
_SEG7_SHIFT) | (((Digit[1] & 0x4) >> 2) << LCD_SEG
16_SHIFT);
00905     HAL_LCD_Write(&LCDHandle, LCD_DIGIT4_C
OM1_1, LCD_DIGIT4_COM1_1_SEG_MASK, data) ; /* 1F 1
A 1C 1D */
00906
00907     data = ((Digit[2] & 0x1) << LCD_SEG6_S
HIFT) | (((Digit[2] & 0x8) >> 3) << LCD_SEG17_SHIFT
);
00908     HAL_LCD_Write(&LCDHandle, LCD_DIGIT4_C
OM2, LCD_DIGIT4_COM2_SEG_MASK, data) ; /* 1Q 1K 1C
o1 1P */
00909
00910     data = (((Digit[2] & 0x2) >> 1) << LCD
_SEG7_SHIFT) | (((Digit[2] & 0x4) >> 2) << LCD_SEG
16_SHIFT);
00911     HAL_LCD_Write(&LCDHandle, LCD_DIGIT4_C
OM2_1, LCD_DIGIT4_COM2_1_SEG_MASK, data) ; /* 1Q 1
K 1Co1 1P */
00912
00913     data = ((Digit[3] & 0x1) << LCD_SEG6_S
HIFT) | (((Digit[3] & 0x8) >> 3) << LCD_SEG17_SHIFT
);
00914     HAL_LCD_Write(&LCDHandle, LCD_DIGIT4_C
OM3, LCD_DIGIT4_COM3_SEG_MASK, data) ; /* 1H 1J 1D
P 1N */
00915
00916     data = (((Digit[3] & 0x2) >> 1) << LCD
_SEG7_SHIFT) | (((Digit[3] & 0x4) >> 2) << LCD_SEG
16_SHIFT);
00917     HAL_LCD_Write(&LCDHandle, LCD_DIGIT4_C
OM3_1, LCD_DIGIT4_COM3_1_SEG_MASK, data) ; /* 1H 1
J 1DP 1N */

```

```

00918         break;
00919
00920         /* Position 5 on LCD (Digit5)*/
00921         case LCD_DIGIT_POSITION_5:
00922             data = (((Digit[0] & 0x2) >> 1) << LCD_SEG9_SHIFT) | (((Digit[0] & 0x4) >> 2) << LCD_SEG14_SHIFT);
00923             HAL_LCD_Write(&LCDHandle, LCD_DIGIT5_COM0, LCD_DIGIT5_COM0_SEG_MASK, data); /* 1G 1B 1M 1E */
00924
00925             data = ((Digit[0] & 0x1) << LCD_SEG8_SHIFT) | (((Digit[0] & 0x8) >> 3) << LCD_SEG15_SHIFT);
00926             HAL_LCD_Write(&LCDHandle, LCD_DIGIT5_COM0_1, LCD_DIGIT5_COM0_1_SEG_MASK, data); /* 1G 1B 1M 1E */
00927
00928             data = (((Digit[1] & 0x2) >> 1) << LCD_SEG9_SHIFT) | (((Digit[1] & 0x4) >> 2) << LCD_SEG14_SHIFT);
00929             HAL_LCD_Write(&LCDHandle, LCD_DIGIT5_COM1, LCD_DIGIT5_COM1_SEG_MASK, data); /* 1F 1A 1C 1D */
00930
00931             data = ((Digit[1] & 0x1) << LCD_SEG8_SHIFT) | (((Digit[1] & 0x8) >> 3) << LCD_SEG15_SHIFT);
00932             HAL_LCD_Write(&LCDHandle, LCD_DIGIT5_COM1_1, LCD_DIGIT5_COM1_1_SEG_MASK, data); /* 1F 1A 1C 1D */
00933
00934             data = (((Digit[2] & 0x2) >> 1) << LCD_SEG9_SHIFT) | (((Digit[2] & 0x4) >> 2) << LCD_SEG14_SHIFT);
00935             HAL_LCD_Write(&LCDHandle, LCD_DIGIT5_COM2, LCD_DIGIT5_COM2_SEG_MASK, data); /* 1Q 1K 1C

```

```

0l 1P  */
00936
00937     data = ((Digit[2] & 0x1) << LCD_SEG8_S
HIFT) | (((Digit[2] & 0x8) >> 3) << LCD_SEG15_SHIFT
);
00938     HAL_LCD_Write(&LCDHandle, LCD_DIGIT5_C
OM2_1, LCD_DIGIT5_COM2_1_SEG_MASK, data) ; /* 1Q 1
K 1C0l 1P  */
00939
00940     data = (((Digit[3] & 0x2) >> 1) << LCD
_SEG9_SHIFT) | (((Digit[3] & 0x4) >> 2) << LCD_SEG
14_SHIFT);
00941     HAL_LCD_Write(&LCDHandle, LCD_DIGIT5_C
OM3, LCD_DIGIT5_COM3_SEG_MASK, data) ; /* 1H 1J 1D
P 1N  */
00942
00943     data = ((Digit[3] & 0x1) << LCD_SEG8_S
HIFT) | (((Digit[3] & 0x8) >> 3) << LCD_SEG15_SHIFT
);
00944     HAL_LCD_Write(&LCDHandle, LCD_DIGIT5_C
OM3_1, LCD_DIGIT5_COM3_1_SEG_MASK, data) ; /* 1H 1
J 1DP 1N  */
00945         break;
00946
00947     /* Position 6 on LCD (Digit6)*/
00948     case LCD_DIGIT_POSITION_6:
00949         data = ((Digit[0] & 0x1) << LCD_SEG10_
SHIFT) | (((Digit[0] & 0x2) >> 1) << LCD_SEG11_SHI
FT)
00950             | (((Digit[0] & 0x4) >> 2) << LCD_
SEG12_SHIFT) | (((Digit[0] & 0x8) >> 3) << LCD_SEG
13_SHIFT);
00951     HAL_LCD_Write(&LCDHandle, LCD_DIGIT6_C
OM0, LCD_DIGIT6_COM0_SEG_MASK, data); /* 1G 1B 1M
1E  */
00952
00953     data = ((Digit[1] & 0x1) << LCD_SEG10_

```

