STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Private TypesDefinitions

STM32F4XX NUCLEO LOW LEVEL

Generated on Fri Jan 20 2017 15:36:09 for STM32F4xx-Nucleo BSP User Manual by doxygen 1.7.6.1

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Private Macros

STM32F4XX NUCLEO LOW LEVEL

Generated on Fri Jan 20 2017 15:36:09 for STM32F4xx-Nucleo BSP User Manual by doxygen 1.7.6.1

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Exported Macros

STM32F4XX NUCLEO LOW LEVEL

Generated on Fri Jan 20 2017 15:36:09 for STM32F4xx-Nucleo BSP User Manual by doxygen 1.7.6.1

STM32F4xx-Nucleo BSP User Manual

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

- _ -

- __STM32F4xx_NUCLEO_BSP_VERSION : **stm32f4xx_nucleo.c**
- __STM32F4xx_NUCLEO_BSP_VERSION_MAIN : **stm32f4xx_nucleo.c**
- __STM32F4xx_NUCLEO_BSP_VERSION_RC : **stm32f4xx_nucleo.c**
- __STM32F4xx_NUCLEO_BSP_VERSION_SUB1 : **stm32f4xx_nucleo.c**
- __STM32F4xx_NUCLEO_BSP_VERSION_SUB2 : **stm32f4xx_nucleo.c**

- a -

- ADAFRUIT_TFT_JOY_SD_ID802 : **stm32f4xx_nucleo.h**
- ADCx_DeInit() : **stm32f4xx_nucleo.c**
- ADCx_Init() : **stm32f4xx_nucleo.c**
- ADCx_MspDeInit() : **stm32f4xx_nucleo.c**
- ADCx_MspInit() : **stm32f4xx_nucleo.c**

- b -

- BSP_GetVersion() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_JOY_DeInit() : **stm32f4xx_nucleo.h** , **stm32f4xx_nucleo.c**

- BSP_JOY_GetState() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_JOY_Init() : **stm32f4xx_nucleo.h** , **stm32f4xx_nucleo.c**
- BSP_LED_DeInit() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_LED_Init() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_LED_Off() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_LED_On() : **stm32f4xx_nucleo.h** , **stm32f4xx_nucleo.c**
- BSP_LED_Toggle() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_PB_DeInit() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_PB_GetState() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_PB_Init() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BUTTON_IRQn : **stm32f4xx_nucleo.c**
- BUTTON_KEY : **stm32f4xx_nucleo.h**
- BUTTON_MODE_EXTI : **stm32f4xx_nucleo.h**
- BUTTON_MODE_GPIO : **stm32f4xx_nucleo.h**
- BUTTON_PIN : **stm32f4xx_nucleo.c**
- BUTTON_PORT : **stm32f4xx_nucleo.c**
- Button_TypeDef : **stm32f4xx_nucleo.h**
- BUTTON_USER : **stm32f4xx_nucleo.h**
- ButtonMode_TypeDef : **stm32f4xx_nucleo.h**
- BUTTONn : **stm32f4xx_nucleo.h**
- BUTTONx_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- BUTTONx_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**

- g -

- GPIO_PIN : **stm32f4xx_nucleo.c**
- GPIO_PORT : **stm32f4xx_nucleo.c**

- h -

- hnucleo_Adc : **stm32f4xx_nucleo.c**
- hnucleo_Spi : **stm32f4xx_nucleo.c**

- j -

- JOY_DOWN : **stm32f4xx_nucleo.h**

- JOY_LEFT : **stm32f4xx_nucleo.h**
- JOY_NONE : **stm32f4xx_nucleo.h**
- JOY_RIGHT : **stm32f4xx_nucleo.h**
- JOY_SEL : **stm32f4xx_nucleo.h**
- JOY_UP : **stm32f4xx_nucleo.h**
- JOYState_TypeDef : **stm32f4xx_nucleo.h**

- k -

- KEY_BUTTON_EXTI_IRQn : **stm32f4xx_nucleo.h**
- KEY_BUTTON_EXTI_LINE : **stm32f4xx_nucleo.h**
- KEY_BUTTON_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- KEY_BUTTON_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- KEY_BUTTON_GPIO_PORT : **stm32f4xx_nucleo.h**
- KEY_BUTTON_PIN : **stm32f4xx_nucleo.h**

- l -

- LCD_CS_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- LCD_CS_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- LCD_CS_GPIO_PORT : **stm32f4xx_nucleo.h**
- LCD_CS_HIGH : **stm32f4xx_nucleo.h**
- LCD_CS_LOW : **stm32f4xx_nucleo.h**
- LCD_CS_PIN : **stm32f4xx_nucleo.h**
- LCD_DC_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- LCD_DC_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- LCD_DC_GPIO_PORT : **stm32f4xx_nucleo.h**
- LCD_DC_HIGH : **stm32f4xx_nucleo.h**
- LCD_DC_LOW : **stm32f4xx_nucleo.h**
- LCD_DC_PIN : **stm32f4xx_nucleo.h**
- LCD_Delay() : **stm32f4xx_nucleo.c**
- LCD_IO_Init() : **stm32f4xx_nucleo.c**
- LCD_IO_WriteData() : **stm32f4xx_nucleo.c**
- LCD_IO_WriteMultipleData() : **stm32f4xx_nucleo.c**
- LCD_IO_WriteReg() : **stm32f4xx_nucleo.c**
- LED2 : **stm32f4xx_nucleo.h**
- LED2_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- LED2_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**

- LED2_GPIO_PORT : **stm32f4xx_nucleo.h**
- LED2_PIN : **stm32f4xx_nucleo.h**
- Led_TypeDef : **stm32f4xx_nucleo.h**
- LEDn : **stm32f4xx_nucleo.h**
- LEDx_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- LEDx_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**

- n -

- NUCLEO_ADCx : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_CHANNEL : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_CLK_DISABLE : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_GPIO_PIN : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_GPIO_PORT : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_MOSI_AF : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_MOSI_GPIO_PORT : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_PIN : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MOSI_PIN : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_AF : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_GPIO_PORT : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_PIN : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_TIMEOUT_MAX : **stm32f4xx_nucleo.h**

