

STM32F2xx_Nucleo_144 BSP User Manual

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

STM32F2XX NUCLEO 144 LOW LEVEL Private TypesDefinitions

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

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

STM32F2XX NUCLEO 144 LOW LEVEL Private Macros

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

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

STM32F2XX NUCLEO 144 LOW LEVEL Exported Macros

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

Main Page		Modules		Files		Directories	
File List		Globals					
All		Functions		Variables		Enumerations	
		Enumerator		Defines			
_		a		b		g	
		h		j		k	
		l		n		o	
		s		u			

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

- _ -

- __STM32F2xx_NUCLEO_BSP_VERSION : [stm32f2xx_nucleo_144.c](#)
- __STM32F2xx_NUCLEO_BSP_VERSION_MAIN : [stm32f2xx_nucleo_144.c](#)
- __STM32F2xx_NUCLEO_BSP_VERSION_RC : [stm32f2xx_nucleo_144.c](#)
- __STM32F2xx_NUCLEO_BSP_VERSION_SUB1 : [stm32f2xx_nucleo_144.c](#)
- __STM32F2xx_NUCLEO_BSP_VERSION_SUB2 : [stm32f2xx_nucleo_144.c](#)

- a -

- ADCx_DeInit() : [stm32f2xx_nucleo_144.c](#)
- ADCx_Init() : [stm32f2xx_nucleo_144.c](#)
- ADCx_MspDeInit() : [stm32f2xx_nucleo_144.c](#)
- ADCx_MspInit() : [stm32f2xx_nucleo_144.c](#)

- b -

- BSP_GetVersion() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)

- BSP_JOY_DeInit() : [stm32f2xx_nucleo_144.h](#) ,
[stm32f2xx_nucleo_144.c](#)
- BSP_JOY_GetState() : [stm32f2xx_nucleo_144.c](#) ,
[stm32f2xx_nucleo_144.h](#)
- BSP_JOY_Init() : [stm32f2xx_nucleo_144.h](#) ,
[stm32f2xx_nucleo_144.c](#)
- BSP_LED_DeInit() : [stm32f2xx_nucleo_144.c](#) ,
[stm32f2xx_nucleo_144.h](#)
- BSP_LED_Init() : [stm32f2xx_nucleo_144.c](#) ,
[stm32f2xx_nucleo_144.h](#)
- BSP_LED_Off() : [stm32f2xx_nucleo_144.c](#) ,
[stm32f2xx_nucleo_144.h](#)
- BSP_LED_On() : [stm32f2xx_nucleo_144.h](#) ,
[stm32f2xx_nucleo_144.c](#)
- BSP_LED_Toggle() : [stm32f2xx_nucleo_144.c](#) ,
[stm32f2xx_nucleo_144.h](#)
- BSP_PB_DeInit() : [stm32f2xx_nucleo_144.c](#) ,
[stm32f2xx_nucleo_144.h](#)
- BSP_PB_GetState() : [stm32f2xx_nucleo_144.c](#) ,
[stm32f2xx_nucleo_144.h](#)
- BSP_PB_Init() : [stm32f2xx_nucleo_144.c](#) ,
[stm32f2xx_nucleo_144.h](#)
- BUTTON_IRQn : [stm32f2xx_nucleo_144.c](#)
- BUTTON_KEY : [stm32f2xx_nucleo_144.h](#)
- BUTTON_MODE_EXTI : [stm32f2xx_nucleo_144.h](#)
- BUTTON_MODE_GPIO : [stm32f2xx_nucleo_144.h](#)
- BUTTON_PIN : [stm32f2xx_nucleo_144.c](#)
- BUTTON_PORT : [stm32f2xx_nucleo_144.c](#)
- Button_TypeDef : [stm32f2xx_nucleo_144.h](#)
- BUTTON_USER : [stm32f2xx_nucleo_144.h](#)
- ButtonMode_TypeDef : [stm32f2xx_nucleo_144.h](#)
- BUTTONn : [stm32f2xx_nucleo_144.h](#)
- BUTTONx_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- BUTTONx_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)

- GPIO_PIN : [stm32f2xx_nucleo_144.c](#)
- GPIO_PORT : [stm32f2xx_nucleo_144.c](#)

- h -

- hnucleo_Adc : [stm32f2xx_nucleo_144.c](#)
- hnucleo_Spi : [stm32f2xx_nucleo_144.c](#)

- j -

- JOY_DOWN : [stm32f2xx_nucleo_144.h](#)
- JOY_LEFT : [stm32f2xx_nucleo_144.h](#)
- JOY_NONE : [stm32f2xx_nucleo_144.h](#)
- JOY_RIGHT : [stm32f2xx_nucleo_144.h](#)
- JOY_SEL : [stm32f2xx_nucleo_144.h](#)
- JOY_UP : [stm32f2xx_nucleo_144.h](#)
- JOYState_TypeDef : [stm32f2xx_nucleo_144.h](#)

- k -

- KEY_BUTTON_EXTI_IRQn : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_EXTI_LINE : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_PIN : [stm32f2xx_nucleo_144.h](#)

- l -

- LCD_CS_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_HIGH : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_LOW : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_PIN : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)

- LCD_DC_HIGH : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_LOW : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_PIN : [stm32f2xx_nucleo_144.h](#)
- LCD_Delay() : [stm32f2xx_nucleo_144.c](#)
- LCD_IO_Init() : [stm32f2xx_nucleo_144.c](#)
- LCD_IO_WriteData() : [stm32f2xx_nucleo_144.c](#)
- LCD_IO_WriteMultipleData() : [stm32f2xx_nucleo_144.c](#)
- LCD_IO_WriteReg() : [stm32f2xx_nucleo_144.c](#)
- LED1 : [stm32f2xx_nucleo_144.h](#)
- LED1_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LED1_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LED1_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LED1_PIN : [stm32f2xx_nucleo_144.h](#)
- LED2 : [stm32f2xx_nucleo_144.h](#)
- LED2_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LED2_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LED2_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LED2_PIN : [stm32f2xx_nucleo_144.h](#)
- LED3 : [stm32f2xx_nucleo_144.h](#)
- LED3_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LED3_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LED3_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LED3_PIN : [stm32f2xx_nucleo_144.h](#)
- LED_BLUE : [stm32f2xx_nucleo_144.h](#)
- LED_GREEN : [stm32f2xx_nucleo_144.h](#)
- LED_RED : [stm32f2xx_nucleo_144.h](#)
- Led_TypeDef : [stm32f2xx_nucleo_144.h](#)
- LEDn : [stm32f2xx_nucleo_144.h](#)
- LEDx_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LEDx_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)

- n -

- NUCLEO_ADCx : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_CHANNEL : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)

- NUCLEO_ADCx_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_GPIO_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CS_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CS_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CS_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CS_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_MOSI_AF : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_MOSI_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MOSI_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_AF : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_TIMEOUT_MAX : [stm32f2xx_nucleo_144.h](#)

- 0 -

- OTG_FS1_OVER_CURRENT_PIN : [stm32f2xx_nucleo_144.h](#)
- OTG_FS1_OVER_CURRENT_PORT : [stm32f2xx_nucleo_144.h](#)
- OTG_FS1_OVER_CURRENT_PORT_CLK_ENABLE :

[stm32f2xx_nucleo_144.h](#)

- OTG_FS1_POWER_SWITCH_PIN : [stm32f2xx_nucleo_144.h](#)
- OTG_FS1_POWER_SWITCH_PORT : [stm32f2xx_nucleo_144.h](#)
- OTG_FS1_POWER_SWITCH_PORT_CLK_ENABLE :
[stm32f2xx_nucleo_144.h](#)

- S -

- sConfig : [stm32f2xx_nucleo_144.c](#)
- SD_CS_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- SD_CS_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- SD_CS_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- SD_CS_HIGH : [stm32f2xx_nucleo_144.h](#)
- SD_CS_LOW : [stm32f2xx_nucleo_144.h](#)
- SD_CS_PIN : [stm32f2xx_nucleo_144.h](#)
- SD_DUMMY_BYTE : [stm32f2xx_nucleo_144.c](#)
- SD_IO_CSState() : [stm32f2xx_nucleo_144.c](#)
- SD_IO_Init() : [stm32f2xx_nucleo_144.c](#)
- SD_IO_WriteByte() : [stm32f2xx_nucleo_144.c](#)
- SD_IO_WriteReadData() : [stm32f2xx_nucleo_144.c](#)
- SD_NO_RESPONSE_EXPECTED : [stm32f2xx_nucleo_144.c](#)
- SPIx__CS_HIGH : [stm32f2xx_nucleo_144.h](#)
- SPIx__CS_LOW : [stm32f2xx_nucleo_144.h](#)
- SPIx_Error() : [stm32f2xx_nucleo_144.c](#)
- SPIx_Init() : [stm32f2xx_nucleo_144.c](#)
- SPIx_MspInit() : [stm32f2xx_nucleo_144.c](#)
- SPIx_Write() : [stm32f2xx_nucleo_144.c](#)
- SPIx_WriteReadData() : [stm32f2xx_nucleo_144.c](#)
- SpixTimeout : [stm32f2xx_nucleo_144.c](#)

- U -

- USER_BUTTON_EXTI_IRQn : [stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_EXTI_LINE : [stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_GPIO_CLK_DISABLE :
[stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_GPIO_CLK_ENABLE :
[stm32f2xx_nucleo_144.h](#)

- USER_BUTTON_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_PIN : [stm32f2xx_nucleo_144.h](#)

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

Main Page		Modules		Files	Directories				
File List		Globals							
All		Functions		Variables		Enumerations		Enumerator	Defines
a	b	l	s						

- a -

- ADCx_DeInit() : [stm32f2xx_nucleo_144.c](#)
- ADCx_Init() : [stm32f2xx_nucleo_144.c](#)
- ADCx_MspDeInit() : [stm32f2xx_nucleo_144.c](#)
- ADCx_MspInit() : [stm32f2xx_nucleo_144.c](#)

- b -

- BSP_GetVersion() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)
- BSP_JOY_DeInit() : [stm32f2xx_nucleo_144.h](#) , [stm32f2xx_nucleo_144.c](#)
- BSP_JOY_GetState() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)
- BSP_JOY_Init() : [stm32f2xx_nucleo_144.h](#) , [stm32f2xx_nucleo_144.c](#)
- BSP_LED_DeInit() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)
- BSP_LED_Init() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)
- BSP_LED_Off() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)
- BSP_LED_On() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)

- BSP_LED_Toggle() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)
- BSP_PB_DeInit() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)
- BSP_PB_GetState() : [stm32f2xx_nucleo_144.c](#) , [stm32f2xx_nucleo_144.h](#)
- BSP_PB_Init() : [stm32f2xx_nucleo_144.h](#) , [stm32f2xx_nucleo_144.c](#)

- I -

- LCD_Delay() : [stm32f2xx_nucleo_144.c](#)
- LCD_IO_Init() : [stm32f2xx_nucleo_144.c](#)
- LCD_IO_WriteData() : [stm32f2xx_nucleo_144.c](#)
- LCD_IO_WriteMultipleData() : [stm32f2xx_nucleo_144.c](#)
- LCD_IO_WriteReg() : [stm32f2xx_nucleo_144.c](#)

- S -

- SD_IO_CSState() : [stm32f2xx_nucleo_144.c](#)
- SD_IO_Init() : [stm32f2xx_nucleo_144.c](#)
- SD_IO_WriteByte() : [stm32f2xx_nucleo_144.c](#)
- SD_IO_WriteReadData() : [stm32f2xx_nucleo_144.c](#)
- SPIx_Error() : [stm32f2xx_nucleo_144.c](#)
- SPIx_Init() : [stm32f2xx_nucleo_144.c](#)
- SPIx_MspInit() : [stm32f2xx_nucleo_144.c](#)
- SPIx_Write() : [stm32f2xx_nucleo_144.c](#)
- SPIx_WriteReadData() : [stm32f2xx_nucleo_144.c](#)

STM32F2xx_Nucleo_144 BSP User Manual

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

- BUTTON_IRQn : [stm32f2xx_nucleo_144.c](#)
- BUTTON_PIN : [stm32f2xx_nucleo_144.c](#)
- BUTTON_PORT : [stm32f2xx_nucleo_144.c](#)
- GPIO_PIN : [stm32f2xx_nucleo_144.c](#)
- GPIO_PORT : [stm32f2xx_nucleo_144.c](#)
- hnucleo_Adc : [stm32f2xx_nucleo_144.c](#)
- hnucleo_Spi : [stm32f2xx_nucleo_144.c](#)
- sConfig : [stm32f2xx_nucleo_144.c](#)
- SpixTimeout : [stm32f2xx_nucleo_144.c](#)

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

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

- Button_TypeDef : [stm32f2xx_nucleo_144.h](#)
- ButtonMode_TypeDef : [stm32f2xx_nucleo_144.h](#)
- JOYState_TypeDef : [stm32f2xx_nucleo_144.h](#)
- Led_TypeDef : [stm32f2xx_nucleo_144.h](#)

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

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

- `BUTTON_KEY` : [stm32f2xx_nucleo_144.h](#)
- `BUTTON_MODE_EXTI` : [stm32f2xx_nucleo_144.h](#)
- `BUTTON_MODE_GPIO` : [stm32f2xx_nucleo_144.h](#)
- `BUTTON_USER` : [stm32f2xx_nucleo_144.h](#)
- `JOY_DOWN` : [stm32f2xx_nucleo_144.h](#)
- `JOY_LEFT` : [stm32f2xx_nucleo_144.h](#)
- `JOY_NONE` : [stm32f2xx_nucleo_144.h](#)
- `JOY_RIGHT` : [stm32f2xx_nucleo_144.h](#)
- `JOY_SEL` : [stm32f2xx_nucleo_144.h](#)
- `JOY_UP` : [stm32f2xx_nucleo_144.h](#)
- `LED1` : [stm32f2xx_nucleo_144.h](#)
- `LED2` : [stm32f2xx_nucleo_144.h](#)
- `LED3` : [stm32f2xx_nucleo_144.h](#)
- `LED_BLUE` : [stm32f2xx_nucleo_144.h](#)
- `LED_GREEN` : [stm32f2xx_nucleo_144.h](#)
- `LED_RED` : [stm32f2xx_nucleo_144.h](#)

STM32F2xx_Nucleo_144 BSP User Manual

Main Page		Modules		Files		Directories			
File List		Globals							
All	Functions		Variables		Enumerations		Enumerator		Defines
—	b	k	l	n	o	s	u		

- _ -

- __STM32F2xx_NUCLEO_BSP_VERSION : [stm32f2xx_nucleo_144.c](#)
- __STM32F2xx_NUCLEO_BSP_VERSION_MAIN : [stm32f2xx_nucleo_144.c](#)
- __STM32F2xx_NUCLEO_BSP_VERSION_RC : [stm32f2xx_nucleo_144.c](#)
- __STM32F2xx_NUCLEO_BSP_VERSION_SUB1 : [stm32f2xx_nucleo_144.c](#)
- __STM32F2xx_NUCLEO_BSP_VERSION_SUB2 : [stm32f2xx_nucleo_144.c](#)

- b -

- BUTTONn : [stm32f2xx_nucleo_144.h](#)
- BUTTONx_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- BUTTONx_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)

- k -

- KEY_BUTTON_EXTI_IRQn : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_EXTI_LINE : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)

- KEY_BUTTON_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- KEY_BUTTON_PIN : [stm32f2xx_nucleo_144.h](#)

- l -

- LCD_CS_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_HIGH : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_LOW : [stm32f2xx_nucleo_144.h](#)
- LCD_CS_PIN : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_HIGH : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_LOW : [stm32f2xx_nucleo_144.h](#)
- LCD_DC_PIN : [stm32f2xx_nucleo_144.h](#)
- LED1_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LED1_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LED1_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LED1_PIN : [stm32f2xx_nucleo_144.h](#)
- LED2_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LED2_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LED2_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LED2_PIN : [stm32f2xx_nucleo_144.h](#)
- LED3_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LED3_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- LED3_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- LED3_PIN : [stm32f2xx_nucleo_144.h](#)
- LEDn : [stm32f2xx_nucleo_144.h](#)
- LEDx_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- LEDx_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)

- n -

- NUCLEO_ADCx : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_CHANNEL : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)

- NUCLEO_ADCx_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_GPIO_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_ADCx_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CS_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CS_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CS_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_CS_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_MOSI_AF : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_MOSI_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MISO_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_MOSI_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_AF : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_SCK_PIN : [stm32f2xx_nucleo_144.h](#)
- NUCLEO_SPIx_TIMEOUT_MAX : [stm32f2xx_nucleo_144.h](#)

- 0 -

- OTG_FS1_OVER_CURRENT_PIN : [stm32f2xx_nucleo_144.h](#)
- OTG_FS1_OVER_CURRENT_PORT : [stm32f2xx_nucleo_144.h](#)

- OTG_FS1_OVER_CURRENT_PORT_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- OTG_FS1_POWER_SWITCH_PIN : [stm32f2xx_nucleo_144.h](#)
- OTG_FS1_POWER_SWITCH_PORT : [stm32f2xx_nucleo_144.h](#)
- OTG_FS1_POWER_SWITCH_PORT_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)

- S -

- SD_CS_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- SD_CS_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- SD_CS_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- SD_CS_HIGH : [stm32f2xx_nucleo_144.h](#)
- SD_CS_LOW : [stm32f2xx_nucleo_144.h](#)
- SD_CS_PIN : [stm32f2xx_nucleo_144.h](#)
- SD_DUMMY_BYTE : [stm32f2xx_nucleo_144.c](#)
- SD_NO_RESPONSE_EXPECTED : [stm32f2xx_nucleo_144.c](#)
- SPIx__CS_HIGH : [stm32f2xx_nucleo_144.h](#)
- SPIx__CS_LOW : [stm32f2xx_nucleo_144.h](#)