```

SHIFT) | (((Digit[1] & 0x2) >> 1) << LCD_SEG11_SHI
FT)
00954          | (((Digit[1] & 0x4) >> 2) << LCD_
SEG12_SHIFT) | (((Digit[1] & 0x8) >> 3) << LCD_SEG
13_SHIFT);
00955          HAL_LCD_Write(&LCDHandle, LCD_DIGIT6_C
OM1, LCD_DIGIT6_COM1_SEG_MASK, data) ; /* 1F 1A 1C
1D */
00956
00957          data = ((Digit[2] & 0x1) << LCD_SEG10_
SHIFT) | (((Digit[2] & 0x2) >> 1) << LCD_SEG11_SHI
FT)
00958          | (((Digit[2] & 0x4) >> 2) << LCD_
SEG12_SHIFT) | (((Digit[2] & 0x8) >> 3) << LCD_SEG
13_SHIFT);
00959          HAL_LCD_Write(&LCDHandle, LCD_DIGIT6_C
OM2, LCD_DIGIT6_COM2_SEG_MASK, data) ; /* 1Q 1K 1C
ol 1P */
00960
00961          data = ((Digit[3] & 0x1) << LCD_SEG10_
SHIFT) | (((Digit[3] & 0x2) >> 1) << LCD_SEG11_SHI
FT)
00962          | (((Digit[3] & 0x4) >> 2) << LCD_
SEG12_SHIFT) | (((Digit[3] & 0x8) >> 3) << LCD_SEG
13_SHIFT);
00963          HAL_LCD_Write(&LCDHandle, LCD_DIGIT6_C
OM3, LCD_DIGIT6_COM3_SEG_MASK, data) ; /* 1H 1J 1D
P 1N */
00964          break;
00965
00966          default:
00967              break;
00968      }
00969 }
00970
00971 /**
00972      * @}

```

```
00973    */
00974
00975  /**
00976    * @}
00977    */
00978
00979  /**
00980    * @}
00981    */
00982
00983  /**
00984    * @}
00985    */
00986
00987  /** ***** (C) COPYRIGHT STMicroelectronics *****END OF FILE*****/
```



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_gyroscope.h

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery_gyroscope.h
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file contains definitions
00008                  for stm32l476g_discovery_gyroscope.c
00009                  firmware driver.
00010      ****
00011      * @attention
00012      *
00013      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00014      * icroelectronics</center></h2>
00015      *
00016      * Redistribution and use in source and bin
00017      * ary forms, with or without modification,
00018      * are permitted provided that the followin
00019      * g conditions are met:
```

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

```

00035      *
00036      ****
00037      */
00038
00039 /* Define to prevent recursive inclusion ---
-----*/
00040 #ifndef __STM32L476G_DISCOVERY_GYROSCOPE_H
00041 #define __STM32L476G_DISCOVERY_GYROSCOPE_H
00042
00043 #ifdef __cplusplus
00044     extern "C" {
00045 #endif
00046
00047 /* Includes -----
-----*/
00048 #include "stm32l476g_discovery.h"
00049 /* Include Gyroscope component driver */
00050 #include "../Components/l3gd20/l3gd20.h"
00051
00052 /** @addtogroup BSP
00053     * @{
00054     */
00055
00056 /** @addtogroup STM32L476G_DISCOVERY
00057     * @{
00058     */
00059
00060 /** @addtogroup STM32L476G_DISCOVERY_GYROSCOPE
00061     * @{
00062     */
00063
00064 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
00065     * _Exported_Types Exported Types
00066     * @{

```

```

00067 typedef enum
00068 {
00069     GYRO_OK = 0,
00070     GYRO_ERROR = 1,
00071     GYRO_TIMEOUT = 2
00072 }
00073 GYRO_StatusTypeDef;
00074
00075 /**
00076  * @}
00077  */
00078
00079 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
    _Exported_Constants Exported Constants
00080  * @{
00081  */
00082
00083 /**
00084  * @}
00085  */
00086
00087 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
    _Exported_Macros Exported Macros
00088  * @{
00089  */
00090
00091 /**
00092  * @}
00093  */
00094
00095 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
    _Exported_Functions Exported Functions
00096  * @{
00097  */
00098 /* Sensor Configuration Functions */
00099 uint8_t BSP_GYRO_Init(void);
00100 void     BSP_GYRO_DeInit(void);

```

```

00101 void      BSP_GYRO_LowPower(void);
00102 void      BSP_GYRO_Reset(void);
00103 uint8_t    BSP_GYRO_ReadID(void);
00104 void      BSP_GYRO_ITConfig(GYRO_InterruptConf
igTypeDef *pIntConfigStruct);
00105 void      BSP_GYRO_EnableIT(uint8_t IntPin);
00106 void      BSP_GYRO_DisableIT(uint8_t IntPin);
00107 void      BSP_GYRO_GetXYZ(float* pfData);
00108
00109 /**
00110  * @}
00111  */
00112
00113 /**
00114  * @}
00115  */
00116
00117 /**
00118  * @}
00119  */
00120
00121 /**
00122  * @}
00123  */
00124
00125 #ifdef __cplusplus
00126 }
00127 #endif
00128
00129 #endif /* __STM32L476G_DISCOVERY_GYROSCOPE_H
 */
00130
00131 /***** (C) COPYRIGHT STMi
croelectronics *****/