- s -

- sConfig : **stm32f4xx_nucleo.c**
- SD_CS_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- SD_CS_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- SD_CS_GPIO_PORT : **stm32f4xx_nucleo.h**
- SD_CS_HIGH : **stm32f4xx_nucleo.h**
- SD_CS_LOW : **stm32f4xx_nucleo.h**
- SD_CS_PIN : **stm32f4xx_nucleo.h**
- SD_DUMMY_BYTE : **stm32f4xx_nucleo.c**
- SD_IO_CSState() : **stm32f4xx_nucleo.c**
- SD_IO_Init() : **stm32f4xx_nucleo.c**
- SD_IO_WriteByte() : **stm32f4xx_nucleo.c**
- SD_IO_WriteReadData() : **stm32f4xx_nucleo.c**
- SD_NO_RESPONSE_EXPECTED : **stm32f4xx_nucleo.c**
- SPIx_Error() : **stm32f4xx_nucleo.c**
- SPIx_Init() : **stm32f4xx_nucleo.c**
- SPIx_MspInit() : **stm32f4xx_nucleo.c**
- SPIx_Write() : **stm32f4xx_nucleo.c**
- SPIx_WriteReadData() : **stm32f4xx_nucleo.c**
- SpixTimeout : **stm32f4xx_nucleo.c**

- u -

- USER_BUTTON_EXTI_IRQn : **stm32f4xx_nucleo.h**
- USER_BUTTON_EXTI_LINE : **stm32f4xx_nucleo.h**
- USER_BUTTON_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- USER_BUTTON_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- USER_BUTTON_GPIO_PORT : **stm32f4xx_nucleo.h**
- USER_BUTTON_PIN : **stm32f4xx_nucleo.h**

STM32F4xx-Nucleo BSP User Manual

- a -

- ADCx_DeInit() : **stm32f4xx_nucleo.c**
- ADCx_Init() : **stm32f4xx_nucleo.c**
- ADCx_MspDeInit() : **stm32f4xx_nucleo.c**
- ADCx_MspInit() : **stm32f4xx_nucleo.c**

- b -

- BSP_GetVersion() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_JOY_DeInit() : **stm32f4xx_nucleo.h** , **stm32f4xx_nucleo.c**
- BSP_JOY_GetState() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_JOY_Init() : **stm32f4xx_nucleo.h** , **stm32f4xx_nucleo.c**
- BSP_LED_DeInit() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_LED_Init() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_LED_Off() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_LED_On() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_LED_Toggle() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_PB_DeInit() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_PB_GetState() : **stm32f4xx_nucleo.c** , **stm32f4xx_nucleo.h**
- BSP_PB_Init() : **stm32f4xx_nucleo.h** , **stm32f4xx_nucleo.c**

- l -

- LCD_Delay() : **stm32f4xx_nucleo.c**
- LCD_IO_Init() : **stm32f4xx_nucleo.c**
- LCD_IO_WriteData() : **stm32f4xx_nucleo.c**
- LCD_IO_WriteMultipleData() : **stm32f4xx_nucleo.c**
- LCD_IO_WriteReg() : **stm32f4xx_nucleo.c**

- s -

- SD_IO_CSState() : **stm32f4xx_nucleo.c**
- SD_IO_Init() : **stm32f4xx_nucleo.c**
- SD_IO_WriteByte() : **stm32f4xx_nucleo.c**
- SD_IO_WriteReadData() : **stm32f4xx_nucleo.c**
- SPIx_Error() : **stm32f4xx_nucleo.c**
- SPIx_Init() : **stm32f4xx_nucleo.c**
- SPIx_MspInit() : **stm32f4xx_nucleo.c**
- SPIx_Write() : **stm32f4xx_nucleo.c**
- SPIx_WriteReadData() : **stm32f4xx_nucleo.c**

STM32F4xx-Nucleo BSP User Manual

- BUTTON_IRQn : **stm32f4xx_nucleo.c**
- BUTTON_PIN : **stm32f4xx_nucleo.c**
- BUTTON_PORT : **stm32f4xx_nucleo.c**
- GPIO_PIN : **stm32f4xx_nucleo.c**
- GPIO_PORT : **stm32f4xx_nucleo.c**
- hnucleo_Adc : **stm32f4xx_nucleo.c**
- hnucleo_Spi : **stm32f4xx_nucleo.c**
- sConfig : **stm32f4xx_nucleo.c**
- SpixTimeout : **stm32f4xx_nucleo.c**

Generated on Fri Jan 20 2017 15:36:09 for STM32F4xx-Nucleo BSP User Manual by doxygen 1.7.6.1

STM32F4xx-Nucleo BSP User Manual

- Button_TypeDef : **stm32f4xx_nucleo.h**
- ButtonMode_TypeDef : **stm32f4xx_nucleo.h**
- JOYState_TypeDef : **stm32f4xx_nucleo.h**
- Led_TypeDef : **stm32f4xx_nucleo.h**

STM32F4xx-Nucleo BSP User Manual

- BUTTON_KEY : **stm32f4xx_nucleo.h**
- BUTTON_MODE_EXTI : **stm32f4xx_nucleo.h**
- BUTTON_MODE_GPIO : **stm32f4xx_nucleo.h**
- BUTTON_USER : **stm32f4xx_nucleo.h**
- JOY_DOWN : **stm32f4xx_nucleo.h**
- JOY_LEFT : **stm32f4xx_nucleo.h**
- JOY_NONE : **stm32f4xx_nucleo.h**
- JOY_RIGHT : **stm32f4xx_nucleo.h**
- JOY_SEL : **stm32f4xx_nucleo.h**
- JOY_UP : **stm32f4xx_nucleo.h**
- LED2 : **stm32f4xx_nucleo.h**

STM32F4xx-Nucleo BSP User Manual

- _ -

- __STM32F4xx_NUCLEO_BSP_VERSION : **stm32f4xx_nucleo.c**
- __STM32F4xx_NUCLEO_BSP_VERSION_MAIN : **stm32f4xx_nucleo.c**
- __STM32F4xx_NUCLEO_BSP_VERSION_RC : **stm32f4xx_nucleo.c**
- __STM32F4xx_NUCLEO_BSP_VERSION_SUB1 : **stm32f4xx_nucleo.c**
- __STM32F4xx_NUCLEO_BSP_VERSION_SUB2 : **stm32f4xx_nucleo.c**