- U -

- USER_BUTTON_EXTI_IRQn : [stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_EXTI_LINE : [stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_GPIO_CLK_DISABLE : [stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_GPIO_CLK_ENABLE : [stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_GPIO_PORT : [stm32f2xx_nucleo_144.h](#)
- USER_BUTTON_PIN : [stm32f2xx_nucleo_144.h](#)

STM32F2xx_Nucleo_144 BSP User Manual

Main Page		Modules		Files	Directories				
File List		Globals							
Drivers	BSP	STM32F2xx_Nucleo_144							
							Defines	Functions	Variables

stm32f2xx_nucleo_144.c File Reference

This file provides set of firmware functions to manage: [More...](#)

```
#include "stm32f2xx_nucleo_144.h"
```

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

Defines

#define	__STM32F2xx_NUCLEO_BSP_VERSION_MAIN	(0x01)	STM32F2xx NUCLEO BSP Driver version number V1.0.1.
#define	__STM32F2xx_NUCLEO_BSP_VERSION_SUB1	(0x00)	
#define	__STM32F2xx_NUCLEO_BSP_VERSION_SUB2	(0x01)	
#define	__STM32F2xx_NUCLEO_BSP_VERSION_RC	(0x00)	
#define	__STM32F2xx_NUCLEO_BSP_VERSION		
#define	SD_DUMMY_BYTE	0xFF	LINK SD Card.
#define	SD_NO_RESPONSE_EXPECTED	0x80	

Functions

static void	SPIx_Init (void)	Initializes SPI HAL.
static void	SPIx_Write (uint8_t Value)	SPI Write a byte to device.
static void	SPIx_Error (void)	SPI error treatment function.
static void	SPIx_MspInit (SPI_HandleTypeDef *hspi)	Initializes SPI MSP.
void	SD_IO_Init (void)	Initializes the SD Card and put it into StandBy State (Ready for data transfer).
void	SD_IO_CSState (uint8_t val)	Set the SD_CS pin.
void	SD_IO_WriteReadData (const uint8_t *DataIn, uint8_t *DataOut, uint16_t DataLength)	Write a byte on the SD.
uint8_t	SD_IO_WriteByte (uint8_t Data)	Writes a byte on the SD.
void	LCD_IO_Init (void)	Initializes the LCD.
void	LCD_IO_WriteData (uint8_t Data)	Writes data to select the LCD register.
void	LCD_IO_WriteMultipleData (uint8_t *pData, uint32_t Size)	Write register value.
void	LCD_IO_WriteReg (uint8_t LCDReg)	Writes command to select the LCD register.
void	LCD_Delay (uint32_t Delay)	Wait for loop in ms.
static void	ADCx_Init (void)	Initializes ADC HAL.

static void	ADCx_DeInit (void)	Initializes ADC HAL.
static void	ADCx_MspInit (ADC_HandleTypeDef *hadc)	Initializes ADC MSP.
static void	ADCx_MspDeInit (ADC_HandleTypeDef *hadc)	DeInitializes ADC MSP.
uint32_t	BSP_GetVersion (void)	This method returns the STM32F2xx NUCLEO BSP Driver revision.
void	BSP_LED_Init (Led_TypeDef Led)	Configures LED GPIO.
void	BSP_LED_DeInit (Led_TypeDef Led)	DeInit LEDs.
void	BSP_LED_On (Led_TypeDef Led)	Turns selected LED On.
void	BSP_LED_Off (Led_TypeDef Led)	Turns selected LED Off.
void	BSP_LED_Toggle (Led_TypeDef Led)	Toggles the selected LED.
void	BSP_PB_Init (Button_TypeDef Button, ButtonMode_TypeDef ButtonMode)	Configures Button GPIO and EXTI Line.
void	BSP_PB_DeInit (Button_TypeDef Button)	Push Button DeInit.
uint32_t	BSP_PB_GetState (Button_TypeDef Button)	Returns the selected Button state.
static void	SPIx_WriteReadData (const uint8_t *DataIn, uint8_t *DataOut, uint16_t DataLegnth)	SPI Write a byte to device.
uint8_t	BSP_JOY_Init (void)	Configures joystick available on adafruit 1.8" TFT shield managed through ADC to detect motion.

void **BSP_JOY_DeInit** (void)
DeInit joystick GPIOs.

JOYState_TypeDef **BSP_JOY_GetState** (void)
Returns the Joystick key pressed.

Variables

GPIO_TypeDef *	GPIO_PORT [LEDn] = {LED1_GPIO_PORT, LED2_GPIO_PORT, LED3_GPIO_PORT}
const uint16_t	GPIO_PIN [LEDn] = {LED1_PIN, LED2_PIN, LED3_PIN}
GPIO_TypeDef *	BUTTON_PORT [BUTTONn] = {USER_BUTTON_GPIO_PORT}
const uint16_t	BUTTON_PIN [BUTTONn] = {USER_BUTTON_PIN}
const uint8_t	BUTTON_IRQn [BUTTONn] = {USER_BUTTON_EXTI_IRQn}
uint32_t	SpixTimeout = NUCLEO_SPIx_TIMEOUT_MAX BUS variables.
static SPI_HandleTypeDef	hnucleo_Spi
static ADC_HandleTypeDef	hnucleo_Adc
static ADC_ChannelConfTypeDef	sConfig

Detailed Description

This file provides set of firmware functions to manage:

Author:

MCD Application Team

Version:

V1.0.0

Date:

20-November-2015

- LEDs and push-button available on STM32F2XX-Nucleo-144 Kit from STMicroelectronics
- LCD, joystick and microSD available on Adafruit 1.8" TFT LCD shield (reference ID 802)

Attention:

© COPYRIGHT(c) 2017 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 [stm32f2xx_nucleo_144.c](#).

STM32F2xx_Nucleo_144 BSP User Manual

Main Page		Modules		Files	Directories				
File List		Globals							
Drivers	BSP	STM32F2xx_Nucleo_144							
							Defines	Enumerations	Functions

stm32f2xx_nucleo_144.h File Reference

This file contains definitions for: [More...](#)

```
#include "stm32f2xx_hal.h"
```

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

Defines

```
#define LEDn 3
#define LED1_PIN GPIO_PIN_0
#define LED1_GPIO_PORT GPIOB
#define LED1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define LED1_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define LED2_PIN GPIO_PIN_7
#define LED2_GPIO_PORT GPIOB
#define LED2_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define LED2_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define LED3_PIN GPIO_PIN_14
#define LED3_GPIO_PORT GPIOB
#define LED3_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define LED3_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_CLK_DISABLE()
#define LEDx_GPIO_CLK_ENABLE(__INDEX__)
#define LEDx_GPIO_CLK_DISABLE(__INDEX__)
#define BUTTONn 1
#define USER_BUTTON_PIN GPIO_PIN_13
    Key push-button.
#define USER_BUTTON_GPIO_PORT GPIOC
#define USER_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC_CLK_ENABLE()
#define USER_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC_CLK_DISABLE()
#define USER_BUTTON_EXTI_LINE GPIO_PIN_13
#define USER_BUTTON_EXTI_IRQn EXTI15_10_IRQn
#define BUTTONx_GPIO_CLK_ENABLE(__INDEX__) USER_BUTTON_GPIO_CLK_ENABLE()
#define BUTTONx_GPIO_CLK_DISABLE(__INDEX__) USER_BUTTON_GPIO_CLK_DISABLE()
#define KEY_BUTTON_PIN USER_BUTTON_PIN
#define KEY_BUTTON_GPIO_PORT USER_BUTTON_GPIO_PORT
#define KEY_BUTTON_GPIO_CLK_ENABLE() USER_BUTTON_GPIO_CLK_ENABLE()
#define KEY_BUTTON_GPIO_CLK_DISABLE() USER_BUTTON_GPIO_CLK_DISABLE()
#define KEY_BUTTON_EXTI_LINE USER_BUTTON_EXTI_LINE
#define KEY_BUTTON_EXTI_IRQn USER_BUTTON_EXTI_IRQn
```

```

#define OTG_FS1_OVER_CURRENT_PIN GPIO_PIN_7
    OTG_FS1 OVER_CURRENT and POWER_SWITCH Pins defi
#define OTG_FS1_OVER_CURRENT_PORT GPIOG
#define OTG_FS1_OVER_CURRENT_PORT_CLK_ENABLE() __HA
#define OTG_FS1_POWER_SWITCH_PIN GPIO_PIN_6
#define OTG_FS1_POWER_SWITCH_PORT GPIOG
#define OTG_FS1_POWER_SWITCH_PORT_CLK_ENABLE() __HA
#define NUCLEO_SPIx SPI1
#define NUCLEO_SPIx_CLK_ENABLE() __HAL_RCC_SPI1_CLK_E
#define NUCLEO_SPIx_SCK_AF GPIO_AF5_SPI1
#define NUCLEO_SPIx_SCK_GPIO_PORT GPIOA
#define NUCLEO_SPIx_SCK_PIN GPIO_PIN_5
#define NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE() __HAL_RCC_
#define NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE() __HAL_RCC_
#define NUCLEO_SPIx_MISO_MOSI_AF GPIO_AF5_SPI1
#define NUCLEO_SPIx_MISO_MOSI_GPIO_PORT GPIOA
#define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE() __HAL
#define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISABLE() __HA
#define NUCLEO_SPIx_MISO_PIN GPIO_PIN_6
#define NUCLEO_SPIx_MOSI_PIN GPIO_PIN_7
#define NUCLEO_SPIx_TIMEOUT_MAX 1000
#define NUCLEO_SPIx_CS_GPIO_PORT GPIOD
#define NUCLEO_SPIx_CS_PIN GPIO_PIN_14
#define NUCLEO_SPIx_CS_GPIO_CLK_ENABLE() __HAL_RCC_G
#define NUCLEO_SPIx_CS_GPIO_CLK_DISABLE() __HAL_RCC_C
#define SPIx__CS_LOW() HAL_GPIO_WritePin(NUCLEO_SPIx_CS
    NUCLEO_SPIx_CS_PIN, GPIO_PIN_RESET)
#define SPIx__CS_HIGH() HAL_GPIO_WritePin(NUCLEO_SPIx_CS
    NUCLEO_SPIx_CS_PIN, GPIO_PIN_SET)
#define SD_CS_LOW() HAL_GPIO_WritePin(SD_CS_GPIO_PORT,
    SD Control Lines management.
#define SD_CS_HIGH() HAL_GPIO_WritePin(SD_CS_GPIO_PORT,
#define LCD_CS_LOW() HAL_GPIO_WritePin(LCD_CS_GPIO_POR
    GPIO_PIN_RESET)

```

LCD Control Lines management.

```
#define LCD_CS_HIGH() HAL_GPIO_WritePin(LCD_CS_GPIO_PORT, LCD_CS_GPIO_PIN, GPIO_PIN_SET)
#define LCD_DC_LOW() HAL_GPIO_WritePin(LCD_DC_GPIO_PORT, LCD_DC_GPIO_PIN, GPIO_PIN_RESET)
#define LCD_DC_HIGH() HAL_GPIO_WritePin(LCD_DC_GPIO_PORT, LCD_DC_GPIO_PIN, GPIO_PIN_SET)
#define SD_CS_PIN GPIO_PIN_14
SD Control Interface pins (shield D4)
#define SD_CS_GPIO_PORT GPIOF
#define SD_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOF_CLK_ENABLE()
#define SD_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPIOF_CLK_DISABLE()
#define LCD_CS_PIN GPIO_PIN_14
LCD Control Interface pins (shield D10)
#define LCD_CS_GPIO_PORT GPIOD
#define LCD_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define LCD_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPIOD_CLK_DISABLE()
#define LCD_DC_PIN GPIO_PIN_12
LCD Data/Command Interface pins (shield D8)
#define LCD_DC_GPIO_PORT GPIOF
#define LCD_DC_GPIO_CLK_ENABLE() __HAL_RCC_GPIOF_CLK_ENABLE()
#define LCD_DC_GPIO_CLK_DISABLE() __HAL_RCC_GPIOF_CLK_DISABLE()
#define NUCLEO_ADCx ADC3
ADCx Interface pins used to detect motion of Joystick available
#define NUCLEO_ADCx_CLK_ENABLE() __HAL_RCC_ADC3_CLK_ENABLE()
#define NUCLEO_ADCx_CLK_DISABLE() __HAL_RCC_ADC3_CLK_DISABLE()
#define NUCLEO_ADCx_CHANNEL ADC_CHANNEL_9
#define NUCLEO_ADCx_GPIO_PORT GPIOF
#define NUCLEO_ADCx_GPIO_PIN GPIO_PIN_3
#define NUCLEO_ADCx_GPIO_CLK_ENABLE() __HAL_RCC_GPIOF_CLK_ENABLE()
#define NUCLEO_ADCx_GPIO_CLK_DISABLE() __HAL_RCC_GPIOF_CLK_DISABLE()
```

Enumerations

```
enum Led_TypeDef {  
    LED1 = 0, LED_GREEN = LED1, LED2 = 1, LED_BLUE =  
    LED2,  
    LED3 = 2, LED_RED = LED3  
}
```

```
enum Button_TypeDef { BUTTON_USER = 0, BUTTON_KEY =  
    BUTTON_USER }
```

```
enum ButtonMode_TypeDef { BUTTON_MODE_GPIO = 0,  
    BUTTON_MODE_EXTI = 1 }
```

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


Functions

uint32_t	BSP_GetVersion (void) This method returns the STM32F2xx NUCLEO BSP Driver revision.
void	BSP_LED_Init (Led_TypeDef Led) Configures LED GPIO.
void	BSP_LED_DeInit (Led_TypeDef Led) DeInit LEDs.
void	BSP_LED_On (Led_TypeDef Led) Turns selected LED On.
void	BSP_LED_Off (Led_TypeDef Led) Turns selected LED Off.
void	BSP_LED_Toggle (Led_TypeDef Led) Toggles the selected LED.
void	BSP_PB_Init (Button_TypeDef Button, ButtonMode_TypeDef ButtonMode) Configures Button GPIO and EXTI Line.
void	BSP_PB_DeInit (Button_TypeDef Button) Push Button DeInit.
uint32_t	BSP_PB_GetState (Button_TypeDef Button) Returns the selected Button state.
uint8_t	BSP_JOY_Init (void) Configures joystick available on adafruit 1.8" TFT shield managed through ADC to detect motion.
JOYState_TypeDef	BSP_JOY_GetState (void) Returns the Joystick key pressed.
void	BSP_JOY_DeInit (void) DeInit joystick GPIOs.

Detailed Description

This file contains definitions for:

Author:

MCD Application Team

Version:

V1.0.0

Date:

20-November-2015

- LEDs and push-button available on STM32F2XX-Nucleo-144 Kit from STMicroelectronics
- LCD, joystick and microSD available on Adafruit 1.8" TFT LCD shield (reference ID 802)

Attention:

© COPYRIGHT(c) 2017 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 [stm32f2xx_nucleo_144.h](#).

STM32F2xx_Nucleo_144 BSP User Manual

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

Modules

Here is a list of all modules:

- **BSP**
 - **STM32F2XX NUCLEO 144**
 - **STM32F2XX NUCLEO 144 LOW LEVEL**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Private TypesDefinitions**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Private Defines**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Private Macros**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Private Variables**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Private FunctionPrototypes**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Private Functions**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Exported Types**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Exported Constants**
 - **STM32F2XX NUCLEO 144 LOW LEVEL LED**
 - **STM32F2XX NUCLEO 144 LOW LEVEL BUTTON**
 - **STM32F2XX NUCLEO 144 LOW LEVEL BUS**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Exported Macros**
 - **STM32F2XX NUCLEO 144 LOW LEVEL Exported**

Functions

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

Main Page	Modules	Files	Directories	
File List	Globals			

File List

Here is a list of all files with brief descriptions:

stm32f2xx_nucleo_144.c [code]	This file provides set of firmware functions to manage:
stm32f2xx_nucleo_144.h [code]	This file contains definitions for:

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

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

Directories

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

- **Drivers**
 - **BSP**
 - **STM32F2xx_Nucleo_144**

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

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

STM32F2XX NUCLEO 144 LOW LEVEL

[STM32F2XX NUCLEO 144](#)

This file provides set of firmware functions to manage Leds and push-button available on STM32F2xx-Nucleo Kit from STMicroelectronics.

[More...](#)

Modules

STM32F2XX NUCLEO 144 LOW LEVEL Private TypesDefinitions
STM32F2XX NUCLEO 144 LOW LEVEL Private Defines
STM32F2XX NUCLEO 144 LOW LEVEL Private Macros
STM32F2XX NUCLEO 144 LOW LEVEL Private Variables
STM32F2XX NUCLEO 144 LOW LEVEL Private FunctionPrototypes
STM32F2XX NUCLEO 144 LOW LEVEL Private Functions
STM32F2XX NUCLEO 144 LOW LEVEL Exported Types
STM32F2XX NUCLEO 144 LOW LEVEL Exported Constants
STM32F2XX NUCLEO 144 LOW LEVEL Exported Macros
STM32F2XX NUCLEO 144 LOW LEVEL Exported Functions

Detailed Description

This file provides set of firmware functions to manage Leds and push-button available on STM32F2xx-Nucleo Kit from STMicroelectronics.

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Defines](#)

STM32F2XX NUCLEO 144 LOW LEVEL Private Defines

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Defines

#define	__STM32F2xx_NUCLEO_BSP_VERSION_MAIN	(0x01)
	STM32F2xx NUCLEO BSP Driver version number V1.0.1.	
#define	__STM32F2xx_NUCLEO_BSP_VERSION_SUB1	(0x00)
#define	__STM32F2xx_NUCLEO_BSP_VERSION_SUB2	(0x01)
#define	__STM32F2xx_NUCLEO_BSP_VERSION_RC	(0x00)
#define	__STM32F2xx_NUCLEO_BSP_VERSION	
#define	SD_DUMMY_BYTE	0xFF
	LINK SD Card.	
#define	SD_NO_RESPONSE_EXPECTED	0x80

Define Documentation

```
#define __STM32F2xx_NUCLEO_BSP_VERSION
```

Value:

```
(( (__STM32F2xx_NUCLEO_BSP_VERSION_MAIN << 24) |\
(_STM32F2xx_NUCLEO_BSP_VERSION_SUB1 << 16) |\
(_STM32F2xx_NUCLEO_BSP_VERSION_SUB2 << 8 ) |\
(_STM32F2xx_NUCLEO_BSP_VERSION_RC)))
```

Definition at line 79 of file `stm32f2xx_nucleo_144.c`.