```

BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<a href="#">File List</a>	<a href="#">Globals</a>			
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>		

## stm32l476g\_discovery\_gyroscope.c

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      ****
00003      * @file      stm32l476g_discovery_gyroscope.c
00004
00004      * @author    MCD Application Team
00005      * @version  $VERSION$
00006      * @date     $DATE$
00007      * @brief    This file provides a set of fun
00008                  ctions needed to manage the L3GD20
00008                  *
00008                  MEMS accelerometer available on
00008                  STM32L476G-Discovery board.
00009      ****
00009      ****
00010      * @attention
00011      *
00012      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00012      icroelectronics</center></h2>
00013      *
00014      * Redistribution and use in source and bin
00014      ary forms, with or without modification,
00015      * are permitted provided that the followin
```

g conditions are met:

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,

00017 \* this list of conditions and the fol  
lowing disclaimer.

00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,

00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation

00020 \* and/or other materials provided wit  
h the distribution.

00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors

00022 \* may be used to endorse or promote p  
roducts derived from this software

00023 \* without specific prior written perm  
ission.

00024 \*

00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"

00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE

00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE

00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE

00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL

00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR

00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER

00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,

00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE

00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE



## POSSIBILITY OF SUCH DAMAGE.

```
00035      *
00036      *****
*****
00037      */
00038
00039 /* Includes -----
----- */
00040 #include "stm32l476g_discovery_gyroscope.h"
00041
00042 /** @addtogroup BSP
00043     * @{
00044     */
00045
00046 /** @addtogroup STM32L476G_DISCOVERY
00047     * @{
00048     */
00049
00050 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
    STM32L476G-DISCOVERY GYROSCOPE
00051     * @{
00052     */
00053
00054 /* Private typedef -----
----- */
00055 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
    _Private_Types Private Types
00056     * @{
00057     */
00058 /**
00059     * @}
00060     */
00061
00062 /* Private defines -----
----- */
00063 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
    _Private_Constants Private Constants
```

```

00064     * @{
00065     */
00066 /**
00067     * @}
00068     */
00069
00070 /* Private macros -----
-----*/
00071 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
_Private_Macros Private Macros
00072     * @{
00073     */
00074 /**
00075     * @}
00076     */
00077
00078 /* Private variables -----
-----*/
00079 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
_Private_Variables Private Variables
00080     * @{
00081     */
00082 static GYRO_DrvTypeDef *GyroscopeDrv;
00083
00084 /**
00085     * @}
00086     */
00087
00088 /* Private function prototypes -----
-----*/
00089 /** @defgroup STM32L476G_DISCOVERY_GYROSCOPE
_Private_FunctionPrototypes Private Functions
00090     * @{
00091     */
00092 /**
00093     * @}
00094     */

```

```

00095
00096 /* Exported functions -----
-----*/
00097 /** @addtogroup STM32L476G_DISCOVERY_GYROSCO
PE_Exported_Functions
00098     * @{
00099     */
00100
00101 /**
00102     * @brief Initialize Gyroscope.
00103     * @retval GYRO_OK or GYRO_ERROR
00104     */
00105 uint8_t BSP_GYRO_Init(void)
00106 {
00107     uint8_t ret = GYRO_ERROR;
00108     uint16_t ctrl = 0x0000;
00109     GYRO_InitTypeDef L3GD20_InitStructure;
00110     GYRO_FilterConfigTypeDef L3GD20_FilterStru
cture={0,0};
00111
00112     if((L3gd20Drv.ReadID() == I_AM_L3GD20) ||
(L3gd20Drv.ReadID() == I_AM_L3GD20_TR))
00113     {
00114         /* Initialize the gyroscope driver struc
ture */
00115         GyroscopeDrv = &L3gd20Drv;
00116
00117         /* Configure Mems : data rate, power mod
e, full scale and axes */
00118         L3GD20_InitStructure.Power_Mode = L3GD20
_MODE_ACTIVE;
00119         L3GD20_InitStructure.Output_DataRate = L
3GD20_OUTPUT_DATARATE_1;
00120         L3GD20_InitStructure.Axes_Enable = L3GD2
0_AXES_ENABLE;
00121         L3GD20_InitStructure.Band_Width = L3GD20
_BANDWIDTH_4;

```

```

00122     L3GD20_InitStructure.BlockData_Update =
L3GD20_BlockDataUpdate_Continuous;
00123     L3GD20_InitStructure.Endianness = L3GD20
_BLE_LSB;
00124     L3GD20_InitStructure.Full_Scale = L3GD20
_FULLSCALE_500;
00125
00126     /* Configure MEMS: data rate, power mode
, full scale and axes */
00127     ctrl = (uint16_t) (L3GD20_InitStructure.
Power_Mode | L3GD20_InitStructure.Output_DataRate
| \
00128                               L3GD20_InitStructure.
Axes_Enable | L3GD20_InitStructure.Band_Width);
00129
00130     ctrl |= (uint16_t) ((L3GD20_InitStructur
e.BlockData_Update | L3GD20_InitStructure.Endianne
ss | \
00131                               L3GD20_InitStructure
.Full_Scale) << 8);
00132
00133     /* Initialize component */
00134     GyroscopeDrv->Init(ctrl);
00135
00136     L3GD20_FilterStructure.HighPassFilter_Mo
de_Selection =L3GD20_HPM_NORMAL_MODE_RES;
00137     L3GD20_FilterStructure.HighPassFilter_Cu
tOff_Frequency = L3GD20_HPFCF_0;
00138
00139     ctrl = (uint8_t) ((L3GD20_FilterStructur
e.HighPassFilter_Mode_Selection |\
00140                               L3GD20_FilterStructur
e.HighPassFilter_CutOff_Frequency));
00141
00142     /* Configure component filter */
00143     GyroscopeDrv->FilterConfig(ctrl) ;
00144

```