- a -

- ADAFRUIT_TFT_JOY_SD_ID802 : **stm32f4xx_nucleo.h**

- b -

- BUTTONn : **stm32f4xx_nucleo.h**
- BUTTONx_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- BUTTONx_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**

- k -

- KEY_BUTTON_EXTI_IRQn : **stm32f4xx_nucleo.h**
- KEY_BUTTON_EXTI_LINE : **stm32f4xx_nucleo.h**

- KEY_BUTTON_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- KEY_BUTTON_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- KEY_BUTTON_GPIO_PORT : **stm32f4xx_nucleo.h**
- KEY_BUTTON_PIN : **stm32f4xx_nucleo.h**

- l -

- LCD_CS_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- LCD_CS_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- LCD_CS_GPIO_PORT : **stm32f4xx_nucleo.h**
- LCD_CS_HIGH : **stm32f4xx_nucleo.h**
- LCD_CS_LOW : **stm32f4xx_nucleo.h**
- LCD_CS_PIN : **stm32f4xx_nucleo.h**
- LCD_DC_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- LCD_DC_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- LCD_DC_GPIO_PORT : **stm32f4xx_nucleo.h**
- LCD_DC_HIGH : **stm32f4xx_nucleo.h**
- LCD_DC_LOW : **stm32f4xx_nucleo.h**
- LCD_DC_PIN : **stm32f4xx_nucleo.h**
- LED2_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- LED2_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- LED2_GPIO_PORT : **stm32f4xx_nucleo.h**
- LED2_PIN : **stm32f4xx_nucleo.h**
- LEDn : **stm32f4xx_nucleo.h**
- LEDx_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- LEDx_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**

- n -

- NUCLEO_ADCx : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_CHANNEL : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_CLK_DISABLE : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_GPIO_PIN : **stm32f4xx_nucleo.h**
- NUCLEO_ADCx_GPIO_PORT : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx : **stm32f4xx_nucleo.h**

- NUCLEO_SPIx_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_MOSI_AF : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_MOSI_GPIO_PORT : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MISO_PIN : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_MOSI_PIN : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_AF : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_GPIO_PORT : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_SCK_PIN : **stm32f4xx_nucleo.h**
- NUCLEO_SPIx_TIMEOUT_MAX : **stm32f4xx_nucleo.h**

- s -

- SD_CS_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- SD_CS_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- SD_CS_GPIO_PORT : **stm32f4xx_nucleo.h**
- SD_CS_HIGH : **stm32f4xx_nucleo.h**
- SD_CS_LOW : **stm32f4xx_nucleo.h**
- SD_CS_PIN : **stm32f4xx_nucleo.h**
- SD_DUMMY_BYTE : **stm32f4xx_nucleo.c**
- SD_NO_RESPONSE_EXPECTED : **stm32f4xx_nucleo.c**

- u -

- USER_BUTTON_EXTI_IRQn : **stm32f4xx_nucleo.h**
- USER_BUTTON_EXTI_LINE : **stm32f4xx_nucleo.h**
- USER_BUTTON_GPIO_CLK_DISABLE : **stm32f4xx_nucleo.h**
- USER_BUTTON_GPIO_CLK_ENABLE : **stm32f4xx_nucleo.h**
- USER_BUTTON_GPIO_PORT : **stm32f4xx_nucleo.h**
- USER_BUTTON_PIN : **stm32f4xx_nucleo.h**

Generated on Fri Jan 20 2017 15:36:09 for STM32F4xx-Nucleo BSP
User Manual by  1.7.6.1

STM32F4xx-Nucleo BSP User Manual

stm32f4xx_nucleo.c File Reference

This file provides set of firmware functions to manage: More...

```
#include "stm32f4xx_nucleo.h"
```

Go to the source code of this file.

Defines

#define	**__STM32F4xx_NUCLEO_BSP_VERSION_MAIN** (0x01) STM32F4xx NUCLEO BSP Driver version number V1.2.6.
#define	**__STM32F4xx_NUCLEO_BSP_VERSION_SUB1** (0x02)
#define	**__STM32F4xx_NUCLEO_BSP_VERSION_SUB2** (0x06)
#define	**__STM32F4xx_NUCLEO_BSP_VERSION_RC** (0x00)
#define	**__STM32F4xx_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_WriteReadData** (const uint8_t *DataIn, uint8_t *DataOut, uint16_t DataLegnth) 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) Writes 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 STM32F4xx 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.

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**] = {**LED2_GPIO_PORT**}
const uint16_t	**GPIO_PIN** [**LEDn**] = {**LED2_PIN**}
GPIO_TypeDef *	**BUTTON_PORT** [**BUTTONn**] = {**KEY_BUTTON_GPIO_PORT**}
const uint16_t	**BUTTON_PIN** [**BUTTONn**] = {**KEY_BUTTON_PIN**}
const uint8_t	**BUTTON_IRQn** [**BUTTONn**] = {**KEY_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:
V1.2.6

Date:
31-January-2017

- LEDs and push-button available on STM32F4XX-Nucleo Kit from STMicroelectronics
- LCD, joystick and microSD available on Adafruit 1.8" TFT LCD shield (reference ID 802)

Attention:

Definition in file **stm32f4xx_nucleo.c**.

STM32F4xx-Nucleo BSP User Manual

Defines | Enumerations | Functions

stm32f4xx_nucleo.h File Reference

This file contains definitions for: More...

```
#include "stm32f4xx_hal.h"
```

Go to the source code of this file.