Referenced by **BSP_GetVersion()**.

```
#define __STM32F2xx_NUCLEO_BSP_VERSION_MAIN (0x01)
```

STM32F2xx NUCLEO BSP Driver version number V1.0.1.

[31:24] main version

Definition at line 75 of file stm32f2xx_nucleo_144.c.

```
#define __STM32F2xx_NUCLEO_BSP_VERSION_RC (0x00)
```

[7:0] release candidate

Definition at line 78 of file stm32f2xx_nucleo_144.c.

```
#define __STM32F2xx_NUCLEO_BSP_VERSION_SUB1 (0x00)
```

[23:16] sub1 version

Definition at line **76** of file **stm32f2xx_nucleo_144.c**.

#define __STM32F2xx_NUCLEO_BSP_VERSION_SUB2 (0x01)

[15:8] sub2 version

Definition at line **77** of file **stm32f2xx_nucleo_144.c**.

#define SD_DUMMY_BYTE 0xFF

LINK SD Card.

Definition at line **87** of file **stm32f2xx_nucleo_144.c**.

Referenced by **SD_IO_Init()**.

#define SD_NO_RESPONSE_EXPECTED 0x80

Definition at line **88** of file **stm32f2xx_nucleo_144.c**.

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Functions](#)

STM32F2XX NUCLEO 144 LOW LEVEL Private FunctionPrototypes

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Functions

static void	SPIx_Init (void)	Initializes SPI HAL.
static void	SPIx_Error (void)	SPI error treatment function.
static void	SPIx_Msplnit (SPI_HandleTypeDef *hspi)	Initializes SPI MSP.
void	SD_IO_Init (void)	Initializes the SD Card and put it into StandBy State (Ready for data transfer).
void	LCD_IO_Init (void)	Initializes the LCD.
static void	ADCx_Init (void)	Initializes ADC HAL.
static void	ADCx_DeInit (void)	Initializes ADC HAL.
static void	ADCx_Msplnit (ADC_HandleTypeDef *hadc)	Initializes ADC MSP.

Function Documentation

static void [ADCx_DeInit](#) (void) [static]

Initializes ADC HAL.

Definition at line **759** of file [stm32f2xx_nucleo_144.c](#).

References [ADCx_MspDeInit\(\)](#), [hnucleo_Adc](#), and [NUCLEO_ADCx](#).

Referenced by [BSP_JOY_DeInit\(\)](#).

static void [ADCx_Init](#) (void) [static]

Initializes ADC HAL.

Definition at line **735** of file [stm32f2xx_nucleo_144.c](#).

References [ADCx_MspInit\(\)](#), [hnucleo_Adc](#), and [NUCLEO_ADCx](#).

Referenced by [BSP_JOY_Init\(\)](#).

static void [ADCx_MspInit](#) (ADC_HandleTypeDef * **hadc**) [static]

Initializes ADC MSP.

Definition at line **693** of file [stm32f2xx_nucleo_144.c](#).

References [NUCLEO_ADCx_CLK_ENABLE](#),
[NUCLEO_ADCx_GPIO_CLK_ENABLE](#),
[NUCLEO_ADCx_GPIO_PIN](#), and [NUCLEO_ADCx_GPIO_PORT](#).

Referenced by [ADCx_Init\(\)](#).

void LCD_IO_Init (void)

Initializes the LCD.

Definition at line 556 of file [stm32f2xx_nucleo_144.c](#).

References [LCD_CS_GPIO_CLK_ENABLE](#), [LCD_CS_GPIO_PORT](#), [LCD_CS_HIGH](#), [LCD_CS_PIN](#), [LCD_DC_GPIO_CLK_ENABLE](#), [LCD_DC_GPIO_PORT](#), [LCD_DC_PIN](#), and [SPIx_Init\(\)](#).

void SD_IO_Init (void)

Initializes the SD Card and put it into StandBy State (Ready for data transfer).

Definition at line 470 of file [stm32f2xx_nucleo_144.c](#).

References [LCD_CS_GPIO_CLK_ENABLE](#), [LCD_CS_GPIO_PORT](#), [LCD_CS_HIGH](#), [LCD_CS_PIN](#), [SD_CS_GPIO_CLK_ENABLE](#), [SD_CS_GPIO_PORT](#), [SD_CS_HIGH](#), [SD_CS_PIN](#), [SD_DUMMY_BYTE](#), [SD_IO_WriteByte\(\)](#), and [SPIx_Init\(\)](#).

static void SPIx_Error (void) [static]

SPI error treatment function.

Definition at line 452 of file [stm32f2xx_nucleo_144.c](#).

References [hnucleo_Spi](#), and [SPIx_Init\(\)](#).

Referenced by [SPIx_Write\(\)](#), and [SPIx_WriteReadData\(\)](#).

static void SPIx_Init (void) [static]

Initializes SPI HAL.

Definition at line **379** of file [stm32f2xx_nucleo_144.c](#).

References [hnucleo_Spi](#), [NUCLEO_SPIx](#), and [SPIx_MspInit\(\)](#).

Referenced by [LCD_IO_Init\(\)](#), [SD_IO_Init\(\)](#), and [SPIx_Error\(\)](#).

static void SPIx_MspInit (SPI_HandleTypeDef * [hspi](#)) [static]

Initializes SPI MSP.

Definition at line **344** of file [stm32f2xx_nucleo_144.c](#).

References [NUCLEO_SPIx_CLK_ENABLE](#),
[NUCLEO_SPIx_MISO_MOSI_AF](#),
[NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE](#),
[NUCLEO_SPIx_MISO_MOSI_GPIO_PORT](#),
[NUCLEO_SPIx_MISO_PIN](#), [NUCLEO_SPIx_MOSI_PIN](#),
[NUCLEO_SPIx_SCK_AF](#),
[NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE](#),
[NUCLEO_SPIx_SCK_GPIO_PORT](#), and [NUCLEO_SPIx_SCK_PIN](#).

Referenced by [SPIx_Init\(\)](#).

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Functions](#)

STM32F2XX NUCLEO 144 LOW LEVEL Private Functions

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Functions

uint32_t	BSP_GetVersion (void) This method returns the STM32F2xx NUCLEO BSP Driver revision.
void	BSP_LED_Init (Led_TypeDef Led) Configures LED GPIO.
void	BSP_LED_DeInit (Led_TypeDef Led) DeInit LEDs.
void	BSP_LED_On (Led_TypeDef Led) Turns selected LED On.
void	BSP_LED_Off (Led_TypeDef Led) Turns selected LED Off.
void	BSP_LED_Toggle (Led_TypeDef Led) Toggles the selected LED.
void	BSP_PB_Init (Button_TypeDef Button, ButtonMode_TypeDef ButtonMode) Configures Button GPIO and EXTI Line.
void	BSP_PB_DeInit (Button_TypeDef Button) Push Button DeInit.
uint32_t	BSP_PB_GetState (Button_TypeDef Button) Returns the selected Button state.
static void	SPIx_WriteReadData (const uint8_t *DataIn, uint8_t *DataOut, uint16_t DataLength) SPI Write a byte to device.
uint8_t	BSP_JOY_Init (void) Configures joystick available on adafruit 1.8" TFT shield managed through ADC to detect motion.
void	BSP_JOY_DeInit (void) DeInit joystick GPIOs.
JOYState_TypeDef	BSP_JOY_GetState (void) Returns the Joystick key pressed.

static void **SPIx_Write** (uint8_t Value)
SPI Write a byte to device.

void **SD_IO_CSState** (uint8_t val)
Set the SD_CS pin.

void **SD_IO_WriteReadData** (const uint8_t *DataIn,
uint8_t *DataOut, uint16_t DataLength)
Write a byte on the SD.

uint8_t **SD_IO_WriteByte** (uint8_t Data)
Writes a byte on the SD.

void **LCD_IO_WriteReg** (uint8_t LCDReg)
Writes command to select the LCD register.

void **LCD_IO_WriteData** (uint8_t Data)
Writes data to select the LCD register.

void **LCD_IO_WriteMultipleData** (uint8_t *pData,
uint32_t Size)
Write register value.

void **LCD_Delay** (uint32_t Delay)
Wait for loop in ms.

static void **ADCx_MspDeInit** (ADC_HandleTypeDef *hadc)
DeInitializes ADC MSP.

Function Documentation

static void [ADCx_MspDeInit](#) (ADC_HandleTypeDef * **hadc**) [static

DeInitializes ADC MSP.

Note:

ADC DeInit does not disable the GPIO clock

Definition at line **716** of file [stm32f2xx_nucleo_144.c](#).

References [NUCLEO_ADCx_CLK_DISABLE](#),
[NUCLEO_ADCx_GPIO_PIN](#), and [NUCLEO_ADCx_GPIO_PORT](#).

Referenced by [ADCx_DeInit\(\)](#).

uint32_t [BSP_GetVersion](#) (void)

This method returns the STM32F2xx NUCLEO BSP Driver revision.

Return values:

version,: 0xXYZR (8bits for each decimal, R for RC)

Definition at line **179** of file [stm32f2xx_nucleo_144.c](#).

References [__STM32F2xx_NUCLEO_BSP_VERSION](#).

void [BSP_JOY_DeInit](#) (void)

DeInit joystick GPIOs.

Note:

JOY DeInit does not disable the Mfx, just set the Mfx pins in Off mode

Definition at line **794** of file [stm32f2xx_nucleo_144.c](#).

References [ADCx_DeInit\(\)](#).

JOYState_TypeDef BSP_JOY_GetState (void)

Returns the Joystick key pressed.

Note:

To know which Joystick key is pressed we need to detect the voltage level on each key output

- None : 3.3 V / 4095
- SEL : 1.055 V / 1308
- DOWN : 0.71 V / 88
- LEFT : 3.0 V / 3720
- RIGHT : 0.595 V / 737
- UP : 1.65 V / 2046

Return values:

JOYState_TypeDef,: Code of the Joystick key pressed.

Definition at line **811** of file [stm32f2xx_nucleo_144.c](#).

References [hnucleo_Adc](#), [JOY_DOWN](#), [JOY_LEFT](#), [JOY_NONE](#), [JOY_RIGHT](#), [JOY_SEL](#), and [JOY_UP](#).

uint8_t BSP_JOY_Init (void)

Configures joystick available on adafruit 1.8" TFT shield managed through ADC to detect motion.

Return values:

Joystickstatus (0=> success, 1=> fail)

Definition at line **774** of file [stm32f2xx_nucleo_144.c](#).

References [ADCx_Init\(\)](#), [hnucleo_Adc](#), [NUCLEO_ADCx_CHANNEL](#), and [sConfig](#).

void BSP_LED_DeInit (Led_TypeDef Led)

DeInit LEDs.

Parameters:

Led,: LED to be de-init. This parameter can be one of the following values:

- LED1
- LED2
- LED3

Note:

Led DeInit does not disable the GPIO clock nor disable the Mfx

Definition at line [218](#) of file [stm32f2xx_nucleo_144.c](#).

References [GPIO_PIN](#), and [GPIO_PORT](#).

void BSP_LED_Init (Led_TypeDef Led)

Configures LED GPIO.

Parameters:

Led,: Specifies the Led to be configured. This parameter can be one of following parameters:

- LED1
- LED2
- LED3

Definition at line [192](#) of file [stm32f2xx_nucleo_144.c](#).

References [GPIO_PIN](#), [GPIO_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:

- LED1
- LED2
- LED3

Definition at line **248** of file **stm32f2xx_nucleo_144.c**.

References **GPIO_PIN**, and **GPIO_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:

- LED2

Definition at line **235** of file **stm32f2xx_nucleo_144.c**.

References **GPIO_PIN**, and **GPIO_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:

- LED1
- LED2
- LED3

Definition at line **261** of file **stm32f2xx_nucleo_144.c**.

References **GPIO_PIN**, and **GPIO_PORT**.

void BSP_PB_DeInit (Button_TypeDef Button)

Push Button DeInit.

Parameters:

Button,: Button to be configured This parameter should be: BUTTON_USER

Note:

PB DeInit does not disable the GPIO clock

Definition at line **313** of file **stm32f2xx_nucleo_144.c**.

References **BUTTON_IRQn**, **BUTTON_PIN**, and **BUTTON_PORT**.

uint32_t BSP_PB_GetState (Button_TypeDef Button)

Returns the selected Button state.

Parameters:

Button,: Specifies the Button to be checked. This parameter should be: BUTTON_USER

Return values:

The Button GPIO pin value.

Definition at line **328** of file [stm32f2xx_nucleo_144.c](#).

References [BUTTON_PIN](#), and [BUTTON_PORT](#).

```
void BSP_PB_Init ( Button_TypeDef      Button,  
                  ButtonMode_TypeDef ButtonMode  
                  )
```

Configures Button GPIO and EXTI Line.

Parameters:

- Button,:** Specifies the Button to be configured. This parameter should be: [BUTTON_USER](#)
- ButtonMode,:** Specifies Button mode. This parameter can be one of following parameters:
- [BUTTON_MODE_GPIO](#): Button will be used as simple IO
 - [BUTTON_MODE_EXTI](#): Button will be connected to EXTI line with interrupt generation capability

Definition at line **276** of file [stm32f2xx_nucleo_144.c](#).

References [BUTTON_IRQn](#), [BUTTON_MODE_EXTI](#), [BUTTON_MODE_GPIO](#), [BUTTON_PIN](#), [BUTTON_PORT](#), and [BUTTONx_GPIO_CLK_ENABLE](#).

```
void LCD_Delay ( uint32_t Delay )
```

Wait for loop in ms.

Parameters:

Delay in ms.

Definition at line **681** of file [stm32f2xx_nucleo_144.c](#).

```
void LCD_IO_WriteData ( uint8_t Data )
```

Writes data to select the LCD register.

This function must be used after `st7735_WriteReg()` function

Parameters:

Data,: data to write to the selected register.

Definition at line **606** of file `stm32f2xx_nucleo_144.c`.

References `LCD_CS_HIGH`, `LCD_CS_LOW`, `LCD_DC_HIGH`, and `SPIx_Write()`.

```
void LCD_IO_WriteMultipleData ( uint8_t * pData,  
                                uint32_t Size  
                                )
```

Write register value.

Parameters:

pData Pointer on the register value

Size Size of byte to transmit to the register

Definition at line **626** of file `stm32f2xx_nucleo_144.c`.

References `hnucleo_Spi`, `LCD_CS_HIGH`, `LCD_CS_LOW`, `LCD_DC_HIGH`, and `SPIx_Write()`.

```
void LCD_IO_WriteReg ( uint8_t LCDReg )
```

Writes command to select the LCD register.

Parameters:

LCDReg,: Address of the selected register.

Definition at line **586** of file **stm32f2xx_nucleo_144.c**.

References **LCD_CS_HIGH**, **LCD_CS_LOW**, **LCD_DC_LOW**, and **SPIx_Write()**.

void SD_IO_CSState (uint8_t val)

Set the SD_CS pin.

Parameters:

val,: pin value.

Definition at line **516** of file **stm32f2xx_nucleo_144.c**.

References **SD_CS_HIGH**, and **SD_CS_LOW**.

uint8_t SD_IO_WriteByte (uint8_t Data)

Writes a byte on the SD.

Parameters:

Data,: byte to send.

Definition at line **544** of file **stm32f2xx_nucleo_144.c**.

References **SPIx_WriteReadData()**.

Referenced by **SD_IO_Init()**.

**void SD_IO_WriteReadData (const uint8_t * DataIn,
uint8_t * DataOut,**

```
uint16_t    DataLength
)
```

Write a byte on the SD.

Parameters:

DataIn,: byte to send.
DataOut,: byte to read
DataLength,: length of data

Definition at line [534](#) of file [stm32f2xx_nucleo_144.c](#).

References [SPIx_WriteReadData\(\)](#).

```
static void SPIx_Write ( uint8_t Value ) [static]
```

SPI Write a byte to device.

Parameters:

Value,: value to be written

Definition at line [434](#) of file [stm32f2xx_nucleo_144.c](#).

References [hnucleo_Spi](#), [SPIx_Error\(\)](#), and [SpixTimeout](#).

Referenced by [LCD_IO_WriteData\(\)](#), [LCD_IO_WriteMultipleData\(\)](#), and [LCD_IO_WriteReg\(\)](#).

```
static void SPIx_WriteReadData ( const uint8_t * DataIn,
                                uint8_t *      DataOut,
                                uint16_t      DataLegnth
                                ) [static]
```

SPI Write a byte to device.

Parameters:

DataIn,: value to be written

DataOut,: value to read

DataLegnth,: length of data

Definition at line **416** of file **stm32f2xx_nucleo_144.c**.

References **hnucleo_Spi**, **SPIx_Error()**, and **SpixTimeout**.

Referenced by **SD_IO_WriteByte()**, and **SD_IO_WriteReadData()**.