```

00145      /* Enable component filter */
00146      GyroscopeDrv->FilterCmd(L3GD20_HIGHPASSF
00147      ILTER_ENABLE);
00148      ret = GYRO_OK;
00149  }
00150  else
00151  {
00152      ret = GYRO_ERROR;
00153  }
00154
00155  return ret;
00156 }
00157
00158
00159 /**
00160  * @brief DeInitialize Gyroscope.
00161  * @retval None
00162  */
00163 void BSP_GYRO_DeInit(void)
00164 {
00165     GYRO_IO_DeInit();
00166 }
00167
00168
00169 /**
00170  * @brief Put Gyroscope in low power mode.
00171  * @retval None
00172  */
00173 void BSP_GYRO_LowPower(void)
00174 {
00175     uint16_t ctrl = 0x0000;
00176     GYRO_InitTypeDef L3GD20_InitStructure;
00177
00178     /* configure only Power_Mode field */
00179     L3GD20_InitStructure.Power_Mode = L3GD20_M
00180     ODE_POWERDOWN;

```

```

00180
00181     ctrl = (uint16_t) (L3GD20_InitStructure.Po
wer_Mode);
00182
00183     /* Set component in low-power mode */
00184     GyroscopeDrv->LowPower(ctrl);
00185
00186
00187 }
00188
00189 /**
00190  * @brief Read ID of Gyroscope component.
00191  * @retval ID
00192  */
00193 uint8_t BSP_GYRO_ReadID(void)
00194 {
00195     uint8_t id = 0x00;
00196
00197     if(GyroscopeDrv->ReadID != NULL)
00198     {
00199         id = GyroscopeDrv->ReadID();
00200     }
00201     return id;
00202 }
00203
00204 /**
00205  * @brief Reboot memory content of Gyrosco
pe.
00206  * @retval None
00207  */
00208 void BSP_GYRO_Reset(void)
00209 {
00210     if(GyroscopeDrv->Reset != NULL)
00211     {
00212         GyroscopeDrv->Reset();
00213     }
00214 }

```

```

00215
00216 /**
00217  * @brief Configure Gyroscope interrupts (
00218  * @param pIntConfig: pointer to a GYRO_In
00219  * structure that contains the conf
00220  * igation setting for the L3GD20 Interrupt.
00221  * @retval None
00222  */
00222 void BSP_GYRO_ITConfig(GYRO_InterruptConfigT
00223 typeDef *pIntConfig)
00223 {
00224     uint16_t interruptconfig = 0x0000;
00225
00226     if(GyroscopeDrv->ConfigIT != NULL)
00227     {
00228         /* Configure latch Interrupt request and
00229         axe interrupts */
00229         interruptconfig |= ((uint8_t)(pIntConfig
00230 ->Latch_Request | \
00231                                     pIntConfig
00232 ->Interrupt_Axes) << 8);
00231
00232         interruptconfig |= (uint8_t)(pIntConfig->Interrupt_ActiveEdge);
00233
00234         GyroscopeDrv->ConfigIT(interruptconfig);
00235     }
00236 }
00237
00238 /**
00239  * @brief Enable Gyroscope interrupts (INT
00240  * 1 or INT2).
00241  * @param IntPin: Interrupt pin
00242  * This parameter can be:
00243  * @arg L3GD20_INT1

```

```

00243     *          @arg L3GD20_INT2
00244     * @retval None
00245     */
00246 void BSP_GYRO_EnableIT(uint8_t IntPin)
00247 {
00248     if(GyroscopeDrv->EnableIT != NULL)
00249     {
00250         GyroscopeDrv->EnableIT(IntPin);
00251     }
00252 }
00253
00254 /**
00255  * @brief Disable Gyroscope interrupts (IN
T1 or INT2).
00256  * @param IntPin: Interrupt pin
00257  *          This parameter can be:
00258  *          @arg L3GD20_INT1
00259  *          @arg L3GD20_INT2
00260  * @retval None
00261  */
00262 void BSP_GYRO_DisableIT(uint8_t IntPin)
00263 {
00264     if(GyroscopeDrv->DisableIT != NULL)
00265     {
00266         GyroscopeDrv->DisableIT(IntPin);
00267     }
00268 }
00269
00270 /**
00271  * @brief Get XYZ angular acceleration fro
m the Gyroscope.
00272  * @param pfData: pointer on floating arra
y
00273  * @retval None
00274  */
00275 void BSP_GYRO_GetXYZ(float* pfData)
00276 {

```



```
00277     if(GyroscopeDrv->GetXYZ!= NULL)
00278     {
00279         GyroscopeDrv->GetXYZ(pfData);
00280     }
00281 }
00282
00283 /**
00284  * @}
00285  */
00286
00287 /**
00288  * @}
00289  */
00290
00291 /**
00292  * @}
00293  */
00294
00295 /**
00296  * @}
00297  */
00298
00299 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE*****/
```

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_idd.h

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      ****
00003      * @file      stm32l476g_discovery_idd.h
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    Header file for stm32l476g_discovery_idd.c module.
00008      ****
00009      ****
00009      * @attention
00010      *
00011      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00012      * icronics</center></h2>
00013      *
00013      * Redistribution and use in source and binary forms, with or without modification,
00014      * are permitted provided that the following conditions are met:
00015      * 1. Redistributions of source code must retain the above copyright notice,
```

00016 \* this list of conditions and the following disclaimer.