Defines

#define	**ADAFRUIT_TFT_JOY_SD_ID802**
#define	**LEDn** 1
#define	**LED2_PIN** GPIO_PIN_5
#define	**LED2_GPIO_PORT** GPIOA
#define	**LED2_GPIO_CLK_ENABLE**() __HAL_RCC_GPIOA_CLK_E
#define	**LED2_GPIO_CLK_DISABLE**() __HAL_RCC_GPIOA_CLK_D
#define	**LEDx_GPIO_CLK_ENABLE**(__INDEX__) **LED2_GPIO_CLK**
#define	**LEDx_GPIO_CLK_DISABLE**(__INDEX__) **LED2_GPIO_CLK**
#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_GPIO
#define	**USER_BUTTON_GPIO_CLK_DISABLE**() __HAL_RCC_GPI
#define	**USER_BUTTON_EXTI_LINE** GPIO_PIN_13
#define	**USER_BUTTON_EXTI_IRQn** EXTI15_10_IRQn
#define	**BUTTONx_GPIO_CLK_ENABLE**(__INDEX__) **USER_BUTT**
#define	**BUTTONx_GPIO_CLK_DISABLE**(__INDEX__) **USER_BUTT**
#define	**KEY_BUTTON_PIN** **USER_BUTTON_PIN**
#define	**KEY_BUTTON_GPIO_PORT** **USER_BUTTON_GPIO_PORT**
#define	**KEY_BUTTON_GPIO_CLK_ENABLE**() **USER_BUTTON_GP**
#define	**KEY_BUTTON_GPIO_CLK_DISABLE**() **USER_BUTTON_G**
#define	**KEY_BUTTON_EXTI_LINE** **USER_BUTTON_EXTI_LINE**
#define	**KEY_BUTTON_EXTI_IRQn** **USER_BUTTON_EXTI_IRQn**
#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	**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_POR** GPIO_PIN_SET)
#define	**LCD_DC_LOW**() HAL_GPIO_WritePin(**LCD_DC_GPIO_POR** GPIO_PIN_RESET)
#define	**LCD_DC_HIGH**() HAL_GPIO_WritePin(**LCD_DC_GPIO_POR** GPIO_PIN_SET)
#define	**SD_CS_PIN** GPIO_PIN_5 SD Control Interface pins (shield D4)
#define	**SD_CS_GPIO_PORT** GPIOB
#define	**SD_CS_GPIO_CLK_ENABLE**() __HAL_RCC_GPIOB_CLK_
#define	**SD_CS_GPIO_CLK_DISABLE**() __HAL_RCC_GPIOB_CLK_
#define	**LCD_CS_PIN** GPIO_PIN_6 LCD Control Interface pins (shield D10)
#define	**LCD_CS_GPIO_PORT** GPIOB
#define	**LCD_CS_GPIO_CLK_ENABLE**() __HAL_RCC_GPIOB_CLK
#define	**LCD_CS_GPIO_CLK_DISABLE**() __HAL_RCC_GPIOB_CLK
#define	**LCD_DC_PIN** GPIO_PIN_9 LCD Data/Command Interface pins (shield D8)
#define	**LCD_DC_GPIO_PORT** GPIOA
#define	**LCD_DC_GPIO_CLK_ENABLE**() __HAL_RCC_GPIOA_CLK
#define	**LCD_DC_GPIO_CLK_DISABLE**() __HAL_RCC_GPIOA_CLK

#define **NUCLEO_ADCx** ADC1

ADC Interface pins used to detect motion of Joystick available

#define **NUCLEO_ADCx_CLK_ENABLE**() __HAL_RCC_ADC1_CLK

#define **NUCLEO_ADCx_CLK_DISABLE**() __HAL_RCC_ADC1_CLK

#define **NUCLEO_ADCx_CHANNEL** ADC_CHANNEL_8

#define **NUCLEO_ADCx_GPIO_PORT** GPIOB

#define **NUCLEO_ADCx_GPIO_PIN** GPIO_PIN_0

#define **NUCLEO_ADCx_GPIO_CLK_ENABLE**() __HAL_RCC_GPIO

#define **NUCLEO_ADCx_GPIO_CLK_DISABLE**() __HAL_RCC_GPI

Enumerations

enum	**Led_TypeDef** { **LED2** = 0 }
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 STM32F4xx 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:
V1.2.6

Date:
31-January-2017

- LEDs and push-button available on STM32F4XX-Nucleo Kit from STMicroelectronics
- LCD, joystick and microSD available on Adafruit 1.8" TFT LCD shield (reference ID 802)

Attention:

Definition in file **stm32f4xx_nucleo.h**.

Define Documentation

#define ADAFRUIT_TFT_JOY_SD_ID802

Definition at line **55** of file **stm32f4xx_nucleo.h**.

STM32F4xx-Nucleo BSP User Manual

Modules

Here is a list of all modules:

Generated on Fri Jan 20 2017 15:36:09 for STM32F4xx-Nucleo BSP
User Manual by  1.7.6.1

STM32F4xx-Nucleo BSP User Manual

File List

Here is a list of all files with brief descriptions:

stm32f4xx_nucleo.c [code]	This file provides set of firmware functions to manage:
stm32f4xx_nucleo.h [code]	This file contains definitions for:

STM32F4xx-Nucleo BSP User Manual

Directories

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

- Drivers
  - BSP
    - STM32F4xx-Nucleo

Generated on Fri Jan 20 2017 15:36:09 for STM32F4xx-Nucleo BSP User Manual by doxygen 1.7.6.1

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL

STM32F4XX NUCLEO

This file provides set of firmware functions to manage Leds and push-button available on STM32F4xx-Nucleo Kit from STMicroelectronics. More...

Modules

Detailed Description

This file provides set of firmware functions to manage Leds and push-button available on STM32F4xx-Nucleo Kit from STMicroelectronics.

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Private Defines

STM32F4XX NUCLEO LOW LEVEL

Defines

#define	**__STM32F4xx_NUCLEO_BSP_VERSION_MAIN** (0x01) STM32F4xx NUCLEO BSP Driver version number V1.2.6.
#define	**__STM32F4xx_NUCLEO_BSP_VERSION_SUB1** (0x02)
#define	**__STM32F4xx_NUCLEO_BSP_VERSION_SUB2** (0x06)
#define	**__STM32F4xx_NUCLEO_BSP_VERSION_RC** (0x00)
#define	**__STM32F4xx_NUCLEO_BSP_VERSION**
#define	**SD_DUMMY_BYTE** 0xFF LINK SD Card.
#define	**SD_NO_RESPONSE_EXPECTED** 0x80

Define Documentation

#define __STM32F4xx_NUCLEO_BSP_VERSION

Value:

```
((__STM32F4xx_NUCLEO_BSP_VERSION_MAIN << 24)\
                                                |(__
STM32F4xx_NUCLEO_BSP_VERSION_SUB1 << 16)\
                                                |(__
STM32F4xx_NUCLEO_BSP_VERSION_SUB2 << 8 )\
                                                |(__
STM32F4xx_NUCLEO_BSP_VERSION_RC))
```

Definition at line **78** of file **stm32f4xx_nucleo.c**.