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Functions](#)

STM32F2XX NUCLEO 144 LOW LEVEL Exported Functions

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Functions

uint32_t	BSP_GetVersion (void) This method returns the STM32F2xx NUCLEO BSP Driver revision.
void	BSP_LED_Init (Led_TypeDef Led) Configures LED GPIO.
void	BSP_LED_DeInit (Led_TypeDef Led) DeInit LEDs.
void	BSP_LED_On (Led_TypeDef Led) Turns selected LED On.
void	BSP_LED_Off (Led_TypeDef Led) Turns selected LED Off.
void	BSP_LED_Toggle (Led_TypeDef Led) Toggles the selected LED.
void	BSP_PB_Init (Button_TypeDef Button, ButtonMode_TypeDef ButtonMode) Configures Button GPIO and EXTI Line.
void	BSP_PB_DeInit (Button_TypeDef Button) Push Button DeInit.
uint32_t	BSP_PB_GetState (Button_TypeDef Button) Returns the selected Button state.
uint8_t	BSP_JOY_Init (void) Configures joystick available on adafruit 1.8" TFT shield managed through ADC to detect motion.
JOYState_TypeDef	BSP_JOY_GetState (void) Returns the Joystick key pressed.
void	BSP_JOY_DeInit (void) DeInit joystick GPIOs.

Function Documentation

uint32_t BSP_GetVersion (void)

This method returns the STM32F2xx NUCLEO BSP Driver revision.

Return values:

version,: 0xXYZR (8bits for each decimal, R for RC)

Definition at line **179** of file [stm32f2xx_nucleo_144.c](#).

References [__STM32F2xx_NUCLEO_BSP_VERSION](#).

void BSP_JOY_DeInit (void)

DeInit joystick GPIOs.

Note:

JOY DeInit does not disable the Mfx, just set the Mfx pins in Off mode

Definition at line **794** of file [stm32f2xx_nucleo_144.c](#).

References [ADCx_DeInit\(\)](#).

JOYState_TypeDef BSP_JOY_GetState (void)

Returns the Joystick key pressed.

Note:

To know which Joystick key is pressed we need to detect the voltage level on each key output

- None : 3.3 V / 4095
- SEL : 1.055 V / 1308

- DOWN : 0.71 V / 88
- LEFT : 3.0 V / 3720
- RIGHT : 0.595 V / 737
- UP : 1.65 V / 2046

Return values:

JOYState_TypeDef,: Code of the Joystick key pressed.

Definition at line **811** of file **stm32f2xx_nucleo_144.c**.

References **hnucleo_Adc**, **JOY_DOWN**, **JOY_LEFT**, **JOY_NONE**, **JOY_RIGHT**, **JOY_SEL**, and **JOY_UP**.

uint8_t BSP_JOY_Init (void)

Configures joystick available on adafruit 1.8" TFT shield managed through ADC to detect motion.

Return values:

Joystickstatus (0=> success, 1=> fail)

Definition at line **774** of file **stm32f2xx_nucleo_144.c**.

References **ADCx_Init()**, **hnucleo_Adc**, **NUCLEO_ADCx_CHANNEL**, and **sConfig**.

void BSP_LED_DeInit (Led_TypeDef Led)

DeInit LEDs.

Parameters:

Led,: LED to be de-init. This parameter can be one of the following values:

- LED1
- LED2

- LED3

Note:

Led DeInit does not disable the GPIO clock nor disable the Mfx

Definition at line **218** of file **stm32f2xx_nucleo_144.c**.

References **GPIO_PIN**, and **GPIO_PORT**.

void BSP_LED_Init (Led_TypeDef Led)

Configures LED GPIO.

Parameters:

Led,: Specifies the Led to be configured. This parameter can be one of following parameters:

- LED1
- LED2
- LED3

Definition at line **192** of file **stm32f2xx_nucleo_144.c**.

References **GPIO_PIN**, **GPIO_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:

- LED1
- LED2
- LED3

Definition at line **248** of file [stm32f2xx_nucleo_144.c](#).

References [GPIO_PIN](#), and [GPIO_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:

- LED2

Definition at line **235** of file [stm32f2xx_nucleo_144.c](#).

References [GPIO_PIN](#), and [GPIO_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:

- LED1
- LED2
- LED3

Definition at line **261** of file [stm32f2xx_nucleo_144.c](#).

References [GPIO_PIN](#), and [GPIO_PORT](#).

void BSP_PB_DeInit (Button_TypeDef Button)

Push Button Delnit.

Parameters:

Button,: Button to be configured This parameter should be: BUTTON_USER

Note:

PB Delnit does not disable the GPIO clock

Definition at line **313** of file [stm32f2xx_nucleo_144.c](#).

References [BUTTON_IRQn](#), [BUTTON_PIN](#), and [BUTTON_PORT](#).

uint32_t BSP_PB_GetState (Button_TypeDef Button)

Returns the selected Button state.

Parameters:

Button,: Specifies the Button to be checked. This parameter should be: BUTTON_USER

Return values:

The Button GPIO pin value.

Definition at line **328** of file [stm32f2xx_nucleo_144.c](#).

References [BUTTON_PIN](#), and [BUTTON_PORT](#).

**void BSP_PB_Init (Button_TypeDef Button,
 ButtonMode_TypeDef ButtonMode
)**

Configures Button GPIO and EXTI Line.

Parameters:

- Button,:** Specifies the Button to be configured. This parameter should be: `BUTTON_USER`
- ButtonMode,:** Specifies Button mode. This parameter can be one of following parameters:
- `BUTTON_MODE_GPIO`: Button will be used as simple IO
 - `BUTTON_MODE_EXTI`: Button will be connected to EXTI line with interrupt generation capability

Definition at line **276** of file `stm32f2xx_nucleo_144.c`.

References `BUTTON_IRQn`, `BUTTON_MODE_EXTI`, `BUTTON_MODE_GPIO`, `BUTTON_PIN`, `BUTTON_PORT`, and `BUTTONx_GPIO_CLK_ENABLE`.

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Variables](#)

STM32F2XX NUCLEO 144 LOW LEVEL Private Variables

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Variables

GPIO_TypeDef *	GPIO_PORT [LEDn] = {LED1_GPIO_PORT, LED2_GPIO_PORT, LED3_GPIO_PORT}
const uint16_t	GPIO_PIN [LEDn] = {LED1_PIN, LED2_PIN, LED3_PIN}
GPIO_TypeDef *	BUTTON_PORT [BUTTONn] = {USER_BUTTON_GPIO_PORT}
const uint16_t	BUTTON_PIN [BUTTONn] = {USER_BUTTON_PIN}
const uint8_t	BUTTON_IRQn [BUTTONn] = {USER_BUTTON_EXTI_IRQn}
uint32_t	SpixTimeout = NUCLEO_SPIx_TIMEOUT_MAX BUS variables.
static SPI_HandleTypeDef	hnucleo_Spi
static ADC_HandleTypeDef	hnucleo_Adc
static ADC_ChannelConfTypeDef	sConfig

Variable Documentation

const uint8_t BUTTON_IRQn[BUTTONn] = {USER_BUTTON_EXTI_I

Definition at line **110** of file **stm32f2xx_nucleo_144.c**.

Referenced by **BSP_PB_DeInit()**, and **BSP_PB_Init()**.

const uint16_t BUTTON_PIN[BUTTONn] = {USER_BUTTON_PIN}

Definition at line **109** of file **stm32f2xx_nucleo_144.c**.

Referenced by **BSP_PB_DeInit()**, **BSP_PB_GetState()**, and **BSP_PB_Init()**.

GPIO_TypeDef* BUTTON_PORT[BUTTONn] = {USER_BUTTON_GPI

Definition at line **108** of file **stm32f2xx_nucleo_144.c**.

Referenced by **BSP_PB_DeInit()**, **BSP_PB_GetState()**, and **BSP_PB_Init()**.

const uint16_t GPIO_PIN[LEDn] = {LED1_PIN, LED2_PIN, LED3_PIN

Definition at line **106** of file **stm32f2xx_nucleo_144.c**.

Referenced by **BSP_LED_DeInit()**, **BSP_LED_Init()**, **BSP_LED_Off()**, **BSP_LED_On()**, and **BSP_LED_Toggle()**.

GPIO_TypeDef* GPIO_PORT[LEDn] = {LED1_GPIO_PORT, LED2_G

Definition at line **104** of file **stm32f2xx_nucleo_144.c**.

Referenced by **BSP_LED_DeInit()**, **BSP_LED_Init()**, **BSP_LED_Off()**, **BSP_LED_On()**, and **BSP_LED_Toggle()**.

ADC_HandleTypeDef hnucleo_Adc [static]

Definition at line **123** of file **stm32f2xx_nucleo_144.c**.

Referenced by **ADCx_DeInit()**, **ADCx_Init()**, **BSP_JOY_GetState()**, and **BSP_JOY_Init()**.

SPI_HandleTypeDef hnucleo_Spi [static]

Definition at line **119** of file **stm32f2xx_nucleo_144.c**.

Referenced by **LCD_IO_WriteMultipleData()**, **SPIx_Error()**, **SPIx_Init()**, **SPIx_Write()**, and **SPIx_WriteReadData()**.

ADC_ChannelConfTypeDef sConfig [static]

Definition at line **125** of file **stm32f2xx_nucleo_144.c**.

Referenced by **BSP_JOY_Init()**.

uint32_t SpixTimeout = **NUCLEO_SPIx_TIMEOUT_MAX**

BUS variables.

Definition at line **118** of file **stm32f2xx_nucleo_144.c**.

Referenced by **SPIx_Write()**, and **SPIx_WriteReadData()**.

BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Enumerations](#)

STM32F2XX NUCLEO 144 LOW LEVEL Exported Types

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Enumerations

```
enum Led_TypeDef {  
    LED1 = 0, LED_GREEN = LED1, LED2 = 1, LED_BLUE =  
    LED2,  
    LED3 = 2, LED_RED = LED3  
}
```

```
enum Button_TypeDef { BUTTON_USER = 0, BUTTON_KEY =  
    BUTTON_USER }
```

```
enum ButtonMode_TypeDef { BUTTON_MODE_GPIO = 0,  
    BUTTON_MODE_EXTI = 1 }
```

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

Enumeration Type Documentation

enum [Button_TypeDef](#)

Enumerator:

BUTTON_USER

BUTTON_KEY

Definition at line **84** of file [stm32f2xx_nucleo_144.h](#).

enum [ButtonMode_TypeDef](#)

Enumerator:

BUTTON_MODE_GPIO

BUTTON_MODE_EXTI

Definition at line **91** of file [stm32f2xx_nucleo_144.h](#).

enum [JOYState_TypeDef](#)

Enumerator:

JOY_NONE

JOY_SEL

JOY_DOWN

JOY_LEFT

JOY_RIGHT

JOY_UP

Definition at line **97** of file [stm32f2xx_nucleo_144.h](#).

enum [Led_TypeDef](#)

Enumerator:*LED1**LED_GREEN**LED2**LED_BLUE**LED3**LED_RED*

Definition at line **74** of file **stm32f2xx_nucleo_144.h**.

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Defines](#)

STM32F2XX NUCLEO 144 LOW LEVEL BUTTON

[STM32F2XX NUCLEO 144 LOW LEVEL Exported Constants](#)

Defines

#define	BUTTONn	1
#define	USER_BUTTON_PIN	GPIO_PIN_13 Key push-button.
#define	USER_BUTTON_GPIO_PORT	GPIOC
#define	USER_BUTTON_GPIO_CLK_ENABLE()	__HAL_RCC_GPIOC_CLK_ENABLE()
#define	USER_BUTTON_GPIO_CLK_DISABLE()	__HAL_RCC_GPIOC_CLK_DISABLE()
#define	USER_BUTTON_EXTI_LINE	GPIO_PIN_13
#define	USER_BUTTON_EXTI_IRQn	EXTI15_10_IRQn
#define	BUTTONx_GPIO_CLK_ENABLE(__INDEX__)	USER_BUTTON_GPIO_CLK_ENABLE()
#define	BUTTONx_GPIO_CLK_DISABLE(__INDEX__)	USER_BUTTON_GPIO_CLK_DISABLE()
#define	KEY_BUTTON_PIN	USER_BUTTON_PIN
#define	KEY_BUTTON_GPIO_PORT	USER_BUTTON_GPIO_PORT
#define	KEY_BUTTON_GPIO_CLK_ENABLE()	USER_BUTTON_GPIO_CLK_ENABLE()
#define	KEY_BUTTON_GPIO_CLK_DISABLE()	USER_BUTTON_GPIO_CLK_DISABLE()
#define	KEY_BUTTON_EXTI_LINE	USER_BUTTON_EXTI_LINE
#define	KEY_BUTTON_EXTI_IRQn	USER_BUTTON_EXTI_IRQn
#define	OTG_FS1_OVER_CURRENT_PIN	GPIO_PIN_7 OTG_FS1 OVER_CURRENT and POWER_SWITCH Pins definition
#define	OTG_FS1_OVER_CURRENT_PORT	GPIOG
#define	OTG_FS1_OVER_CURRENT_PORT_CLK_ENABLE()	__HAL_RCC_GPIOG_CLK_ENABLE()
#define	OTG_FS1_POWER_SWITCH_PIN	GPIO_PIN_6
#define	OTG_FS1_POWER_SWITCH_PORT	GPIOG
#define	OTG_FS1_POWER_SWITCH_PORT_CLK_ENABLE()	__HAL_RCC_GPIOG_CLK_ENABLE()

Define Documentation

#define BUTTONn 1

Definition at line **153** of file **stm32f2xx_nucleo_144.h**.

#define BUTTONx_GPIO_CLK_DISABLE (__INDEX__) USER_BUTTONx_GPIO_CLK_DISABLE

Definition at line **166** of file **stm32f2xx_nucleo_144.h**.

#define BUTTONx_GPIO_CLK_ENABLE (__INDEX__) USER_BUTTONx_GPIO_CLK_ENABLE

Definition at line **165** of file **stm32f2xx_nucleo_144.h**.

Referenced by **BSP_PB_Init()**.

#define KEY_BUTTON_EXTI_IRQn USER_BUTTON_EXTI_IRQn

Definition at line **174** of file **stm32f2xx_nucleo_144.h**.

#define KEY_BUTTON_EXTI_LINE USER_BUTTON_EXTI_LINE

Definition at line **173** of file **stm32f2xx_nucleo_144.h**.

#define KEY_BUTTON_GPIO_CLK_DISABLE () USER_BUTTON_GPIO_CLK_DISABLE

Definition at line **172** of file **stm32f2xx_nucleo_144.h**.

#define KEY_BUTTON_GPIO_CLK_ENABLE () USER_BUTTON_GPIO_CLK_ENABLE

Definition at line **171** of file [stm32f2xx_nucleo_144.h](#).

#define KEY_BUTTON_GPIO_PORT USER_BUTTON_GPIO_PORT

Definition at line **170** of file [stm32f2xx_nucleo_144.h](#).

#define KEY_BUTTON_PIN USER_BUTTON_PIN

Definition at line **169** of file [stm32f2xx_nucleo_144.h](#).

#define OTG_FS1_OVER_CURRENT_PIN GPIO_PIN_7

OTG_FS1 OVER_CURRENT and POWER_SWITCH Pins definition.

Definition at line **182** of file [stm32f2xx_nucleo_144.h](#).

#define OTG_FS1_OVER_CURRENT_PORT GPIOG

Definition at line **183** of file [stm32f2xx_nucleo_144.h](#).

#define OTG_FS1_OVER_CURRENT_PORT_CLK_ENABLE () __F

Definition at line **184** of file [stm32f2xx_nucleo_144.h](#).

#define OTG_FS1_POWER_SWITCH_PIN GPIO_PIN_6

Definition at line **186** of file [stm32f2xx_nucleo_144.h](#).

#define OTG_FS1_POWER_SWITCH_PORT GPIOG

Definition at line **187** of file **stm32f2xx_nucleo_144.h**.

#define OTG_FS1_POWER_SWITCH_PORT_CLK_ENABLE () __r

Definition at line **188** of file **stm32f2xx_nucleo_144.h**.

#define USER_BUTTON_EXTI_IRQn EXTI15_10_IRQn

Definition at line **163** of file **stm32f2xx_nucleo_144.h**.

#define USER_BUTTON_EXTI_LINE GPIO_PIN_13

Definition at line **162** of file **stm32f2xx_nucleo_144.h**.

#define USER_BUTTON_GPIO_CLK_DISABLE () __HAL_RCC_GI

Definition at line **161** of file **stm32f2xx_nucleo_144.h**.

#define USER_BUTTON_GPIO_CLK_ENABLE () __HAL_RCC_GF

Definition at line **160** of file **stm32f2xx_nucleo_144.h**.

#define USER_BUTTON_GPIO_PORT GPIOC

Definition at line **159** of file **stm32f2xx_nucleo_144.h**.

#define USER_BUTTON_PIN GPIO_PIN_13

Key push-button.

Definition at line **158** of file **stm32f2xx_nucleo_144.h**.

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Defines](#)

STM32F2XX NUCLEO 144 LOW LEVEL BUS

[STM32F2XX NUCLEO 144 LOW LEVEL Exported Constants](#)

Defines

#define	NUCLEO_SPIx	SPI1
#define	NUCLEO_SPIx_CLK_ENABLE()	__HAL_RCC_SPI1_CLK_E
#define	NUCLEO_SPIx_SCK_AF	GPIO_AF5_SPI1
#define	NUCLEO_SPIx_SCK_GPIO_PORT	GPIOA
#define	NUCLEO_SPIx_SCK_PIN	GPIO_PIN_5
#define	NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE()	__HAL_RCC_
#define	NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE()	__HAL_RCC_
#define	NUCLEO_SPIx_MISO_MOSI_AF	GPIO_AF5_SPI1
#define	NUCLEO_SPIx_MISO_MOSI_GPIO_PORT	GPIOA
#define	NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE()	__HAL
#define	NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISABLE()	__HA
#define	NUCLEO_SPIx_MISO_PIN	GPIO_PIN_6
#define	NUCLEO_SPIx_MOSI_PIN	GPIO_PIN_7
#define	NUCLEO_SPIx_TIMEOUT_MAX	1000
#define	NUCLEO_SPIx_CS_GPIO_PORT	GPIOD
#define	NUCLEO_SPIx_CS_PIN	GPIO_PIN_14
#define	NUCLEO_SPIx_CS_GPIO_CLK_ENABLE()	__HAL_RCC_G
#define	NUCLEO_SPIx_CS_GPIO_CLK_DISABLE()	__HAL_RCC_C
#define	SPIx_CS_LOW()	HAL_GPIO_WritePin(NUCLEO_SPIx_CS , NUCLEO_SPIx_CS_PIN , GPIO_PIN_RESET)
#define	SPIx_CS_HIGH()	HAL_GPIO_WritePin(NUCLEO_SPIx_CS , NUCLEO_SPIx_CS_PIN , GPIO_PIN_SET)
#define	SD_CS_LOW()	HAL_GPIO_WritePin(SD_CS_GPIO_PORT , , SD Control Lines management.
#define	SD_CS_HIGH()	HAL_GPIO_WritePin(SD_CS_GPIO_PORT , , SD Control Lines management.
#define	LCD_CS_LOW()	HAL_GPIO_WritePin(LCD_CS_GPIO_POR , GPIO_PIN_RESET) LCD Control Lines management.
#define	LCD_CS_HIGH()	HAL_GPIO_WritePin(LCD_CS_GPIO_POF , GPIO_PIN_SET)
#define	LCD_DC_LOW()	HAL_GPIO_WritePin(LCD_DC_GPIO_POR , , LCD Control Lines management.