00017 \* 2. Redistributions in binary form must reproduce the above copyright notice,

00018 \* this list of conditions and the following disclaimer in the documentation

00019 \* and/or other materials provided with the distribution.

00020 \* 3. Neither the name of STMicroelectronics nor the names of its contributors

00021 \* may be used to endorse or promote products derived from this software

00022 \* without specific prior written permission.

00023 \*

00024 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

00025 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

00026 \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

00027 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

00028 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

00029 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

00030 \* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

00031 \* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

00032 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

00033 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

00034 \*

00035 \*\*\*\*\*

```

*****
00036    */
00037
00038 /* Define to prevent recursive inclusion ---
-----*/
00039 #ifndef __STM32L476G_DISCOVERY_IDD_H
00040 #define __STM32L476G_DISCOVERY_IDD_H
00041
00042 #ifdef __cplusplus
00043     extern "C" {
00044 #endif
00045
00046 /* Includes -----
-----*/
00047 #include "stm32l476g_discovery.h"
00048 /* Include Idd measurement component driver
*/
00049 #include "../Components/mfxstm32l152/mfxstm3
2l152.h"
00050
00051 /** @addtogroup BSP
00052     * @{
00053     */
00054
00055 /** @addtogroup STM32L476G_DISCOVERY
00056     * @{
00057     */
00058
00059 /** @addtogroup STM32L476G_DISCOVERY_IDD
00060     * @{
00061     */
00062
00063 /** @defgroup STM32L476G_DISCOVERY_IDD_Export
ed_Types Exported Types
00064     * @{
00065     */
00066

```

```

00067 /** @defgroup IDD_Config IDD Config
00068     * @{
00069     */
00070 typedef enum
00071 {
00072     IDD_OK = 0,
00073     IDD_TIMEOUT = 1,
00074     IDD_ZERO_VALUE = 2,
00075     IDD_ERROR = 0xFF
00076 }
00077 IDD_StatusTypeDef;
00078 /**
00079     * @}
00080     */
00081
00082 /**
00083     * @}
00084     */
00085
00086 /** @defgroup STM32L476G_DISCOVERY_IDD_Exported_Defines Exported Defines
00087     * @{
00088     */
00089 /**
00090     * @brief Shunt values on discovery in milli ohms
00091     */
00092 #define DISCOVERY_IDD_SHUNT0_VALUE
00093     ((uint16_t) 1000) /*!< value in milliohm */
00094 #define DISCOVERY_IDD_SHUNT1_VALUE
00095     ((uint16_t) 24) /*!< value in ohm */
00096 #define DISCOVERY_IDD_SHUNT2_VALUE
00097     ((uint16_t) 620) /*!< value in ohm */
00098 #define DISCOVERY_IDD_SHUNT4_VALUE
00099     ((uint16_t) 10000) /*!< value in ohm */
00100
00101

```

```

00097 /**
00098  * @brief Shunt stabilization delay on discovery in milli ohms
00099  */
00100 #define DISCOVERY_IDD_SHUNT0_STABDELAY
        ((uint8_t) 149)          /*!< value in millis
ec */
00101 #define DISCOVERY_IDD_SHUNT1_STABDELAY
        ((uint8_t) 149)          /*!< value in millis
ec */
00102 #define DISCOVERY_IDD_SHUNT2_STABDELAY
        ((uint8_t) 149)          /*!< value in millis
ec */
00103 #define DISCOVERY_IDD_SHUNT4_STABDELAY
        ((uint8_t) 255)          /*!< value in millis
ec */
00104
00105 /**
00106  * @brief IDD Ampli Gain on discovery
00107  */
00108 #if defined(USE_STM32L476G_DISCO_REVC)
00109 #define DISCOVERY_IDD_AMPLI_GAIN
        ((uint16_t) 4967)        /*!< value is gain *
100 */
00110 #else
00111 #define DISCOVERY_IDD_AMPLI_GAIN
        ((uint16_t) 4990)        /*!< value is gain *
100 */
00112 #endif
00113
00114 /**
00115  * @brief IDD Vdd Min on discovery
00116  */
00117 #define DISCOVERY_IDD_VDD_MIN
        ((uint16_t) 2000)        /*!< value in milliv
olt */
00118

```

```

00119 /**
00120      * @}
00121      */
00122
00123 /* Exported functions -----
-----*/
00124 /** @defgroup STM32L476G_DISCOVERY_IDD_Export
ted_Functions Exported Functions
00125      * @{
00126      */
00127 uint8_t    BSP_IDD_Init(void);
00128 void       BSP_IDD_DeInit(void);
00129 void       BSP_IDD_Reset(void);
00130 void       BSP_IDD_LowPower(void);
00131 void       BSP_IDD_WakeUp(void);
00132 void       BSP_IDD_StartMeasure(void);
00133 void       BSP_IDD_Config(IDD_ConfigTypeDef I
ddConfig);
00134 void       BSP_IDD_GetValue(uint32_t *IddValu
e);
00135 void       BSP_IDD_EnableIT(void);
00136 void       BSP_IDD_ClearIT(void);
00137 uint8_t    BSP_IDD_GetITStatus(void);
00138 void       BSP_IDD_DisableIT(void);
00139 uint8_t    BSP_IDD_ErrorGetCode(void);
00140 void       BSP_IDD_ErrorEnableIT(void);
00141 void       BSP_IDD_ErrorClearIT(void);
00142 uint8_t    BSP_IDD_ErrorGetITStatus(void);
00143 void       BSP_IDD_ErrorDisableIT(void);
00144
00145 /**
00146      * @}
00147      */
00148
00149 /**
00150      * @}
00151      */