Referenced by **BSP_GetVersion()**.

#define __STM32F4xx_NUCLEO_BSP_VERSION_MAIN (0x01)

STM32F4xx NUCLEO BSP Driver version number V1.2.6.

[31:24] main version

Definition at line **74** of file **stm32f4xx_nucleo.c**.

#define __STM32F4xx_NUCLEO_BSP_VERSION_RC (0x00)

[7:0] release candidate

Definition at line **77** of file **stm32f4xx_nucleo.c**.

#define __STM32F4xx_NUCLEO_BSP_VERSION_SUB1 (0x02)

[23:16] sub1 version

Definition at line **75** of file **stm32f4xx_nucleo.c**.

#define __STM32F4xx_NUCLEO_BSP_VERSION_SUB2 (0x06)

[15:8] sub2 version

Definition at line **76** of file **stm32f4xx_nucleo.c**.

#define SD_DUMMY_BYTE 0xFF

LINK SD Card.

Definition at line **86** of file **stm32f4xx_nucleo.c**.

Referenced by **SD_IO_Init()**.

#define SD_NO_RESPONSE_EXPECTED 0x80

Definition at line **87** of file **stm32f4xx_nucleo.c**.

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Private FunctionPrototypes

STM32F4XX NUCLEO LOW LEVEL

Functions

static void	**SPIx_Init** (void) Initializes SPI HAL.
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	**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_MspInit** (ADC_HandleTypeDef *hadc) Initializes ADC MSP.

Function Documentation

static void ADCx_DeInit (void) [static]

Initializes ADC HAL.

Definition at line 741 of file stm32f4xx_nucleo.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 717 of file stm32f4xx_nucleo.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 675 of file stm32f4xx_nucleo.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 538 of file stm32f4xx_nucleo.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 463 of file stm32f4xx_nucleo.c.

References 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 445 of file stm32f4xx_nucleo.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 **372** of file **stm32f4xx_nucleo.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 **337** of file **stm32f4xx_nucleo.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()**.

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Private Functions

STM32F4XX NUCLEO LOW LEVEL

Functions

uint32_t	**BSP_GetVersion** (void) This method returns the STM32F4xx 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.
void	**BSP_JOY_DeInit** (void) DeInit joystick GPIOs.
JOYState_TypeDef	**BSP_JOY_GetState** (void) Returns the Joystick key pressed.
static void	**SPIx_WriteReadData** (const uint8_t *DataIn, uint8_t *DataOut, uint16_t DataLegnth) SPI Write a byte to device.

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) Writes 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 **698** of file **stm32f4xx_nucleo.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 STM32F4xx NUCLEO BSP Driver revision.

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

Definition at line **179** of file **stm32f4xx_nucleo.c**.

References **__STM32F4xx_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 modee.

Definition at line 776 of file stm32f4xx_nucleo.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 793 of file stm32f4xx_nucleo.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 756 of file stm32f4xx_nucleo.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:

- LED2

Note:

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

Definition at line **215** of file **stm32f4xx_nucleo.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:

- LED2

Definition at line **190** of file **stm32f4xx_nucleo.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:

- LED2

Definition at line **243** of file **stm32f4xx_nucleo.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 **232** of file **stm32f4xx_nucleo.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:

- LED2

Definition at line **254** of file **stm32f4xx_nucleo.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_KEY

Note:

PB DeInit does not disable the GPIO clock

Definition at line **306** of file **stm32f4xx_nucleo.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_KEY

Return values:

The Button GPIO pin value.

Definition at line **321** of file **stm32f4xx_nucleo.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_KEY
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 **269** of file **stm32f4xx_nucleo.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 **663** of file **stm32f4xx_nucleo.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 **588** of file **stm32f4xx_nucleo.c**.

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

void LCD_IO_WriteMultipleData (uint8_t * pData, uint32_t Size)

Writes register value.

Parameters:

pData,: Pointer on the register value

Size,: Size of byte to transmit to the register

Definition at line **608** of file **stm32f4xx_nucleo.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 **568** of file **stm32f4xx_nucleo.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 **498** of file **stm32f4xx_nucleo.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 **526** of file **stm32f4xx_nucleo.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,: value to be written

DataOut,: value to be read

DataLength,: length of data

Definition at line **516** of file **stm32f4xx_nucleo.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 **427** of file **stm32f4xx_nucleo.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 be read
DataLegnth,: length of data

Definition at line **409** of file **stm32f4xx_nucleo.c**.

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

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

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Exported Functions

STM32F4XX NUCLEO LOW LEVEL

Functions

uint32_t	**BSP_GetVersion** (void) This method returns the STM32F4xx 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 STM32F4xx NUCLEO BSP Driver revision.

Return values:

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

Definition at line 179 of file stm32f4xx_nucleo.c.

References __STM32F4xx_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 modee.

Definition at line 776 of file stm32f4xx_nucleo.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 **793** of file **stm32f4xx_nucleo.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 **756** of file **stm32f4xx_nucleo.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:

- LED2

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

Definition at line **215** of file **stm32f4xx_nucleo.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:
- LED2

Definition at line **190** of file **stm32f4xx_nucleo.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:
- LED2

Definition at line **243** of file **stm32f4xx_nucleo.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 **232** of file **stm32f4xx_nucleo.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:

- LED2

Definition at line **254** of file **stm32f4xx_nucleo.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_KEY

Note:

PB DeInit does not disable the GPIO clock

Definition at line **306** of file **stm32f4xx_nucleo.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_KEY

Return values:

The Button GPIO pin value.