	GPIO_PIN_RESET)
#define	LCD_DC_HIGH() HAL_GPIO_WritePin(LCD_DC_GPIO_PORT , GPIO_PIN_SET)
#define	SD_CS_PIN GPIO_PIN_14 SD Control Interface pins (shield D4)
#define	SD_CS_GPIO_PORT GPIOF
#define	SD_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOF_CLK_
#define	SD_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPIOF_CLK_
#define	LCD_CS_PIN GPIO_PIN_14 LCD Control Interface pins (shield D10)
#define	LCD_CS_GPIO_PORT GPIOD
#define	LCD_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK
#define	LCD_CS_GPIO_CLK_DISABLE() __HAL_RCC_GPIOD_CLI
#define	LCD_DC_PIN GPIO_PIN_12 LCD Data/Command Interface pins (shield D8)
#define	LCD_DC_GPIO_PORT GPIOF
#define	LCD_DC_GPIO_CLK_ENABLE() __HAL_RCC_GPIOF_CLK
#define	LCD_DC_GPIO_CLK_DISABLE() __HAL_RCC_GPIOF_CLI
#define	NUCLEO_ADCx ADC3 ADCx Interface pins used to detect motion of Joystick available
#define	NUCLEO_ADCx_CLK_ENABLE() __HAL_RCC_ADC3_CLK
#define	NUCLEO_ADCx_CLK_DISABLE() __HAL_RCC_ADC3_CLI
#define	NUCLEO_ADCx_CHANNEL ADC_CHANNEL_9
#define	NUCLEO_ADCx_GPIO_PORT GPIOF
#define	NUCLEO_ADCx_GPIO_PIN GPIO_PIN_3
#define	NUCLEO_ADCx_GPIO_CLK_ENABLE() __HAL_RCC_GPIOF_CLK
#define	NUCLEO_ADCx_GPIO_CLK_DISABLE() __HAL_RCC_GPIOF_CLI

Define Documentation

#define LCD_CS_GPIO_CLK_DISABLE () __HAL_RCC_GPIOID_C

Definition at line 258 of file [stm32f2xx_nucleo_144.h](#).

#define LCD_CS_GPIO_CLK_ENABLE () __HAL_RCC_GPIOID_CI

Definition at line 257 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [LCD_IO_Init\(\)](#), and [SD_IO_Init\(\)](#).

#define LCD_CS_GPIO_PORT GPIOID

Definition at line 256 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [LCD_IO_Init\(\)](#), and [SD_IO_Init\(\)](#).

#define LCD_CS_HIGH () HAL_GPIO_WritePin(LCD_CS_GPIO_P

Definition at line 240 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [LCD_IO_Init\(\)](#), [LCD_IO_WriteData\(\)](#), [LCD_IO_WriteMultipleData\(\)](#), [LCD_IO_WriteReg\(\)](#), and [SD_IO_Init\(\)](#).

#define LCD_CS_LOW () HAL_GPIO_WritePin(LCD_CS_GPIO_PC

LCD Control Lines management.

Definition at line 239 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [LCD_IO_WriteData\(\)](#), [LCD_IO_WriteMultipleData\(\)](#), and [LCD_IO_WriteReg\(\)](#).

#define LCD_CS_PIN GPIO_PIN_14

LCD Control Interface pins (shield D10)

Definition at line [255](#) of file [stm32f2xx_nucleo_144.h](#).

Referenced by [LCD_IO_Init\(\)](#), and [SD_IO_Init\(\)](#).

#define LCD_DC_GPIO_CLK_DISABLE () __HAL_RCC_GPIOF_C

Definition at line [266](#) of file [stm32f2xx_nucleo_144.h](#).

#define LCD_DC_GPIO_CLK_ENABLE () __HAL_RCC_GPIOF_CI

Definition at line [265](#) of file [stm32f2xx_nucleo_144.h](#).

Referenced by [LCD_IO_Init\(\)](#).

#define LCD_DC_GPIO_PORT GPIOF

Definition at line [264](#) of file [stm32f2xx_nucleo_144.h](#).

Referenced by [LCD_IO_Init\(\)](#).

#define LCD_DC_HIGH () HAL_GPIO_WritePin(LCD_DC_GPIO_P

Definition at line [242](#) of file [stm32f2xx_nucleo_144.h](#).

Referenced by [LCD_IO_WriteData\(\)](#), and

LCD_IO_WriteMultipleData().

#define LCD_DC_LOW () HAL_GPIO_WritePin(LCD_DC_GPIO_P

Definition at line **241** of file **stm32f2xx_nucleo_144.h**.

Referenced by **LCD_IO_WriteReg()**.

#define LCD_DC_PIN GPIO_PIN_12

LCD Data/Command Interface pins (shield D8)

Definition at line **263** of file **stm32f2xx_nucleo_144.h**.

Referenced by **LCD_IO_Init()**.

#define NUCLEO_ADCx ADC3

ADCx Interface pins used to detect motion of Joystick available on Adafruit 1.8" TFT shield.

Definition at line **278** of file **stm32f2xx_nucleo_144.h**.

Referenced by **ADCx_DeInit()**, and **ADCx_Init()**.

#define NUCLEO_ADCx_CHANNEL ADC_CHANNEL_9

Definition at line **282** of file **stm32f2xx_nucleo_144.h**.

Referenced by **BSP_JOY_Init()**.

#define NUCLEO_ADCx_CLK_DISABLE () __HAL_RCC_ADC3_C

Definition at line **280** of file **stm32f2xx_nucleo_144.h**.

Referenced by **ADCx_MspDeInit()**.

#define NUCLEO_ADCx_CLK_ENABLE () __HAL_RCC_ADC3_C

Definition at line **279** of file **stm32f2xx_nucleo_144.h**.

Referenced by **ADCx_MspInit()**.

#define NUCLEO_ADCx_GPIO_CLK_DISABLE () __HAL_RCC_GI

Definition at line **287** of file **stm32f2xx_nucleo_144.h**.

#define NUCLEO_ADCx_GPIO_CLK_ENABLE () __HAL_RCC_GF

Definition at line **286** of file **stm32f2xx_nucleo_144.h**.

Referenced by **ADCx_MspInit()**.

#define NUCLEO_ADCx_GPIO_PIN GPIO_PIN_3

Definition at line **285** of file **stm32f2xx_nucleo_144.h**.

Referenced by **ADCx_MspDeInit()**, and **ADCx_MspInit()**.

#define NUCLEO_ADCx_GPIO_PORT GPIOF

Definition at line **284** of file **stm32f2xx_nucleo_144.h**.

Referenced by **ADCx_MspDeInit()**, and **ADCx_MspInit()**.

#define NUCLEO_SPIx SPI1

Definition at line 200 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [SPIx_Init\(\)](#).

#define NUCLEO_SPIx_CLK_ENABLE () __HAL_RCC_SPI1_CLK

Definition at line 201 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [SPIx_Msplnit\(\)](#).

#define NUCLEO_SPIx_CS_GPIO_CLK_DISABLE () __HAL_RCC

Definition at line 225 of file [stm32f2xx_nucleo_144.h](#).

#define NUCLEO_SPIx_CS_GPIO_CLK_ENABLE () __HAL_RCC

Definition at line 224 of file [stm32f2xx_nucleo_144.h](#).

#define NUCLEO_SPIx_CS_GPIO_PORT GPIOD

Definition at line 222 of file [stm32f2xx_nucleo_144.h](#).

#define NUCLEO_SPIx_CS_PIN GPIO_PIN_14

Definition at line 223 of file [stm32f2xx_nucleo_144.h](#).

#define NUCLEO_SPIx_MISO_MOSI_AF GPIO_AF5_SPI1

Definition at line **209** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISABLE () __H

Definition at line **212** of file **stm32f2xx_nucleo_144.h**.

#define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE () __H

Definition at line **211** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_MISO_MOSI_GPIO_PORT GPIOA

Definition at line **210** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_MISO_PIN GPIO_PIN_6

Definition at line **213** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_MOSI_PIN GPIO_PIN_7

Definition at line **214** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_SCK_AF GPIO_AF5_SPI1

Definition at line **203** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE () __HAL_RCC

Definition at line **207** of file **stm32f2xx_nucleo_144.h**.

#define NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE () __HAL_RCC

Definition at line **206** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_SCK_GPIO_PORT GPIOA

Definition at line **204** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_SCK_PIN GPIO_PIN_5

Definition at line **205** of file **stm32f2xx_nucleo_144.h**.

Referenced by **SPIx_Msplnit()**.

#define NUCLEO_SPIx_TIMEOUT_MAX 1000

Definition at line **220** of file **stm32f2xx_nucleo_144.h**.

#define SD_CS_GPIO_CLK_DISABLE () __HAL_RCC_GPIOF_CLK

Definition at line 250 of file [stm32f2xx_nucleo_144.h](#).

#define SD_CS_GPIO_CLK_ENABLE () __HAL_RCC_GPIOF_CLK

Definition at line 249 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [SD_IO_Init\(\)](#).

#define SD_CS_GPIO_PORT GPIOF

Definition at line 248 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [SD_IO_Init\(\)](#).

#define SD_CS_HIGH () HAL_GPIO_WritePin(SD_CS_GPIO_PORT,

Definition at line 234 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [SD_IO_CSState\(\)](#), and [SD_IO_Init\(\)](#).

#define SD_CS_LOW () HAL_GPIO_WritePin(SD_CS_GPIO_PORT,

SD Control Lines management.

Definition at line 233 of file [stm32f2xx_nucleo_144.h](#).

Referenced by [SD_IO_CSState\(\)](#).

#define SD_CS_PIN GPIO_PIN_14

SD Control Interface pins (shield D4)

Definition at line **247** of file [stm32f2xx_nucleo_144.h](#).

Referenced by [SD_IO_Init\(\)](#).

```
#define SPIx__CS_HIGH ( ) HAL_GPIO_WritePin(NUCLEO_SPIx_C
```

Definition at line **228** of file [stm32f2xx_nucleo_144.h](#).

```
#define SPIx__CS_LOW ( ) HAL_GPIO_WritePin(NUCLEO_SPIx_C
```

Definition at line **227** of file [stm32f2xx_nucleo_144.h](#).

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Defines](#)

STM32F2XX NUCLEO 144 LOW LEVEL LED

[STM32F2XX NUCLEO 144 LOW LEVEL Exported Constants](#)

Define for STM32F2XX_NUCLEO_144 board. [More...](#)

Defines

#define	LEDn	3
#define	LED1_PIN	GPIO_PIN_0
#define	LED1_GPIO_PORT	GPIOB
#define	LED1_GPIO_CLK_ENABLE()	__HAL_RCC_GPIOB_CLK_ENABLE()
#define	LED1_GPIO_CLK_DISABLE()	__HAL_RCC_GPIOB_CLK_DISABLE()
#define	LED2_PIN	GPIO_PIN_7
#define	LED2_GPIO_PORT	GPIOB
#define	LED2_GPIO_CLK_ENABLE()	__HAL_RCC_GPIOB_CLK_ENABLE()
#define	LED2_GPIO_CLK_DISABLE()	__HAL_RCC_GPIOB_CLK_DISABLE()
#define	LED3_PIN	GPIO_PIN_14
#define	LED3_GPIO_PORT	GPIOB
#define	LED3_GPIO_CLK_ENABLE()	__HAL_RCC_GPIOB_CLK_ENABLE()
#define	LED3_GPIO_CLK_DISABLE()	__HAL_RCC_GPIOB_CLK_DISABLE()
#define	LEDx_GPIO_CLK_ENABLE(__INDEX__)	
#define	LEDx_GPIO_CLK_DISABLE(__INDEX__)	

Detailed Description

Define for STM32F2XX_NUCLEO_144 board.

Define Documentation

#define LED1_GPIO_CLK_DISABLE () __HAL_RCC_GPIOB_CLK

Definition at line 130 of file [stm32f2xx_nucleo_144.h](#).

#define LED1_GPIO_CLK_ENABLE () __HAL_RCC_GPIOB_CLK

Definition at line 129 of file [stm32f2xx_nucleo_144.h](#).

#define LED1_GPIO_PORT GPIOB

Definition at line 128 of file [stm32f2xx_nucleo_144.h](#).

#define LED1_PIN GPIO_PIN_0

Definition at line 127 of file [stm32f2xx_nucleo_144.h](#).

#define LED2_GPIO_CLK_DISABLE () __HAL_RCC_GPIOB_CLK

Definition at line 135 of file [stm32f2xx_nucleo_144.h](#).

#define LED2_GPIO_CLK_ENABLE () __HAL_RCC_GPIOB_CLK

Definition at line 134 of file [stm32f2xx_nucleo_144.h](#).

#define LED2_GPIO_PORT GPIOB

Definition at line 133 of file [stm32f2xx_nucleo_144.h](#).

```
#define LED2_PIN GPIO_PIN_7
```

Definition at line **132** of file **stm32f2xx_nucleo_144.h**.

```
#define LED3_GPIO_CLK_DISABLE ( ) __HAL_RCC_GPIOB_CLK
```

Definition at line **140** of file **stm32f2xx_nucleo_144.h**.

```
#define LED3_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIOB_CLK
```

Definition at line **139** of file **stm32f2xx_nucleo_144.h**.

```
#define LED3_GPIO_PORT GPIOB
```

Definition at line **138** of file **stm32f2xx_nucleo_144.h**.

```
#define LED3_PIN GPIO_PIN_14
```

Definition at line **137** of file **stm32f2xx_nucleo_144.h**.

```
#define LEDn 3
```

Definition at line **125** of file **stm32f2xx_nucleo_144.h**.

```
#define LEDx_GPIO_CLK_DISABLE ( __INDEX__ )
```

Value:

```
do { if((__INDEX__) == 0) {__HAL_RCC_GPIOB_CLK_DI  
SABLE();} else\
```



```
        {__HAL_RCC_GPIOB_CLK_DISABLE()  
;    } } while(0)
```

Definition at line **144** of file **stm32f2xx_nucleo_144.h**.

#define LEDx_GPIO_CLK_ENABLE (__INDEX__)

Value:

```
do { if((__INDEX__) == 0) {__HAL_RCC_GPIOB_CLK_EN  
ABLE();} else\  
  
        {__HAL_RCC_GPIOB_CLK_ENABLE();  
    } } while(0)
```

Definition at line **142** of file **stm32f2xx_nucleo_144.h**.

Referenced by **BSP_LED_Init()**.

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Drivers](#)

Drivers Directory Reference

Directories

directory **BSP**

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

[Main Page](#)[Modules](#)[Files](#)[Directories](#)[Drivers](#)[BSP](#)

BSP Directory Reference

Directories

directory **STM32F2xx_Nucleo_144**

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

Main Page	Modules	Files	Directories	
Drivers	BSP	STM32F2xx_Nucleo_144		

STM32F2xx_Nucleo_144 Directory Reference

Files

file [stm32f2xx_nucleo_144.c](#) [code]

This file provides set of firmware functions to manage:

file [stm32f2xx_nucleo_144.h](#) [code]

This file contains definitions for:

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by [doxygen](#) 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

Main Page	Modules	Files	Directories
File List	Globals		
Drivers	BSP	STM32F2xx_Nucleo_144	

stm32f2xx_nucleo_144.h

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

```
00001  /**
00002      ****
00003      ****
00004      * @file      stm32f2xx_nucleo_144.h
00005      * @author    MCD Application Team
00006      * @version    V1.0.0
00007      * @date      20-November-2015
00008      * @brief     This file contains definitions
00009      for:
00010      *           - LEDs and push-button available on STM32F2XX-Nucleo-144 Kit
00011      *           from STMicroelectronics
00012      *           - LCD, joystick and microSD available on Adafruit 1.8" TFT LCD
00013      *           shield (reference ID 802)
00014      ****
00015      ****
00016      * @attention
00017      *
00018      * <h2><center>©; COPYRIGHT(c) 2017 STMicroelectronics</center></h2>
00019      *
00020      * Redistribution and use in source and binary forms, with or without modification,
```


00018 * are permitted provided that the following conditions are met:

00019 * 1. Redistributions of source code must retain the above copyright notice,

00020 * this list of conditions and the following disclaimer.

00021 * 2. Redistributions in binary form must reproduce the above copyright notice,

00022 * this list of conditions and the following disclaimer in the documentation

00023 * and/or other materials provided with the distribution.

00024 * 3. Neither the name of STMicroelectronics nor the names of its contributors

00025 * may be used to endorse or promote products derived from this software

00026 * without specific prior written permission.