```

```
00152
00153  /**
00154     * @}
00155     */
00156
00157  /**
00158     * @}
00159     */
00160
00161 #ifdef __cplusplus
00162 }
00163 #endif
00164
00165 #endif /* __STM32L476G_DISCOVERY_IDD_H */
00166
00167 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE*****/
```

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>
<a href="#">Directories</a>			
<a href="#">File List</a>	<a href="#">Globals</a>		
<a href="#">Drivers</a>	<a href="#">BSP</a>	<a href="#">STM32L476G-Discovery</a>	

## stm32l476g\_discovery\_idd.c

[Go to the documentation of this file.](#)

```
00001  /**
00002      ****
00003      * @file      stm32l476g_discovery_idd.c
00004      * @author    MCD Application Team
00005      * @version    $VERSION$
00006      * @date      $DATE$
00007      * @brief    This file provides a set of fir
00008      *           Id measurement driver for STM3
00009      ****
00010      * @attention
00011      *
00012      * <h2><center>&copy; COPYRIGHT(c) 2015 STM
00013      *           icroelectronics</center></h2>
00014      * Redistribution and use in source and bin
00015      * are permitted provided that the followin
00016      g conditions are met:
```

00016 \* 1. Redistributions of source code must  
retain the above copyright notice,  
00017 \* this list of conditions and the fol  
lowing disclaimer.  
00018 \* 2. Redistributions in binary form must  
reproduce the above copyright notice,  
00019 \* this list of conditions and the fol  
lowing disclaimer in the documentation  
00020 \* and/or other materials provided wit  
h the distribution.  
00021 \* 3. Neither the name of STMicroelectron  
ics nor the names of its contributors  
00022 \* may be used to endorse or promote p  
roducts derived from this software  
00023 \* without specific prior written perm  
ission.  
00024 \*  
00025 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00026 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00027 \* IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00028 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00029 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00030 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00031 \* SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00032 \* CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00033 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00034 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

```

00035      *
00036      ****
00037      */
00038
00039 /* Includes -----
----- */
00040 #include "stm32l476g_discovery_idd.h"
00041
00042 /** @addtogroup BSP
00043     * @{
00044     */
00045
00046 /** @addtogroup STM32L476G_DISCOVERY
00047     * @{
00048     */
00049
00050 /** @defgroup STM32L476G_DISCOVERY_IDD STM32
L476G-DISCOVERY IDD
00051     * @brief This file includes the Idd driver
for STM32L476G-DISCOVERY board.
00052     *          It allows user to measure MCU Idd
current on board, especially in
00053     *          different low power modes.
00054     * @{
00055     */
00056
00057 /** @defgroup STM32L476G_DISCOVERY_IDD_Priva
te_Defines Private Defines
00058     * @{
00059     */
00060
00061 /**
00062     * @}
00063     */
00064
00065

```

```

00066 /** @defgroup STM32L476G_DISCOVERY_IDD_Priva
te_Variables Private Variables
00067     * @{
00068     */
00069 static IDD_DrvTypeDef *IddDrv;
00070
00071 /**
00072     * @}
00073     */
00074
00075 /** @defgroup STM32L476G_DISCOVERY_IDD_Priva
te_Functions Private Functions
00076     * @{
00077     */
00078
00079 /**
00080     * @}
00081     */
00082
00083 /** @defgroup STM32L476G_DISCOVERY_IDD_Expor
ted_Functions Exported Functions
00084     * @{
00085     */
00086
00087 /**
00088     * @brief Configures IDD measurement compo
nent.
00089     * @retval IDD_OK if no problem during init
ialization
00090     */
00091 uint8_t BSP_IDD_Init(void)
00092 {
00093     IDD_ConfigTypeDef iddconfig = {0};
00094     uint8_t mfxstm32l152_id = 0;
00095     uint8_t ret = 0;
00096
00097     /* wake up mfx component in case it went t

```

```

o standby mode */
00098     mfxstm32l152_idd_drv.WakeUp(IDD_I2C_ADDRESS
);
00099     HAL_Delay(5);
00100
00101     /* Read ID and verify if the MFX is ready
*/
00102     mfxstm32l152_id = mfxstm32l152_idd_drv.ReadID(IDD_I2C_ADDRESS);
00103
00104     if((mfxstm32l152_id == MFXSTM32L152_ID_1)
|| (mfxstm32l152_id == MFXSTM32L152_ID_2))
00105     {
00106         /* Initialize the Idd driver structure */

00107         IddDrv = &mfxstm32l152_idd_drv;
00108
00109         /* Initialize the Idd driver */
00110         if(IddDrv->Init != NULL)
00111         {
00112             IddDrv->Init(IDD_I2C_ADDRESS);
00113         }
00114
00115         /* Configure Idd component with default
values */
00116         iddconfig.AmpliGain = DISCOVERY_IDD_AMPLI_GAIN;
00117         iddconfig.VddMin = DISCOVERY_IDD_VDD_MIN
;
00118         iddconfig.Shunt0Value = DISCOVERY_IDD_SHUNT0_VALUE;
00119         iddconfig.Shunt1Value = DISCOVERY_IDD_SHUNT1_VALUE;
00120         iddconfig.Shunt2Value = DISCOVERY_IDD_SHUNT2_VALUE;
00121         iddconfig.Shunt3Value = 0;
00122         iddconfig.Shunt4Value = DISCOVERY_IDD_SH