Definition at line **321** of file **stm32f4xx_nucleo.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_KEY

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 **269** of file **stm32f4xx_nucleo.c**.

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

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Private Variables

STM32F4XX NUCLEO LOW LEVEL

Variables

GPIO_TypeDef *	**GPIO_PORT** [**LEDn**] = {**LED2_GPIO_PORT**}
const uint16_t	**GPIO_PIN** [**LEDn**] = {**LED2_PIN**}
GPIO_TypeDef *	**BUTTON_PORT** [**BUTTONn**] = {**KEY_BUTTON_GPIO_PORT**}
const uint16_t	**BUTTON_PIN** [**BUTTONn**] = {**KEY_BUTTON_PIN**}
const uint8_t	**BUTTON_IRQn** [**BUTTONn**] = {**KEY_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] = {KEY_BUTTON_EXTI_IR

Definition at line 109 of file stm32f4xx_nucleo.c.

Referenced by BSP_PB_DeInit(), and BSP_PB_Init().

const uint16_t BUTTON_PIN[BUTTONn] = {KEY_BUTTON_PIN}

Definition at line 108 of file stm32f4xx_nucleo.c.

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

GPIO_TypeDef* BUTTON_PORT[BUTTONn] = {KEY_BUTTON_GPIC

Definition at line 107 of file stm32f4xx_nucleo.c.

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

const uint16_t GPIO_PIN[LEDn] = {LED2_PIN}

Definition at line 105 of file stm32f4xx_nucleo.c.

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

GPIO_TypeDef* GPIO_PORT[LEDn] = {LED2_GPIO_PORT}

Definition at line 103 of file stm32f4xx_nucleo.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 **122** of file **stm32f4xx_nucleo.c**.

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

SPI_HandleTypeDef hnucleo_Spi [static]

Definition at line **118** of file **stm32f4xx_nucleo.c**.

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

ADC_ChannelConfTypeDef sConfig [static]

Definition at line **124** of file **stm32f4xx_nucleo.c**.

Referenced by **BSP_JOY_Init()**.

uint32_t SpixTimeout = NUCLEO_SPIx_TIMEOUT_MAX

BUS variables.

Definition at line **117** of file **stm32f4xx_nucleo.c**.

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

User Manual by  1.7.6.1

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Exported Types

STM32F4XX NUCLEO LOW LEVEL

Enumerations

enum	**Led_TypeDef** { **LED2** = 0 }
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 77 of file stm32f4xx_nucleo.h.

enum ButtonMode_TypeDef

Enumerator:

BUTTON_MODE_GPIO

BUTTON_MODE_EXTI

Definition at line 84 of file stm32f4xx_nucleo.h.

enum JOYState_TypeDef

Enumerator:

JOY_NONE

JOY_SEL

JOY_DOWN

JOY_LEFT

JOY_RIGHT

JOY_UP

Definition at line 90 of file stm32f4xx_nucleo.h.

enum Led_TypeDef

Enumerator:

LED2

Definition at line **72** of file **stm32f4xx_nucleo.h**.

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL BUTTON

STM32F4XX NUCLEO 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_GPIC
#define	**USER_BUTTON_GPIO_CLK_DISABLE**() __HAL_RCC_GPI
#define	**USER_BUTTON_EXTI_LINE** GPIO_PIN_13
#define	**USER_BUTTON_EXTI_IRQn** EXTI15_10_IRQn
#define	**BUTTONx_GPIO_CLK_ENABLE**(__INDEX__) **USER_BUTT**
#define	**BUTTONx_GPIO_CLK_DISABLE**(__INDEX__) **USER_BUT**
#define	**KEY_BUTTON_PIN** **USER_BUTTON_PIN**
#define	**KEY_BUTTON_GPIO_PORT** **USER_BUTTON_GPIO_PORT**
#define	**KEY_BUTTON_GPIO_CLK_ENABLE**() **USER_BUTTON_GP**
#define	**KEY_BUTTON_GPIO_CLK_DISABLE**() **USER_BUTTON_G**
#define	**KEY_BUTTON_EXTI_LINE** **USER_BUTTON_EXTI_LINE**
#define	**KEY_BUTTON_EXTI_IRQn** **USER_BUTTON_EXTI_IRQn**

Define Documentation

#define BUTTONn 1

Definition at line 134 of file stm32f4xx_nucleo.h.

#define BUTTONx_GPIO_CLK_DISABLE (__INDEX__) USER_BU

Definition at line 147 of file stm32f4xx_nucleo.h.

#define BUTTONx_GPIO_CLK_ENABLE (__INDEX__) USER_BU

Definition at line 146 of file stm32f4xx_nucleo.h.

Referenced by BSP_PB_Init().

#define KEY_BUTTON_EXTI_IRQn USER_BUTTON_EXTI_IRQn

Definition at line 155 of file stm32f4xx_nucleo.h.

#define KEY_BUTTON_EXTI_LINE USER_BUTTON_EXTI_LINE

Definition at line 154 of file stm32f4xx_nucleo.h.

#define KEY_BUTTON_GPIO_CLK_DISABLE ( ) USER_BUTTON_

Definition at line 153 of file stm32f4xx_nucleo.h.

#define KEY_BUTTON_GPIO_CLK_ENABLE ( ) USER_BUTTON_C

Definition at line **152** of file **stm32f4xx_nucleo.h**.

#define KEY_BUTTON_GPIO_PORT USER_BUTTON_GPIO_PORT

Definition at line **151** of file **stm32f4xx_nucleo.h**.

#define KEY_BUTTON_PIN USER_BUTTON_PIN

Definition at line **150** of file **stm32f4xx_nucleo.h**.

#define USER_BUTTON_EXTI_IRQn EXTI15_10_IRQn

Definition at line **144** of file **stm32f4xx_nucleo.h**.

#define USER_BUTTON_EXTI_LINE GPIO_PIN_13

Definition at line **143** of file **stm32f4xx_nucleo.h**.

#define USER_BUTTON_GPIO_CLK_DISABLE ( ) __HAL_RCC_GI

Definition at line **142** of file **stm32f4xx_nucleo.h**.