00027 *

00028 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

00029 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

00030 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

00031 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

00032 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

00033 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

00034 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

00035 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

00036 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

```

00037      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
        POSSIBILITY OF SUCH DAMAGE.
00038      *
00039      ****
*****
00040      */
00041
00042 /* Define to prevent recursive inclusion ---
----- */
00043 #ifndef __STM32F2XX_NUCLEO_144_H
00044 #define __STM32F2XX_NUCLEO_144_H
00045
00046 #ifdef __cplusplus
00047     extern "C" {
00048 #endif
00049
00050 /* Includes -----
----- */
00051 #include "stm32f2xx_hal.h"
00052
00053 /* To be defined only if the board is provided with the related shield */
00054 /* https://www.adafruit.com/products/802 */
00055 #ifndef ADAFRUIT_TFT_JOY_SD_ID802
00056 #define ADAFRUIT_TFT_JOY_SD_ID802
00057 #endif
00058
00059 /** @addtogroup BSP
    * @{
00061     */
00062
00063 /** @addtogroup STM32F2XX_NUCLEO_144
    * @{
00065     */
00066
00067 /** @addtogroup STM32F2XX_NUCLEO_144_LOW_LEVEL
EL

```

```

00068      * @{
00069      */
00070
00071 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
_Exported_Types STM32F2XX NUCLEO 144 LOW LEVEL Exp
orted Types
00072      * @{
00073      */
00074 typedef enum
00075 {
00076     LED1 = 0,
00077     LED_GREEN = LED1,
00078     LED2 = 1,
00079     LED_BLUE = LED2,
00080     LED3 = 2,
00081     LED_RED = LED3
00082 }Led_TypeDef;
00083
00084 typedef enum
00085 {
00086     BUTTON_USER = 0,
00087     /* Alias */
00088     BUTTON_KEY = BUTTON_USER
00089 }Button_TypeDef;
00090
00091 typedef enum
00092 {
00093     BUTTON_MODE_GPIO = 0,
00094     BUTTON_MODE_EXTI = 1
00095 }ButtonMode_TypeDef;
00096
00097 typedef enum
00098 {
00099     JOY_NONE = 0,
00100     JOY_SEL = 1,
00101     JOY_DOWN = 2,
00102     JOY_LEFT = 3,

```

```

00103     JOY_RIGHT = 4,
00104     JOY_UP     = 5
00105 }JOYState_TypeDef;
00106
00107 /**
00108  * @}
00109  */
00110
00111 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
    _Exported_Constants STM32F2XX NUCLEO 144 LOW LEVEL
    Exported Constants
00112  * @{
00113  */
00114
00115 /**
00116  * @brief Define for STM32F2XX_NUCLEO_144 board
00117  */
00118 #if !defined (USE_STM32F2XX_NUCLEO_144)
00119 #define USE_STM32F2XX_NUCLEO_144
00120 #endif
00121
00122 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
    _LED STM32F2XX NUCLEO 144 LOW LEVEL LED
00123  * @{
00124  */
00125 #define LEDn
    3
00126
00127 #define LED1_PIN
    GPIO_PIN_0
00128 #define LED1_GPIO_PORT
    GPIOB
00129 #define LED1_GPIO_CLK_ENABLE()
    __HAL_RCC_GPIOB_CLK_ENABLE()
00130 #define LED1_GPIO_CLK_DISABLE()
    __HAL_RCC_GPIOB_CLK_DISABLE()

```

```

00131
00132 #define LED2_PIN
        GPIO_PIN_7
00133 #define LED2_GPIO_PORT
        GPIOB
00134 #define LED2_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00135 #define LED2_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOB_CLK_DISABLE()
00136
00137 #define LED3_PIN
        GPIO_PIN_14
00138 #define LED3_GPIO_PORT
        GPIOB
00139 #define LED3_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00140 #define LED3_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOB_CLK_DISABLE()
00141
00142 #define LEDx_GPIO_CLK_ENABLE(__INDEX__) do
    { if((__INDEX__) == 0) {__HAL_RCC_GPIOB_CLK_ENABL
00143
        {__HAL_RCC_GPIOB_CLK_ENABL
E();    }} while(0)
00144 #define LEDx_GPIO_CLK_DISABLE(__INDEX__) do
    { if((__INDEX__) == 0) {__HAL_RCC_GPIOB_CLK_DISAB
LE();} else\
00145
        {__HAL_RCC_GPIOB_CLK_DISAB
LE();    }} while(0)
00146 /**
00147     * @}
00148     */
00149
00150 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
    _BUTTON STM32F2XX NUCLEO 144 LOW LEVEL BUTTON

```

```
00151     * @{
00152     */
00153 #define BUTTONn
00154     1
00155 /**
00156  * @brief Key push-button
00157  */
00158 #define USER_BUTTON_PIN
00159     GPIO_PIN_13
00160 #define USER_BUTTON_GPIO_PORT
00161     GPIOC
00162 #define USER_BUTTON_GPIO_CLK_ENABLE()
00163     __HAL_RCC_GPIOC_CLK_ENABLE()
00164 #define USER_BUTTON_GPIO_CLK_DISABLE()
00165     __HAL_RCC_GPIOC_CLK_DISABLE()
00166 #define USER_BUTTON_EXTI_LINE
00167     GPIO_PIN_13
00168 #define USER_BUTTON_EXTI_IRQn
00169     EXTI15_10_IRQn
00170
00171 #define BUTTONx_GPIO_CLK_ENABLE(__INDEX__)
00172     USER_BUTTON_GPIO_CLK_ENABLE()
00173 #define BUTTONx_GPIO_CLK_DISABLE(__INDEX__)
00174     USER_BUTTON_GPIO_CLK_DISABLE()
00175
00176 /* Aliases */
00177 #define KEY_BUTTON_PIN
00178     USER_BUTTON_PIN
00179 #define KEY_BUTTON_GPIO_PORT
00180     USER_BUTTON_GPIO_PORT
00181 #define KEY_BUTTON_GPIO_CLK_ENABLE()
00182     USER_BUTTON_GPIO_CLK_ENABLE()
00183 #define KEY_BUTTON_GPIO_CLK_DISABLE()
00184     USER_BUTTON_GPIO_CLK_DISABLE()
00185 #define KEY_BUTTON_EXTI_LINE
00186     USER_BUTTON_EXTI_LINE
```

```

00174 #define KEY_BUTTON_EXTI_IRQn
      USER_BUTTON_EXTI_IRQn
00175
00176
00177 /**
00178  * @brief OTG_FS1 OVER_CURRENT and POWER_SW
ITCH Pins definition
00179  */
00180
00181
00182 #define OTG_FS1_OVER_CURRENT_PIN
      GPIO_PIN_7
00183 #define OTG_FS1_OVER_CURRENT_PORT
      GPIOG
00184 #define OTG_FS1_OVER_CURRENT_PORT_CLK_ENABLE
( )      __HAL_RCC_GPIOG_CLK_ENABLE( )
00185
00186 #define OTG_FS1_POWER_SWITCH_PIN
      GPIO_PIN_6
00187 #define OTG_FS1_POWER_SWITCH_PORT
      GPIOG
00188 #define OTG_FS1_POWER_SWITCH_PORT_CLK_ENABLE
( )      __HAL_RCC_GPIOG_CLK_ENABLE( )
00189
00190 /**
00191  * @}
00192  */
00193
00194 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
_BUS STM32F2XX NUCLEO 144 LOW LEVEL BUS
00195  * @{
00196  */
00197 /*##### SPI_A #####
#####*/
00198 #ifdef HAL_SPI_MODULE_ENABLED
00199
00200 #define NUCLEO_SPIx

```

```

        SPI1
00201 #define NUCLEO_SPIx_CLK_ENABLE()
        __HAL_RCC_SPI1_CLK_ENABLE()
00202
00203 #define NUCLEO_SPIx_SCK_AF
        GPIO_AF5_SPI1
00204 #define NUCLEO_SPIx_SCK_GPIO_PORT
        GPIOA
00205 #define NUCLEO_SPIx_SCK_PIN
        GPIO_PIN_5
00206 #define NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOA_CLK_ENABLE()
00207 #define NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOA_CLK_DISABLE()
00208
00209 #define NUCLEO_SPIx_MISO_MOSI_AF
        GPIO_AF5_SPI1
00210 #define NUCLEO_SPIx_MISO_MOSI_GPIO_PORT
        GPIOA
00211 #define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABL
E()        __HAL_RCC_GPIOA_CLK_ENABLE()
00212 #define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISAB
LE()        __HAL_RCC_GPIOA_CLK_DISABLE()
00213 #define NUCLEO_SPIx_MISO_PIN
        GPIO_PIN_6
00214 #define NUCLEO_SPIx_MOSI_PIN
        GPIO_PIN_7
00215 /* Maximum Timeout values for flags waiting
loops. These timeout are not based
00216     on accurate values, they just guarantee t
hat the application will not remain
00217     stuck if the SPI communication is corrupt
ed.
00218     You may modify these timeout values depen
ding on CPU frequency and application
00219     conditions (interrupts routines ...). */

```



```

00220 #define NUCLEO_SPIx_TIMEOUT_MAX
      1000
00221
00222 #define NUCLEO_SPIx_CS_GPIO_PORT
      GPIOD
00223 #define NUCLEO_SPIx_CS_PIN
      GPIO_PIN_14
00224 #define NUCLEO_SPIx_CS_GPIO_CLK_ENABLE()
      __HAL_RCC_GPIOD_CLK_ENABLE()
00225 #define NUCLEO_SPIx_CS_GPIO_CLK_DISABLE()
      __HAL_RCC_GPIOD_CLK_DISABLE()
00226
00227 #define SPIx__CS_LOW()          HAL_GPIO_WriteP
in(NUCLEO_SPIx_CS_GPIO_PORT, NUCLEO_SPIx_CS_PIN, G
PIO_PIN_RESET)
00228 #define SPIx__CS_HIGH()       HAL_GPIO_WriteP
in(NUCLEO_SPIx_CS_GPIO_PORT, NUCLEO_SPIx_CS_PIN, G
PIO_PIN_SET)
00229
00230 /**
00231  * @brief SD Control Lines management
00232  */
00233 #define SD_CS_LOW()            HAL_GPIO_WritePin(
SD_CS_GPIO_PORT, SD_CS_PIN, GPIO_PIN_RESET)
00234 #define SD_CS_HIGH()          HAL_GPIO_WritePin(
SD_CS_GPIO_PORT, SD_CS_PIN, GPIO_PIN_SET)
00235
00236 /**
00237  * @brief LCD Control Lines management
00238  */
00239 #define LCD_CS_LOW()           HAL_GPIO_WritePin(
LCD_CS_GPIO_PORT, LCD_CS_PIN, GPIO_PIN_RESET)
00240 #define LCD_CS_HIGH()         HAL_GPIO_WritePin(
LCD_CS_GPIO_PORT, LCD_CS_PIN, GPIO_PIN_SET)
00241 #define LCD_DC_LOW()          HAL_GPIO_WritePin(
LCD_DC_GPIO_PORT, LCD_DC_PIN, GPIO_PIN_RESET)
00242 #define LCD_DC_HIGH()         HAL_GPIO_WritePin(

```

```

LCD_DC_GPIO_PORT, LCD_DC_PIN, GPIO_PIN_SET)
00243
00244 /**
00245  * @brief SD Control Interface pins (shield D4)
00246  */
00247 #define SD_CS_PIN
        GPIO_PIN_14
00248 #define SD_CS_GPIO_PORT
        GPIOF
00249 #define SD_CS_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOF_CLK_ENABLE()
00250 #define SD_CS_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOF_CLK_DISABLE()
00251
00252 /**
00253  * @brief LCD Control Interface pins (shield D10)
00254  */
00255 #define LCD_CS_PIN
        GPIO_PIN_14
00256 #define LCD_CS_GPIO_PORT
        GPIOD
00257 #define LCD_CS_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOD_CLK_ENABLE()
00258 #define LCD_CS_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOD_CLK_DISABLE()
00259
00260 /**
00261  * @brief LCD Data/Command Interface pins (shield D8)
00262  */
00263 #define LCD_DC_PIN
        GPIO_PIN_12
00264 #define LCD_DC_GPIO_PORT
        GPIOF
00265 #define LCD_DC_GPIO_CLK_ENABLE()

```

```

    __HAL_RCC_GPIOF_CLK_ENABLE()
00266 #define LCD_DC_GPIO_CLK_DISABLE()
    __HAL_RCC_GPIOF_CLK_DISABLE()
00267
00268 #endif /* HAL_SPI_MODULE_ENABLED */
00269
00270 /*##### ADCx for
Nucleo 144 board #####
#####*/
00271 /**
00272  * @brief ADCx Interface pins
00273  *          used to detect motion of Joystic
k available on Adafruit 1.8" TFT shield
00274  */
00275
00276 #ifdef HAL_ADC_MODULE_ENABLED
00277
00278 #define NUCLEO_ADCx                                A
DC3
00279 #define NUCLEO_ADCx_CLK_ENABLE()                    __H
AL_RCC_ADC3_CLK_ENABLE()
00280 #define NUCLEO_ADCx_CLK_DISABLE()                  __H
AL_RCC_ADC3_CLK_DISABLE()
00281
00282 #define NUCLEO_ADCx_CHANNEL                          A
DC_CHANNEL_9
00283
00284 #define NUCLEO_ADCx_GPIO_PORT                        G
PIOF
00285 #define NUCLEO_ADCx_GPIO_PIN                        G
PIO_PIN_3
00286 #define NUCLEO_ADCx_GPIO_CLK_ENABLE()              __H
AL_RCC_GPIOF_CLK_ENABLE()
00287 #define NUCLEO_ADCx_GPIO_CLK_DISABLE()              __H
AL_RCC_GPIOF_CLK_DISABLE()
00288
00289 #endif /* HAL_ADC_MODULE_ENABLED */

```

```

00290
00291 /**
00292  * @}
00293  */
00294
00295 /**
00296  * @}
00297  */
00298
00299 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
_Exported_Macros STM32F2XX NUCLEO 144 LOW LEVEL Ex
ported Macros
00300  * @{
00301  */
00302 /**
00303  * @}
00304  */
00305
00306 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
_Exported_Functions STM32F2XX NUCLEO 144 LOW LEVEL
Exported Functions
00307  * @{
00308  */
00309 uint32_t      BSP_GetVersion(void);
00310 void          BSP_LED_Init(Led_TypeDef Le
d);
00311 void          BSP_LED_DeInit(Led_TypeDef
Led);
00312 void          BSP_LED_On(Led_TypeDef Led)
;
00313 void          BSP_LED_Off(Led_TypeDef Led
);
00314 void          BSP_LED_Toggle(Led_TypeDef
Led);
00315 void          BSP_PB_Init(Button_TypeDef
Button, ButtonMode_TypeDef ButtonMode);
00316 void          BSP_PB_DeInit(Button_TypeDef

```

```

    Button);
00317 uint32_t          BSP_PB_GetState(Button_Type
Def Button);
00318 #ifdef HAL_ADC_MODULE_ENABLED
00319 uint8_t             BSP_JOY_Init(void);
00320 JOYState_TypeDef    BSP_JOY_GetState(void);
00321 void               BSP_JOY_DeInit(void);
00322 #endif /* HAL_ADC_MODULE_ENABLED */
00323
00324
00325 /**
00326  * @}
00327  */
00328
00329 /**
00330  * @}
00331  */
00332
00333 /**
00334  * @}
00335  */
00336
00337 /**
00338  * @}
00339  */
00340
00341 #ifdef __cplusplus
00342 }
00343 #endif
00344
00345 #endif /* __STM32F2XX_NUCLEO_144_H */
00346
00347 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE*****/

```

BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

Main Page	Modules	Files	Directories	
File List	Globals			
Drivers	BSP	STM32F2xx_Nucleo_144		

stm32f2xx_nucleo_144.c

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

```
00001  /**
00002      ****
00003      ****
00004      * @file      stm32f2xx_nucleo_144.c
00005      * @author    MCD Application Team
00006      * @version    V1.0.0
00007      * @date      20-November-2015
00008      * @brief     This file provides set of firmw
are functions to manage:
00009      *             - LEDs and push-button availabl
e on STM32F2XX-Nucleo-144 Kit
00010      *             from STMicroelectronics
00011      *             - LCD, joystick and microSD ava
ilable on Adafruit 1.8" TFT LCD
00012      *             shield (reference ID 802)
00013      ****
00014      ****
00015      * @attention
00016      *
00017      * <h2><center>&copy; COPYRIGHT(c) 2017 STM
icroelectronics</center></h2>
00018      *
00019      * Redistribution and use in source and bin
ary forms, with or without modification,
```

00018 * are permitted provided that the following conditions are met:

00019 * 1. Redistributions of source code must retain the above copyright notice,

00020 * this list of conditions and the following disclaimer.

00021 * 2. Redistributions in binary form must reproduce the above copyright notice,

00022 * this list of conditions and the following disclaimer in the documentation

00023 * and/or other materials provided with the distribution.

00024 * 3. Neither the name of STMicroelectronics nor the names of its contributors

00025 * may be used to endorse or promote products derived from this software

00026 * without specific prior written permission.