```

```

UNT4_VALUE;
00123     iddconfig.Sh0StabDelay = DISCOVERY_IDD_S
HUNT0_STABDELAY;
00124     iddconfig.Sh1StabDelay = DISCOVERY_IDD_S
HUNT1_STABDELAY;
00125     iddconfig.Sh2StabDelay = DISCOVERY_IDD_S
HUNT2_STABDELAY;
00126     iddconfig.Sh3StabDelay = 0;
00127     iddconfig.Sh4StabDelay = DISCOVERY_IDD_S
HUNT4_STABDELAY;
00128     iddconfig.ShuntNbOnBoard = MFXSTM32L152_
IDD_SHUNT_NB_4;
00129     iddconfig.ShuntNbUsed = MFXSTM32L152_IDD
_SHUNT_NB_4;
00130     iddconfig.VrefMeasurement = MFXSTM32L152
_IDD_VREF_AUTO_MEASUREMENT_ENABLE;
00131     iddconfig.Calibration = MFXSTM32L152_IDD
_AUTO_CALIBRATION_ENABLE;
00132     iddconfig.PreDelayUnit = MFXSTM32L152_ID
D_PREDELAY_20_MS;
00133     iddconfig.PreDelayValue = 0x7F;
00134     iddconfig.MeasureNb = 100;
00135     iddconfig.DeltaDelayUnit= MFXSTM32L152_I
DD_DELTADELAY_0_5_MS;
00136     iddconfig.DeltaDelayValue = 10;
00137     BSP_IDD_Config(iddconfig);
00138
00139     ret = IDD_OK;
00140 }
00141 else
00142 {
00143     ret = IDD_ERROR;
00144 }
00145
00146     return ret;
00147 }
00148

```

```

00149 /**
00150  * @brief Unconfigures IDD measurement com
00151  * @retval IDD_OK if no problem during dein
00152  */
00153 void BSP_IDD_DeInit(void)
00154 {
00155     if(IddDrv->DeInit!= NULL)
00156     {
00157         IddDrv->DeInit(IDD_I2C_ADDRESS);
00158     }
00159 }
00160
00161 /**
00162  * @brief Reset Idd measurement component.
00163  * @retval None
00164  */
00165 void BSP_IDD_Reset(void)
00166 {
00167     if(IddDrv->Reset != NULL)
00168     {
00169         IddDrv->Reset(IDD_I2C_ADDRESS);
00170     }
00171 }
00172
00173 /**
00174  * @brief Turn Idd measurement component i
00175  * @retval None
00176  */
00177 void BSP_IDD_LowPower(void)
00178 {
00179     if(IddDrv->LowPower != NULL)
00180     {
00181         IddDrv->LowPower(IDD_I2C_ADDRESS);
00182     }

```

```

00183 }
00184
00185 /**
00186  * @brief Start Measurement campaign
00187  * @retval None
00188  */
00189 void BSP_IDD_StartMeasure(void)
00190 {
00191     if(IddDrv->Start != NULL)
00192     {
00193         IddDrv->Start(IDD_I2C_ADDRESS);
00194     }
00195 }
00196
00197 /**
00198  * @brief Configure Idd component
00199  * @param IddConfig: structure of idd para
00200  * @retval None
00201  */
00202 void BSP_IDD_Config(IDD_ConfigTypeDef IddCon
fig)
00203 {
00204     if(IddDrv->Config != NULL)
00205     {
00206         IddDrv->Config(IDD_I2C_ADDRESS, IddConfi
g);
00207     }
00208 }
00209
00210 /**
00211  * @brief Get Idd current value.
00212  * @param IddValue: Pointer on u32 to stor
00213  * @retval None
00214  */
00215 void BSP_IDD_GetValue(uint32_t *IddValue)

```



```

00216 {
00217     if(IddDrv->GetValue != NULL)
00218     {
00219         IddDrv->GetValue(IDD_I2C_ADDRESS, IddVal
ue);
00220     }
00221 }
00222
00223 /**
00224  * @brief Enable Idd interrupt that warn e
nd of measurement
00225  * @retval None
00226  */
00227 void BSP_IDD_EnableIT(void)
00228 {
00229     if(IddDrv->EnableIT != NULL)
00230     {
00231         IddDrv->EnableIT(IDD_I2C_ADDRESS);
00232     }
00233 }
00234
00235 /**
00236  * @brief Clear Idd interrupt that warn en
d of measurement
00237  * @retval None
00238  */
00239 void BSP_IDD_ClearIT(void)
00240 {
00241     if(IddDrv->ClearIT != NULL)
00242     {
00243         IddDrv->ClearIT(IDD_I2C_ADDRESS);
00244     }
00245 }
00246
00247 /**
00248  * @brief Get Idd interrupt status
00249  * @retval status

```

```

00250     */
00251 uint8_t BSP_IDD_GetITStatus(void)
00252 {
00253     if(IddDrv->GetITStatus != NULL)
00254     {
00255         return (IddDrv->GetITStatus(IDD_I2C_ADDR
ESS));
00256     }
00257     else
00258     {
00259         return IDD_ERROR;
00260     }
00261 }
00262
00263 /**
00264  * @brief Disable Idd interrupt that warn
end of measurement
00265  * @retval None
00266  */
00267 void BSP_IDD_DisableIT(void)
00268 {
00269     if(IddDrv->DisableIT != NULL)
00270     {
00271         IddDrv->DisableIT(IDD_I2C_ADDRESS);
00272     }
00273 }
00274
00275 /**
00276  * @brief Get Error Code .
00277  * @retval Error code or error status
00278  */
00279 uint8_t BSP_IDD_ErrorGetCode(void)
00280 {
00281     if(IddDrv->ErrorGetSrc != NULL)
00282     {
00283         if((IddDrv->ErrorGetSrc(IDD_I2C_ADDRESS)
& MFXSTM32L152_IDD_ERROR_SRC) != RESET)