#define USER_BUTTON_GPIO_CLK_ENABLE ( ) __HAL_RCC_GP

Definition at line **141** of file **stm32f4xx_nucleo.h**.

#define USER_BUTTON_GPIO_PORT GPIOC

Definition at line **140** of file **stm32f4xx_nucleo.h**.

#define USER_BUTTON_PIN GPIO_PIN_13

Key push-button.

Definition at line **139** of file **stm32f4xx_nucleo.h**.

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL BUS

STM32F4XX NUCLEO 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	**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_POR** GPIO_PIN_SET)
#define	**LCD_DC_LOW**() HAL_GPIO_WritePin(**LCD_DC_GPIO_POR** GPIO_PIN_RESET)
#define	**LCD_DC_HIGH**() HAL_GPIO_WritePin(**LCD_DC_GPIO_POR** GPIO_PIN_SET)
#define	**SD_CS_PIN** GPIO_PIN_5 SD Control Interface pins (shield D4)
#define	**SD_CS_GPIO_PORT** GPIOB
#define	**SD_CS_GPIO_CLK_ENABLE**() __HAL_RCC_GPIOB_CLK_
#define	**SD_CS_GPIO_CLK_DISABLE**() __HAL_RCC_GPIOB_CLK_

#define **LCD_CS_PIN** GPIO_PIN_6
LCD Control Interface pins (shield D10)

#define **LCD_CS_GPIO_PORT** GPIOB

#define **LCD_CS_GPIO_CLK_ENABLE**() __HAL_RCC_GPIOB_CLK

#define **LCD_CS_GPIO_CLK_DISABLE**() __HAL_RCC_GPIOB_CLK

#define **LCD_DC_PIN** GPIO_PIN_9
LCD Data/Command Interface pins (shield D8)

#define **LCD_DC_GPIO_PORT** GPIOA

#define **LCD_DC_GPIO_CLK_ENABLE**() __HAL_RCC_GPIOA_CLK

#define **LCD_DC_GPIO_CLK_DISABLE**() __HAL_RCC_GPIOA_CLK

#define **NUCLEO_ADCx** ADC1
ADC Interface pins used to detect motion of Joystick available

#define **NUCLEO_ADCx_CLK_ENABLE**() __HAL_RCC_ADC1_CLK

#define **NUCLEO_ADCx_CLK_DISABLE**() __HAL_RCC_ADC1_CLK

#define **NUCLEO_ADCx_CHANNEL** ADC_CHANNEL_8

#define **NUCLEO_ADCx_GPIO_PORT** GPIOB

#define **NUCLEO_ADCx_GPIO_PIN** GPIO_PIN_0

#define **NUCLEO_ADCx_GPIO_CLK_ENABLE**() __HAL_RCC_GPIO

#define **NUCLEO_ADCx_GPIO_CLK_DISABLE**() __HAL_RCC_GPI

Define Documentation

#define LCD_CS_GPIO_CLK_DISABLE ( ) __HAL_RCC_GPIOB_C

Definition at line 217 of file stm32f4xx_nucleo.h.

#define LCD_CS_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIOB_CL

Definition at line 216 of file stm32f4xx_nucleo.h.

Referenced by LCD_IO_Init().

#define LCD_CS_GPIO_PORT GPIOB

Definition at line 215 of file stm32f4xx_nucleo.h.

Referenced by LCD_IO_Init().

#define LCD_CS_HIGH ( ) HAL_GPIO_WritePin(LCD_CS_GPIO_P

Definition at line 199 of file stm32f4xx_nucleo.h.

Referenced by LCD_IO_Init(), LCD_IO_WriteData(), LCD_IO_WriteMultipleData(), and LCD_IO_WriteReg().

#define LCD_CS_LOW ( ) HAL_GPIO_WritePin(LCD_CS_GPIO_P

LCD Control Lines management.

Definition at line 198 of file stm32f4xx_nucleo.h.

Referenced by LCD_IO_WriteData(), LCD_IO_WriteMultipleData(),

and **LCD_IO_WriteReg()**.

#define LCD_CS_PIN GPIO_PIN_6

LCD Control Interface pins (shield D10)

Definition at line **214** of file **stm32f4xx_nucleo.h**.

Referenced by **LCD_IO_Init()**.

#define LCD_DC_GPIO_CLK_DISABLE ( ) __HAL_RCC_GPIOA_C

Definition at line **225** of file **stm32f4xx_nucleo.h**.

#define LCD_DC_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIOA_CI

Definition at line **224** of file **stm32f4xx_nucleo.h**.

Referenced by **LCD_IO_Init()**.

#define LCD_DC_GPIO_PORT GPIOA

Definition at line **223** of file **stm32f4xx_nucleo.h**.

Referenced by **LCD_IO_Init()**.

#define LCD_DC_HIGH ( ) HAL_GPIO_WritePin(LCD_DC_GPIO_P

Definition at line **201** of file **stm32f4xx_nucleo.h**.

Referenced by **LCD_IO_WriteData()**, and **LCD_IO_WriteMultipleData()**.

#define LCD_DC_LOW ( ) HAL_GPIO_WritePin(LCD_DC_GPIO_P(

Definition at line 200 of file stm32f4xx_nucleo.h.

Referenced by LCD_IO_WriteReg().

#define LCD_DC_PIN GPIO_PIN_9

LCD Data/Command Interface pins (shield D8)

Definition at line 222 of file stm32f4xx_nucleo.h.

Referenced by LCD_IO_Init().

#define NUCLEO_ADCx ADC1

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

Definition at line 237 of file stm32f4xx_nucleo.h.

Referenced by ADCx_DeInit(), and ADCx_Init().

#define NUCLEO_ADCx_CHANNEL ADC_CHANNEL_8

Definition at line 241 of file stm32f4xx_nucleo.h.

Referenced by BSP_JOY_Init().

#define NUCLEO_ADCx_CLK_DISABLE ( ) __HAL_RCC_ADC1_C

Definition at line 239 of file stm32f4xx_nucleo.h.

Referenced by **ADCx_MspDeInit()**.

#define NUCLEO_ADCx_CLK_ENABLE ( ) __HAL_RCC_ADC1_CL

Definition at line **238** of file **stm32f4xx_nucleo.h**.