00027 *

00028 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

00029 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

00030 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

00031 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

00032 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

00033 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

00034 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

00035 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

00036 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE


```

00037      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
      POSSIBILITY OF SUCH DAMAGE.
00038      *
00039      ****
      ****
00040      */
00041
00042 /* Includes -----
----- */
00043 #include "stm32f2xx_nucleo_144.h"
00044
00045
00046 /** @defgroup BSP BSP
00047     * @{
00048     */
00049
00050 /** @defgroup STM32F2XX_NUCLEO_144 STM32F2XX
    NUCLEO 144
00051     * @{
00052     */
00053
00054 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
    STM32F2XX NUCLEO 144 LOW LEVEL
00055     * @brief This file provides set of firmwar
    e functions to manage Leds and push-button
00056     *         available on STM32F2xx-Nucleo Kit
    from STMicroelectronics.
00057     * @{
00058     */
00059
00060 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
    _Private_TypesDefinitions STM32F2XX NUCLEO 144 LOW
    LEVEL Private TypesDefinitions
00061     * @{
00062     */
00063 /**
00064     * @}

```

```

00065    */
00066
00067
00068 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
_Private_Defines STM32F2XX NUCLEO 144 LOW LEVEL Pr
ivate Defines
00069    * @{
00070    */
00071
00072 /**
00073    * @brief STM32F2xx NUCLEO BSP Driver versi
on number V1.0.1
00074    */
00075 #define __STM32F2xx_NUCLEO_BSP_VERSION_MAIN
(0x01) /*!< [31:24] main version */
00076 #define __STM32F2xx_NUCLEO_BSP_VERSION_SUB1
(0x00) /*!< [23:16] sub1 version */
00077 #define __STM32F2xx_NUCLEO_BSP_VERSION_SUB2
(0x01) /*!< [15:8] sub2 version */
00078 #define __STM32F2xx_NUCLEO_BSP_VERSION_RC
(0x00) /*!< [7:0] release candidate */
00079 #define __STM32F2xx_NUCLEO_BSP_VERSION
(( __STM32F2xx_NUCLEO_BSP_VERSION_MAIN << 24) |\
00080
((__STM32F2xx_NUCLEO_BSP_VERSION_SUB1 << 16) |\
00081
((__STM32F2xx_NUCLEO_BSP_VERSION_SUB2 << 8 ) |\
00082
((__STM32F2xx_NUCLEO_BSP_VERSION_RC)))
00083
00084 /**
00085    * @brief LINK SD Card
00086    */
00087 #define SD_DUMMY_BYTE 0xFF
00088 #define SD_NO_RESPONSE_EXPECTED 0x80
00089
00090 /**

```

```

00091     * @}
00092     */
00093
00094 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
    _Private_Macros STM32F2XX NUCLEO 144 LOW LEVEL Pri
    vate Macros
00095     * @{
00096     */
00097 /**
00098     * @}
00099     */
00100
00101 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
    _Private_Variables STM32F2XX NUCLEO 144 LOW LEVEL
    Private Variables
00102     * @{
00103     */
00104 GPIO_TypeDef* GPIO_PORT[LEDn] = {LED1_GPIO_P
    ORT, LED2_GPIO_PORT, LED3_GPIO_PORT};
00105
00106 const uint16_t GPIO_PIN[LEDn] = {LED1_PIN, L
    ED2_PIN, LED3_PIN};
00107
00108 GPIO_TypeDef* BUTTON_PORT[BUTTONn] = {USER_B
    UTTON_GPIO_PORT};
00109 const uint16_t BUTTON_PIN[BUTTONn] = {USER_B
    UTTON_PIN};
00110 const uint8_t BUTTON_IRQn[BUTTONn] = {USER_B
    UTTON_EXTI_IRQn};
00111
00112 /**
00113     * @brief BUS variables
00114     */
00115
00116 #ifdef ADAFRUIT_TFT_JOY_SD_ID802
00117 #ifdef HAL_SPI_MODULE_ENABLED
00118 uint32_t SpixTimeout = NUCLEO_SPIx_TIMEOUT_M

```

```

AX; /*<! Value of Timeout when SPI communication fails */
00119 static SPI_HandleTypeDef hnucleo_Spi;
00120 #endif /* HAL_SPI_MODULE_ENABLED */
00121
00122 #ifdef HAL_ADC_MODULE_ENABLED
00123 static ADC_HandleTypeDef hnucleo_Adc;
00124 /* ADC channel configuration structure declaration */
00125 static ADC_ChannelConfTypeDef sConfig;
00126 #endif /* HAL_ADC_MODULE_ENABLED */
00127 #endif /* ADAFRUIT_TFT_JOY_SD_ID802 */
00128
00129 /**
00130  * @}
00131  */
00132
00133 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
00134  * @private
00135  * Private Function Prototypes
00136  * @{
00137  */
00138 #ifdef HAL_SPI_MODULE_ENABLED
00139 static void SPIx_Init(void);
00140 static void SPIx_Write(uint8_t Value);
00141 static void SPIx_Error(void);
00142 static void SPIx_MspInit(SPI_HandleTypeDef *hspi);
00143
00144 /* SD IO functions */
00145 void SD_IO_Init(void);
00146 void SD_IO_CSState(uint8_t state);
00147 void SD_IO_WriteReadData(const uint8_t *DataIn, uint8_t *DataOut, uint16_t DataLength);
00148 uint8_t SD_IO_WriteByte(uint8_t Data);

```

```

00149
00150 /* LCD IO functions */
00151 void LCD_IO_Init(void);
00152 void LCD_IO_WriteData(uint8_t Data);
00153 void LCD_IO_WriteMultipleData(uint8_t *pData
, uint32_t Size);
00154 void LCD_IO_WriteReg(uint8_t LCDReg);
00155 void LCD_Delay(uint32_t delay);
00156 #endif /* HAL_SPI_MODULE_ENABLED */
00157
00158 #ifdef HAL_ADC_MODULE_ENABLED
00159 static void ADCx_Init(void);
00160 static void ADCx_DeInit(void);
00161 static void ADCx_MspInit(ADC_HandleTypeDef *
hadc);
00162 static void ADCx_MspDeInit(ADC_HandleTypeDef
*hadc);
00163 #endif /* HAL_ADC_MODULE_ENABLED */
00164
00165 #endif /* ADAFRUIT_TFT_JOY_SD_ID802 */
00166
00167 /**
00168  * @}
00169  */
00170
00171 /** @defgroup STM32F2XX_NUCLEO_144_LOW_LEVEL
_Private_Functions STM32F2XX NUCLEO 144 LOW LEVEL
Private Functions
00172  * @{
00173  */
00174
00175 /**
00176  * @brief This method returns the STM32F2x
x NUCLEO BSP Driver revision
00177  * @retval version: 0xXYZR (8bits for each
decimal, R for RC)
00178  */

```

```

00179 uint32_t BSP_GetVersion(void)
00180 {
00181     return __STM32F2xx_NUCLEO_BSP_VERSION;
00182 }
00183
00184 /**
00185  * @brief Configures LED GPIO.
00186  * @param Led: Specifies the Led to be con
00187  *           figured.
00188  *           This parameter can be one of following
00189  *           parameters:
00190  *           @arg LED1
00191  *           @arg LED2
00192  *           @arg LED3
00193  */
00194 void BSP_LED_Init(Led_TypeDef Led)
00195 {
00196     GPIO_InitTypeDef GPIO_InitStructure;
00197     /* Enable the GPIO_LED Clock */
00198     LEDx_GPIO_CLK_ENABLE(Led);
00199     /* Configure the GPIO_LED pin */
00200     GPIO_InitStructure.Pin = GPIO_PIN[Led];
00201     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP
00202     ;
00203     GPIO_InitStructure.Pull = GPIO_NOPULL;
00204     GPIO_InitStructure.Speed = GPIO_SPEED_FAST;
00205     HAL_GPIO_Init(GPIO_PORT[Led], &GPIO_InitSt
00206     ruct);
00207     HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN
00208     [Led], GPIO_PIN_RESET);
00209 }
00210 /**
00211  * @brief DeInit LEDs.

```

```

00211     * @param Led: LED to be de-init.
00212     *     This parameter can be one of the following values:
00213     *         @arg LED1
00214     *         @arg LED2
00215     *         @arg LED3
00216     * @note Led DeInit does not disable the GPIO clock nor disable the Mfx
00217     */
00218 void BSP_LED_DeInit(Led_TypeDef Led)
00219 {
00220     GPIO_InitTypeDef  gpio_init_structure;
00221
00222     /* Turn off LED */
00223     HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN[Led], GPIO_PIN_RESET);
00224     /* DeInit the GPIO_LED pin */
00225     gpio_init_structure.Pin = GPIO_PIN[Led];
00226     HAL_GPIO_DeInit(GPIO_PORT[Led], gpio_init_structure.Pin);
00227 }
00228
00229 /**
00230  * @brief Turns selected LED On.
00231  * @param Led: Specifies the Led to be set on.
00232  *     This parameter can be one of following parameters:
00233  *         @arg LED2
00234  */
00235 void BSP_LED_On(Led_TypeDef Led)
00236 {
00237     HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN[Led], GPIO_PIN_SET);
00238 }
00239
00240 /**

```

```

00241     * @brief Turns selected LED Off.
00242     * @param Led: Specifies the Led to be set
    off.
00243     * This parameter can be one of following
    parameters:
00244     * @arg LED1
00245     * @arg LED2
00246     * @arg LED3
00247     */
00248 void BSP_LED_Off(Led_TypeDef Led)
00249 {
00250     HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN
[Led], GPIO_PIN_RESET);
00251 }
00252
00253 /**
00254     * @brief Toggles the selected LED.
00255     * @param Led: Specifies the Led to be tog
    gled.
00256     * This parameter can be one of following
    parameters:
00257     * @arg LED1
00258     * @arg LED2
00259     * @arg LED3
00260     */
00261 void BSP_LED_Toggle(Led_TypeDef Led)
00262 {
00263     HAL_GPIO_TogglePin(GPIO_PORT[Led], GPIO_PIN
[Led]);
00264 }
00265
00266 /**
00267     * @brief Configures Button GPIO and EXTI
    Line.
00268     * @param Button: Specifies the Button to
    be configured.
00269     * This parameter should be: BUTTON_USER

```



```

00270      * @param ButtonMode: Specifies Button mode.
00271      *      This parameter can be one of following
        parameters:
00272      *      @arg BUTTON_MODE_GPIO: Button will be used as simple IO
00273      *      @arg BUTTON_MODE_EXTI: Button will be connected to EXTI line with interrupt
00274      *      generation capability
00275      */
00276 void BSP_PB_Init(Button_TypeDef Button, ButtonMode_TypeDef ButtonMode)
00277 {
00278     GPIO_InitTypeDef GPIO_InitStructure;
00279
00280     /* Enable the BUTTON Clock */
00281     BUTTONx_GPIO_CLK_ENABLE(Button);
00282
00283     if(ButtonMode == BUTTON_MODE_GPIO)
00284     {
00285         /* Configure Button pin as input */
00286         GPIO_InitStructure.Pin = BUTTON_PIN[Button]
;
00287         GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
00288         GPIO_InitStructure.Pull = GPIO_PULLDOWN;
00289         GPIO_InitStructure.Speed = GPIO_SPEED_FAST;
00290         HAL_GPIO_Init(BUTTON_PORT[Button], &GPIO
_InitStructure);
00291     }
00292
00293     if(ButtonMode == BUTTON_MODE_EXTI)
00294     {
00295         /* Configure Button pin as input with Ex
ternal interrupt */
00296         GPIO_InitStructure.Pin = BUTTON_PIN[Button]
;

```

```

00297     GPIO_InitStruct.Pull = GPIO_NOPULL;
00298     GPIO_InitStruct.Mode = GPIO_MODE_IT_FALL
ING;
00299     HAL_GPIO_Init(BUTTON_PORT[Button], &GPIO
_InitStruct);
00300
00301     /* Enable and set Button EXTI Interrupt
to the lowest priority */
00302     HAL_NVIC_SetPriority((IRQn_Type)(BUTTON_
IRQn[Button]), 0x0F, 0x00);
00303     HAL_NVIC_EnableIRQ((IRQn_Type)(BUTTON_IR
Qn[Button]));
00304 }
00305 }
00306
00307 /**
00308  * @brief Push Button DeInit.
00309  * @param Button: Button to be configured
00310  * This parameter should be: BUTTON_USER
00311  * @note PB DeInit does not disable the GPI
O clock
00312  */
00313 void BSP_PB_DeInit(Button_TypeDef Button)
00314 {
00315     GPIO_InitTypeDef gpio_init_structure;
00316
00317     gpio_init_structure.Pin = BUTTON_PIN[Butto
n];
00318     HAL_NVIC_DisableIRQ((IRQn_Type)(BUTTON_IRQn
[Button]));
00319     HAL_GPIO_DeInit(BUTTON_PORT[Button], gpio_
init_structure.Pin);
00320 }
00321
00322 /**
00323  * @brief Returns the selected Button stat
e.

```

```

00324     * @param Button: Specifies the Button to
be checked.
00325     *     This parameter should be: BUTTON_USER

00326     * @retval The Button GPIO pin value.
00327     */
00328 uint32_t BSP_PB_GetState(Button_TypeDef Butt
on)
00329 {
00330     return HAL_GPIO_ReadPin(BUTTON_PORT[Button
], BUTTON_PIN[Button]);
00331 }
00332
00333 /*****
*****
00334                                     BUS OPERATIONS
00335     *****/
00336 #ifdef ADAFRUIT_TFT_JOY_SD_ID802
00337
00338 /***** SPI *****/
00339 #ifdef HAL_SPI_MODULE_ENABLED
00340
00341 /**
00342     * @brief Initializes SPI MSP.
00343     */
00344 static void SPIx_MspInit(SPI_HandleTypeDef *
hspi)
00345 {
00346     GPIO_InitTypeDef GPIO_InitStruct;
00347
00348     /*** Configure the GPIOs ***/
00349     /* Enable GPIO clock */
00350     NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE();
00351     NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE();
00352

```

```

00353  /* Configure SPI SCK */
00354  GPIO_InitStruct.Pin = NUCLEO_SPIx_SCK_PIN;
00355  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
00356  GPIO_InitStruct.Pull  = GPIO_PULLUP;
00357  GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
00358  GPIO_InitStruct.Alternate = NUCLEO_SPIx_SCK_AF;
00359  HAL_GPIO_Init(NUCLEO_SPIx_SCK_GPIO_PORT, &
GPIO_InitStruct);
00360
00361  /* Configure SPI MISO and MOSI */
00362  GPIO_InitStruct.Pin = NUCLEO_SPIx_MOSI_PIN
;
00363  GPIO_InitStruct.Alternate = NUCLEO_SPIx_MISO_MOSI_AF;
00364  GPIO_InitStruct.Pull  = GPIO_PULLDOWN;
00365  HAL_GPIO_Init(NUCLEO_SPIx_MISO_MOSI_GPIO_P
ORT, &GPIO_InitStruct);
00366
00367  GPIO_InitStruct.Pin = NUCLEO_SPIx_MISO_PIN
;
00368  GPIO_InitStruct.Pull  = GPIO_PULLDOWN;
00369  HAL_GPIO_Init(NUCLEO_SPIx_MISO_MOSI_GPIO_P
ORT, &GPIO_InitStruct);
00370
00371  /*** Configure the SPI peripheral ***/
00372  /* Enable SPI clock */
00373  NUCLEO_SPIx_CLK_ENABLE();
00374 }
00375
00376 /**
00377  * @brief Initializes SPI HAL.
00378  */
00379 static void SPIx_Init(void)
00380 {
00381     if(HAL_SPI_GetState(&hnucleo_Spi) == HAL_S
PI_STATE_RESET)

```

```

00382     {
00383         /* SPI Config */
00384         hnucleo_Spi.Instance = NUCLEO_SPIx;
00385         /* SPI configuration constraints
00386             - ST7735 LCD SPI interface max baud
rate is 15MHz for write and 6.66MHz for read
00387             Since the provided driver doesn'
t use read capability from LCD, only constraint
00388             on write baudrate is considered.
00389             - SD card SPI interface max baudra
te is 25MHz for write/read
00390             to feat these constraints SPI baudrat
e is set to:
00391             - For STM32F207xx devices: 10,
5 MHz maximum (PCLK2/SPI_BAUDRATEPRESCALER_8 = 84
MHz/8 = 10,5 MHz)
00392         */
00393         hnucleo_Spi.Init.BaudRatePrescaler = SPI
_BAUDRATEPRESCALER_8;
00394         hnucleo_Spi.Init.Direction = SPI_DIRECTI
ON_2LINES;
00395         hnucleo_Spi.Init.CLKPhase = SPI_PHASE_2E
DGE;
00396         hnucleo_Spi.Init.CLKPolarity = SPI_POLAR
ITY_HIGH;
00397         hnucleo_Spi.Init.CRCCalculation = SPI_CR
CCALCULATION_DISABLED;
00398         hnucleo_Spi.Init.CRCPolynomial = 7;
00399         hnucleo_Spi.Init.DataSize = SPI_DATASIZE
_8BIT;
00400         hnucleo_Spi.Init.FirstBit = SPI_FIRSTBIT
_MSB;
00401         hnucleo_Spi.Init.NSS = SPI_NSS_SOFT;
00402         hnucleo_Spi.Init.TIMode = SPI_TIMODE_DIS
ABLED;
00403         hnucleo_Spi.Init.Mode = SPI_MODE_MASTER;
00404

```

```

00405     SPIx_MspInit(&hnucleo_Spi);
00406     HAL_SPI_Init(&hnucleo_Spi);
00407 }
00408 }
00409
00410 /**
00411  * @brief SPI Write a byte to device
00412  * @param DataIn: value to be written
00413  * @param DataOut: value to read
00414  * @param DataLegnth: length of data
00415  */
00416 static void SPIx_WriteReadData(const uint8_t
    *DataIn, uint8_t *DataOut, uint16_t DataLegnth)
00417 {
00418     HAL_StatusTypeDef status = HAL_OK;
00419
00420     status = HAL_SPI_TransmitReceive(&hnucleo_
Spi, (uint8_t*) DataIn, DataOut, DataLegnth, SpiXT
imeout);
00421
00422     /* Check the communication status */
00423     if(status != HAL_OK)
00424     {
00425         /* Execute user timeout callback */
00426         SPIx_Error();
00427     }
00428 }
00429
00430 /**
00431  * @brief SPI Write a byte to device.
00432  * @param Value: value to be written
00433  */
00434 static void SPIx_Write(uint8_t Value)
00435 {
00436     HAL_StatusTypeDef status = HAL_OK;
00437     uint8_t data;
00438

```

```

00439     status = HAL_SPI_TransmitReceive(&hnucleo_
Spi, (uint8_t*) &Value, &data, 1, SpiTimeout);
00440
00441     /* Check the communication status */
00442     if(status != HAL_OK)
00443     {
00444         /* Execute user timeout callback */
00445         SPIx_Error();
00446     }
00447 }
00448
00449 /**
00450  * @brief SPI error treatment function
00451  */
00452 static void SPIx_Error (void)
00453 {
00454     /* De-initialize the SPI communication BUS
00455     */
00456     HAL_SPI_DeInit(&hnucleo_Spi);
00457     /* Re-Initiaize the SPI communication BUS
00458     */
00459     SPIx_Init();
00460 }
00461
00462     LINK OPERATIONS
00463     *****
00464     /
00465     ***** LINK SD *
00466     /
00467     /**
00468     * @brief Initializes the SD Card and put
00469     it into StandBy State (Ready for
00470     * data transfer).