```

```

00284     {
00285         if(IddDrv->ErrorGetCode != NULL)
00286         {
00287             return IddDrv->ErrorGetCode(IDD_I2C_A
DDRESS);
00288         }
00289         else
00290         {
00291             return IDD_ERROR;
00292         }
00293     }
00294     else
00295     {
00296         return IDD_ERROR;
00297     }
00298 }
00299 else
00300 {
00301     return IDD_ERROR;
00302 }
00303 }
00304
00305
00306 /**
00307  * @brief Enable error interrupt that warn
end of measurement
00308  * @retval None
00309  */
00310 void BSP_IDD_ErrorEnableIT(void)
00311 {
00312     if(IddDrv->ErrorEnableIT != NULL)
00313     {
00314         IddDrv->ErrorEnableIT(IDD_I2C_ADDRESS);
00315     }
00316 }
00317
00318 /**

```

```

00319     * @brief Clear Error interrupt that warn
end of measurement
00320     * @retval None
00321     */
00322 void BSP_IDD_ErrorClearIT(void)
00323 {
00324     if(IddDrv->ErrorClearIT != NULL)
00325     {
00326         IddDrv->ErrorClearIT(IDD_I2C_ADDRESS);
00327     }
00328 }
00329
00330 /**
00331     * @brief Get Error interrupt status
00332     * @retval Status
00333     */
00334 uint8_t BSP_IDD_ErrorGetITStatus(void)
00335 {
00336     if(IddDrv->ErrorGetITStatus != NULL)
00337     {
00338         return (IddDrv->ErrorGetITStatus(IDD_I2C
_ADDRESS));
00339     }
00340     else
00341     {
00342         return 0;
00343     }
00344 }
00345
00346 /**
00347     * @brief Disable Error interrupt
00348     * @retval None
00349     */
00350 void BSP_IDD_ErrorDisableIT(void)
00351 {
00352     if(IddDrv->ErrorDisableIT != NULL)
00353     {

```

```
00354     IddDrv->ErrorDisableIT(IDD_I2C_ADDRESS);
00355 }
00356 }
00357
00358 /**
00359  * @brief Wake up Idd measurement componen
00360  * @retval None
00361  */
00362 void BSP_IDD_WakeUp(void)
00363 {
00364     if(IddDrv->WakeUp != NULL)
00365     {
00366         IddDrv->WakeUp(IDD_I2C_ADDRESS);
00367     }
00368 }
00369
00370 /**
00371  * @}
00372  */
00373
00374 /**
00375  * @}
00376  */
00377
00378 /**
00379  * @}
00380  */
00381
00382 /**
00383  * @}
00384  */
00385
00386 /***** (C) COPYRIGHT STMi
00387 croelectronics *****END OF FILE*****/
```

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

Modules

## BSP

Define for STM32L476G\_DISCOVERY board. [More...](#)

## Modules

---

STM32L476G-DISCOVERY
----------------------

---



## Detailed Description

Define for STM32L476G\_DISCOVERY board.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				
<b>STM32L476G- DISCOVERY</b>				<a href="#">Modules</a>
<a href="#">BSP</a>				

## Modules

---

### **STM32L476G-DISCOVERY Common**

### **STM32L476G-DISCOVERY ACCELEROMETER**

### **STM32L476G-DISCOVERY AUDIO**

This file includes the low layer driver for cs43l22 Audio Codec available on STM32L476G-Discovery board(MB1184).

---

### **STM32L476G-DISCOVERY COMPASS**

### **STM32L476G-DISCOVERY GLASS LCD**

This file includes the LCD Glass driver for LCD Module of STM32L476G-DISCOVERY board.

---

### **STM32L476G-DISCOVERY GYROSCOPE**

### **STM32L476G-DISCOVERY IDD**

This file includes the Idd driver for STM32L476G-DISCOVERY board.

---

### **STM32L476G-DISCOVERY QSPI**

---

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Modules](#)

## Exported Constants

[STM32L476G-DISCOVERY Common](#)

## Modules

<b>BATTERY Detection Constants</b>
<b>LED Constants</b>
<b>BUTTON Constants</b>
<b>BUS Constants</b>

---

Generated on Tue Jun 23 2015 17:15:45 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

Modules

## STM32L476G- DISCOVERY GLASS LCD

[STM32L476G-DISCOVERY](#)

This file includes the LCD Glass driver for LCD Module of STM32L476G-DISCOVERY board. [More...](#)

# Modules

Private Constants
Private Variables
Private Functions
Exported Types
Exported Constants
Exported Functions

## Detailed Description

This file includes the LCD Glass driver for LCD Module of STM32L476G-DISCOVERY board.

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1



# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

[Modules](#)

## Exported Types

[STM32L476G-DISCOVERY IDD](#)

## Modules

---

### IDD Config

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1

# STM32L476G-Discovery BSP User Manual

<a href="#">Main Page</a>	<a href="#">Modules</a>	<a href="#">Data Structures</a>	<a href="#">Files</a>	
<a href="#">Directories</a>				

Modules

## STM32L476G- DISCOVERY QSPI

[STM32L476G-DISCOVERY](#)

## Modules

<b>Private Variables</b>
<b>Private Functions</b>
<b>Exported Constants</b>
<b>Exported Types</b>
<b>Exported Functions</b>

---

Generated on Tue Jun 23 2015 17:15:46 for STM32L476G-Discovery  
BSP User Manual by doxygen 1.7.6.1