```

```

00469     */
00470 void SD_IO_Init(void)
00471 {
00472     GPIO_InitTypeDef  GPIO_InitStructure;
00473     uint8_t counter;
00474
00475     /* SD_CS_GPIO Periph clock enable */
00476     SD_CS_GPIO_CLK_ENABLE();
00477
00478     /* Configure SD_CS_PIN pin: SD Card CS pin
00479     */
00479     GPIO_InitStructure.Pin = SD_CS_PIN;
00480     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP
00481     ;
00481     GPIO_InitStructure.Pull = GPIO_PULLUP;
00482     GPIO_InitStructure.Speed = GPIO_SPEED_HIGH;
00483     HAL_GPIO_Init(SD_CS_GPIO_PORT, &GPIO_InitS
00484     truct);
00484
00485
00486     /* LCD chip select line perturbs SD also
00487     when the LCD is not used */
00487     /* this is a workaround to avoid sporadic
00488     failures during r/w operations */
00488     LCD_CS_GPIO_CLK_ENABLE();
00489     GPIO_InitStructure.Pin = LCD_CS_PIN;
00490     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP
00491     ;
00491     GPIO_InitStructure.Pull = GPIO_NOPULL;
00492     GPIO_InitStructure.Speed = GPIO_SPEED_HIGH;
00493     HAL_GPIO_Init(LCD_CS_GPIO_PORT, &GPIO_Init
00494     Struct);
00494     LCD_CS_HIGH();
00495
00496     /*-----Put SD in SPI mode-----
00497     -----*/
00497     /* SD SPI Config */

```



```

00498     SPIx_Init();
00499
00500     /* SD chip select high */
00501     SD_CS_HIGH();
00502
00503     /* Send dummy byte 0xFF, 10 times with CS
high */
00504     /* Rise CS and MOSI for 80 clocks cycles */

00505     for (counter = 0; counter <= 9; counter++)
00506     {
00507         /* Send dummy byte 0xFF */
00508         SD_IO_WriteByte(SD_DUMMY_BYTE);
00509     }
00510 }
00511
00512 /**
00513  * @brief Set the SD_CS pin.
00514  * @param val: pin value.
00515  */
00516 void SD_IO_CSState(uint8_t val)
00517 {
00518     if(val == 1)
00519     {
00520         SD_CS_HIGH();
00521     }
00522     else
00523     {
00524         SD_CS_LOW();
00525     }
00526 }
00527
00528 /**
00529  * @brief Write a byte on the SD.
00530  * @param DataIn: byte to send.
00531  * @param DataOut: byte to read
00532  * @param DataLength: length of data

```

```

00533     */
00534 void SD_IO_WriteReadData(const uint8_t *Data
In, uint8_t *DataOut, uint16_t DataLength)
00535 {
00536     /* Send the byte */
00537     SPIx_WriteReadData(DataIn, DataOut, DataLe
ngth);
00538 }
00539
00540 /**
00541  * @brief Writes a byte on the SD.
00542  * @param Data: byte to send.
00543  */
00544 uint8_t SD_IO_WriteByte(uint8_t Data)
00545 {
00546     uint8_t tmp;
00547     /* Send the byte */
00548     SPIx_WriteReadData(&Data,&tmp,1);
00549     return tmp;
00550 }
00551
00552 /***** LINK LCD
***** */
00553 /**
00554  * @brief Initializes the LCD
00555  */
00556 void LCD_IO_Init(void)
00557 {
00558     GPIO_InitTypeDef  GPIO_InitStruct;
00559
00560     /* LCD_CS_GPIO and LCD_DC_GPIO Periph cloc
k enable */
00561     LCD_CS_GPIO_CLK_ENABLE();
00562     LCD_DC_GPIO_CLK_ENABLE();
00563
00564     /* Configure LCD_CS_PIN pin: LCD Card CS p
in */

```

```

00565     GPIO_InitStruct.Pin = LCD_CS_PIN;
00566     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP
;
00567     GPIO_InitStruct.Pull = GPIO_NOPULL;
00568     GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
00569     HAL_GPIO_Init(LCD_CS_GPIO_PORT, &GPIO_Init
Struct);
00570
00571     /* Configure LCD_DC_PIN pin: LCD Card DC p
in */
00572     GPIO_InitStruct.Pin = LCD_DC_PIN;
00573     HAL_GPIO_Init(LCD_DC_GPIO_PORT, &GPIO_Init
Struct);
00574
00575     /* LCD chip select high */
00576     LCD_CS_HIGH();
00577
00578     /* LCD SPI Config */
00579     SPIx_Init();
00580 }
00581
00582 /**
00583  * @brief Writes command to select the LCD
    register.
00584  * @param LCDReg: Address of the selected
    register.
00585  */
00586 void LCD_IO_WriteReg(uint8_t LCDReg)
00587 {
00588     /* Reset LCD control line CS */
00589     LCD_CS_LOW();
00590
00591     /* Set LCD data/command line DC to Low */
00592     LCD_DC_LOW();
00593
00594     /* Send Command */
00595     SPIx_Write(LCDReg);

```

```

00596
00597     /* Deselect : Chip Select high */
00598     LCD_CS_HIGH();
00599 }
00600
00601 /**
00602  * @brief Writes data to select the LCD re
gister.
00603  *          This function must be used after
st7735_WriteReg() function
00604  * @param Data: data to write to the selec
ted register.
00605  */
00606 void LCD_IO_WriteData(uint8_t Data)
00607 {
00608     /* Reset LCD control line CS */
00609     LCD_CS_LOW();
00610
00611     /* Set LCD data/command line DC to High */
00612     LCD_DC_HIGH();
00613
00614     /* Send Data */
00615     SPIx_Write(Data);
00616
00617     /* Deselect : Chip Select high */
00618     LCD_CS_HIGH();
00619 }
00620
00621 /**
00622  * @brief Write register value.
00623  * @param pData Pointer on the register va
lue
00624  * @param Size Size of byte to transmit to
the register
00625  */
00626 void LCD_IO_WriteMultipleData(uint8_t *pData
, uint32_t Size)

```

```

00627 {
00628     uint32_t counter = 0;
00629     __IO uint32_t data = 0;
00630
00631     /* Reset LCD control line CS */
00632     LCD_CS_LOW();
00633
00634     /* Set LCD data/command line DC to High */
00635     LCD_DC_HIGH();
00636
00637     if (Size == 1)
00638     {
00639         /* Only 1 byte to be sent to LCD - gener
al interface can be used */
00640         /* Send Data */
00641         SPIx_Write(*pData);
00642     }
00643     else
00644     {
00645         /* Several data should be sent in a raw
*/
00646         /* Direct SPI accesses for optimization
*/
00647         for (counter = Size; counter != 0; count
er--)
00648         {
00649             while(((hnucleo_Spi.Instance->SR) & SP
I_FLAG_TXE) != SPI_FLAG_TXE)
00650             {
00651             }
00652             /* Need to invert bytes for LCD*/
00653             *((__IO uint8_t*)&hnucleo_Spi.Instance
->DR) = *(pData+1);
00654
00655             while(((hnucleo_Spi.Instance->SR) & SP
I_FLAG_TXE) != SPI_FLAG_TXE)
00656             {

```

```

00657     }
00658     *((__IO uint8_t*)&hnucleo_Spi.Instance
->DR) = *pData;
00659     counter--;
00660     pData += 2;
00661 }
00662
00663     /* Wait until the bus is ready before re
leasing Chip select */
00664     while(((hnucleo_Spi.Instance->SR) & SPI_
FLAG_BSY) != RESET)
00665     {
00666     }
00667 }
00668
00669     /* Empty the Rx fifo */
00670     data = (&hnucleo_Spi.Instance->DR);
00671     UNUSED(data); /* Remove GNU warning */
00672
00673     /* Deselect : Chip Select high */
00674     LCD_CS_HIGH();
00675 }
00676
00677 /**
00678  * @brief Wait for loop in ms.
00679  * @param Delay in ms.
00680  */
00681 void LCD_Delay(uint32_t Delay)
00682 {
00683     HAL_Delay(Delay);
00684 }
00685 #endif /* HAL_SPI_MODULE_ENABLED */
00686
00687 /***** ADC driver
***** */
00688 #ifdef HAL_ADC_MODULE_ENABLED
00689

```

```

00690 /**
00691  * @brief Initializes ADC MSP.
00692  */
00693 static void ADCx_MspInit(ADC_HandleTypeDef *
hadc)
00694 {
00695     GPIO_InitTypeDef  GPIO_InitStructure;
00696
00697     /*** Configure the GPIOs ***/
00698     /* Enable GPIO clock */
00699     NUCLEO_ADCx_GPIO_CLK_ENABLE();
00700
00701     /* Configure the selected ADC Channel as a
analog input */
00702     GPIO_InitStructure.Pin = NUCLEO_ADCx_GPIO_PIN
;
00703     GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
00704     GPIO_InitStructure.Pull = GPIO_NOPULL;
00705     HAL_GPIO_Init(NUCLEO_ADCx_GPIO_PORT, &GPIO
_InitStruct);
00706
00707     /*** Configure the ADC peripheral ***/
00708     /* Enable ADC clock */
00709     NUCLEO_ADCx_CLK_ENABLE();
00710 }
00711
00712 /**
00713  * @brief DeInitializes ADC MSP.
00714  * @note ADC DeInit does not disable the GP
IO clock
00715  */
00716 static void ADCx_MspDeInit(ADC_HandleTypeDef
*hadc)
00717 {
00718     GPIO_InitTypeDef  GPIO_InitStructure;
00719
00720     /*** DeInit the ADC peripheral ***/

```

```

00721  /* Disable ADC clock */
00722  NUCLEO_ADCx_CLK_DISABLE();
00723
00724  /* Configure the selected ADC Channel as a
analog input */
00725  GPIO_InitStruct.Pin = NUCLEO_ADCx_GPIO_PIN
;
00726  HAL_GPIO_DeInit(NUCLEO_ADCx_GPIO_PORT, GPI
O_InitStruct.Pin);
00727
00728  /* Disable GPIO clock has to be done by th
e application*/
00729  /* NUCLEO_ADCx_GPIO_CLK_DISABLE(); */
00730 }
00731
00732 /**
00733  * @brief Initializes ADC HAL.
00734  */
00735 static void ADCx_Init(void)
00736 {
00737     if(HAL_ADC_GetState(&hnucleo_Adc) == HAL_A
DC_STATE_RESET)
00738     {
00739         /* ADC Config */
00740         hnucleo_Adc.Instance =
NUCLEO_ADCx;
00741         hnucleo_Adc.Init.ClockPrescaler =
ADC_CLOCKPRESCALER_PCLK_DIV4; /* (must not exceed
36MHz) */
00742         hnucleo_Adc.Init.Resolution =
ADC_RESOLUTION12b;
00743         hnucleo_Adc.Init.DataAlign =
ADC_DATAALIGN_RIGHT;
00744         hnucleo_Adc.Init.ContinuousConvMode =
DISABLE;
00745         hnucleo_Adc.Init.DiscontinuousConvMode =
DISABLE;

```



```

00746     hnucleo_Adc.Init.ExternalTrigConvEdge  =
    ADC_EXTERNALTRIGCONVEDGE_NONE;
00747     hnucleo_Adc.Init.EOCSelection           =
    EOC_SINGLE_CONV;
00748     hnucleo_Adc.Init.NbrOfConversion       =
    1;
00749     hnucleo_Adc.Init.DMAContinuousRequests =
    DISABLE;
00750
00751     ADCx_MspInit(&hnucleo_Adc);
00752     HAL_ADC_Init(&hnucleo_Adc);
00753 }
00754 }
00755
00756 /**
00757  * @brief Initializes ADC HAL.
00758  */
00759 static void ADCx_DeInit(void)
00760 {
00761     hnucleo_Adc.Instance = NUCLEO_ADCx;
00762
00763     HAL_ADC_DeInit(&hnucleo_Adc);
00764     ADCx_MspDeInit(&hnucleo_Adc);
00765 }
00766
00767 /** ***** LINK JOYSTI
CK ***** */
00768
00769 /**
00770  * @brief Configures joystick available on
    adafruit 1.8" TFT shield
00771  *          managed through ADC to detect mo
    tion.
00772  * @retval Joystickstatus (0=> success, 1=>
    fail)
00773  */
00774 uint8_t BSP_JOY_Init(void)

```

```

00775 {
00776     uint8_t status = HAL_ERROR;
00777
00778     ADCx_Init();
00779
00780     /* Select the ADC Channel to be converted
    */
00781     sConfig.Channel      = NUCLEO_ADCx_CHANNEL
    ;
00782     sConfig.SamplingTime = ADC_SAMPLETIME_3CYC
    LES;
00783     sConfig.Rank          = 1;
00784     status = HAL_ADC_ConfigChannel(&hnucleo_Adc
    , &sConfig);
00785
00786     /* Return Joystick initialization status */

00787     return status;
00788 }
00789
00790 /**
00791  * @brief DeInit joystick GPIOs.
00792  * @note JOY DeInit does not disable the
    Mfx, just set the Mfx pins in Off mode
00793  */
00794 void BSP_JOY_DeInit(void)
00795 {
00796     ADCx_DeInit();
00797 }
00798
00799 /**
00800  * @brief Returns the Joystick key pressed.

00801  * @note To know which Joystick key is pr
    essed we need to detect the voltage
00802  *         level on each key output
00803  *         - None : 3.3 V / 4095

```

```

00804      *          - SEL      : 1.055 V / 1308
00805      *          - DOWN    : 0.71 V / 88
00806      *          - LEFT    : 3.0 V / 3720
00807      *          - RIGHT   : 0.595 V / 737
00808      *          - UP      : 1.65 V / 2046
00809      * @retval JOYState_TypeDef: Code of the Joystick key pressed.
00810      */
00811 JOYState_TypeDef BSP_JOY_GetState(void)
00812 {
00813     JOYState_TypeDef state;
00814     uint16_t keyconvertedvalue = 0;
00815
00816     /* Start the conversion process */
00817     HAL_ADC_Start(&hnucleo_Adc);
00818
00819     /* Wait for the end of conversion */
00820     HAL_ADC_PollForConversion(&hnucleo_Adc, 10);
00821
00822     /* Check if the continuous conversion of regular channel is finished */
00823     if((HAL_ADC_GetState(&hnucleo_Adc) & HAL_ADC_STATE_EOC_REG) == HAL_ADC_STATE_EOC_REG)
00824     {
00825         /* Get the converted value of regular channel */
00826         keyconvertedvalue = HAL_ADC_GetValue(&hnucleo_Adc);
00827     }
00828
00829     if((keyconvertedvalue > 2010) && (keyconvertedvalue < 2090))
00830     {
00831         state = JOY_UP;
00832     }
00833     else if((keyconvertedvalue > 680) && (keyc

```

```

convertedvalue < 780))
00834     {
00835         state = JOY_RIGHT;
00836     }
00837     else if((keyconvertedvalue > 1270) && (key
convertedvalue < 1350))
00838     {
00839         state = JOY_SEL;
00840     }
00841     else if((keyconvertedvalue > 50) && (keyco
nvertedvalue < 130))
00842     {
00843         state = JOY_DOWN;
00844     }
00845     else if((keyconvertedvalue > 3680) && (key
convertedvalue < 3760))
00846     {
00847         state = JOY_LEFT;
00848     }
00849     else
00850     {
00851         state = JOY_NONE;
00852     }
00853
00854     /* Loop while a key is pressed */
00855     if(state != JOY_NONE)
00856     {
00857         keyconvertedvalue = HAL_ADC_GetValue(&h
ucleo_Adc);
00858     }
00859     /* Return the code of the Joystick key pre
ssed */
00860     return state;
00861 }
00862 #endif /* HAL_ADC_MODULE_ENABLED */
00863
00864 #endif /* ADAFRUIT_TFT_JOY_SD_ID802 */

```

```
00865
00866
00867  /*
00868   * @
00869   */
00870
00871  /*
00872   * @
00873   */
00874
00875  /*
00876   * @
00877   */
00878
00879  /*
00880   * @
00881   */
00882
00883  /***** (C) COPYRIGHT STMicroelectronics *****/
00884  *****END OF FILE*****/
```

STM32F2xx_Nucleo_144 BSP User Manual

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

BSP

Modules

STM32F2XX NUCLEO 144

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

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

STM32F2XX NUCLEO 144

[BSP](#)

Modules

STM32F2XX NUCLEO 144 LOW LEVEL

This file provides set of firmware functions to manage Leds and push-button available on STM32F2xx-Nucleo Kit from STMicroelectronics.

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by doxygen 1.7.6.1

STM32F2xx_Nucleo_144 BSP User Manual

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

STM32F2XX NUCLEO 144 LOW LEVEL Exported Constants

[STM32F2XX NUCLEO 144 LOW LEVEL](#)

Modules

STM32F2XX NUCLEO 144 LOW LEVEL LED

Define for STM32F2XX_NUCLEO_144 board.

STM32F2XX NUCLEO 144 LOW LEVEL BUTTON

STM32F2XX NUCLEO 144 LOW LEVEL BUS

Generated on Fri Jan 27 2017 11:11:58 for STM32F2xx_Nucleo_144
BSP User Manual by doxygen 1.7.6.1