Referenced by **ADCx_MspInit()**.

#define NUCLEO_ADCx_GPIO_CLK_DISABLE ( ) __HAL_RCC_GI

Definition at line **246** of file **stm32f4xx_nucleo.h**.

#define NUCLEO_ADCx_GPIO_CLK_ENABLE ( ) __HAL_RCC_GP

Definition at line **245** of file **stm32f4xx_nucleo.h**.

Referenced by **ADCx_MspInit()**.

#define NUCLEO_ADCx_GPIO_PIN GPIO_PIN_0

Definition at line **244** of file **stm32f4xx_nucleo.h**.

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

#define NUCLEO_ADCx_GPIO_PORT GPIOB

Definition at line **243** of file **stm32f4xx_nucleo.h**.

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

#define NUCLEO_SPIx SPI1

Definition at line **167** of file **stm32f4xx_nucleo.h**.

Referenced by **SPIx_Init()**.

#define NUCLEO_SPIx_CLK_ENABLE ( ) __HAL_RCC_SPI1_CLK

Definition at line **168** of file **stm32f4xx_nucleo.h**.

Referenced by **SPIx_MspInit()**.

#define NUCLEO_SPIx_MISO_MOSI_AF GPIO_AF5_SPI1

Definition at line **176** of file **stm32f4xx_nucleo.h**.

Referenced by **SPIx_MspInit()**.

#define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISABLE ( ) __H

Definition at line **179** of file **stm32f4xx_nucleo.h**.

#define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE ( ) __H

Definition at line **178** of file **stm32f4xx_nucleo.h**.

Referenced by **SPIx_MspInit()**.

#define NUCLEO_SPIx_MISO_MOSI_GPIO_PORT GPIOA

Definition at line **177** of file **stm32f4xx_nucleo.h**.

Referenced by **SPIx_MspInit()**.

#define NUCLEO_SPIx_MISO_PIN GPIO_PIN_6

Definition at line 180 of file stm32f4xx_nucleo.h.

Referenced by SPIx_MspInit().

#define NUCLEO_SPIx_MOSI_PIN GPIO_PIN_7

Definition at line 181 of file stm32f4xx_nucleo.h.

Referenced by SPIx_MspInit().

#define NUCLEO_SPIx_SCK_AF GPIO_AF5_SPI1

Definition at line 170 of file stm32f4xx_nucleo.h.

Referenced by SPIx_MspInit().

#define NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE ( ) __HAL_RC

Definition at line 174 of file stm32f4xx_nucleo.h.

#define NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE ( ) __HAL_RC

Definition at line 173 of file stm32f4xx_nucleo.h.

Referenced by SPIx_MspInit().

#define NUCLEO_SPIx_SCK_GPIO_PORT GPIOA

Definition at line 171 of file stm32f4xx_nucleo.h.

Referenced by SPIx_MspInit().

#define NUCLEO_SPIx_SCK_PIN GPIO_PIN_5

Definition at line 172 of file stm32f4xx_nucleo.h.

Referenced by SPIx_MspInit().

#define NUCLEO_SPIx_TIMEOUT_MAX 1000

Definition at line 187 of file stm32f4xx_nucleo.h.

#define SD_CS_GPIO_CLK_DISABLE ( ) __HAL_RCC_GPIOB_CL

Definition at line 209 of file stm32f4xx_nucleo.h.

#define SD_CS_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIOB_CLK

Definition at line 208 of file stm32f4xx_nucleo.h.

Referenced by SD_IO_Init().

#define SD_CS_GPIO_PORT GPIOB

Definition at line 207 of file stm32f4xx_nucleo.h.

Referenced by SD_IO_Init().

#define SD_CS_HIGH ( ) HAL_GPIO_WritePin(SD_CS_GPIO_POR

Definition at line 193 of file stm32f4xx_nucleo.h.

Referenced by **SD_IO_CSState()**, and **SD_IO_Init()**.

#define SD_CS_LOW ( ) HAL_GPIO_WritePin(SD_CS_GPIO_POR

SD Control Lines management.

Definition at line **192** of file **stm32f4xx_nucleo.h**.

Referenced by **SD_IO_CSState()**.

#define SD_CS_PIN GPIO_PIN_5

SD Control Interface pins (shield D4)

Definition at line **206** of file **stm32f4xx_nucleo.h**.

Referenced by **SD_IO_Init()**.

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL LED

STM32F4XX NUCLEO LOW LEVEL Exported Constants

Define for STM32F4XX_NUCLEO board. More...

Defines

#define **LEDn** 1
#define **LED2_PIN** GPIO_PIN_5
#define **LED2_GPIO_PORT** GPIOA
#define **LED2_GPIO_CLK_ENABLE**() __HAL_RCC_GPIOA_CLK_E
#define **LED2_GPIO_CLK_DISABLE**() __HAL_RCC_GPIOA_CLK_D
#define **LEDx_GPIO_CLK_ENABLE**(__INDEX__) **LED2_GPIO_CLK**
#define **LEDx_GPIO_CLK_DISABLE**(__INDEX__) **LED2_GPIO_CLK**

Detailed Description

Define for STM32F4XX_NUCLEO board.

Define Documentation

#define LED2_GPIO_CLK_DISABLE ( ) __HAL_RCC_GPIOA_CLK

Definition at line 123 of file stm32f4xx_nucleo.h.

#define LED2_GPIO_CLK_ENABLE ( ) __HAL_RCC_GPIOA_CLK_

Definition at line 122 of file stm32f4xx_nucleo.h.

#define LED2_GPIO_PORT GPIOA

Definition at line 121 of file stm32f4xx_nucleo.h.

#define LED2_PIN GPIO_PIN_5

Definition at line 120 of file stm32f4xx_nucleo.h.

#define LEDn 1

Definition at line 118 of file stm32f4xx_nucleo.h.

#define LEDx_GPIO_CLK_DISABLE (__INDEX__) LED2_GPIO_C

Definition at line 126 of file stm32f4xx_nucleo.h.

#define LEDx_GPIO_CLK_ENABLE (__INDEX__) LED2_GPIO_C

Definition at line 125 of file stm32f4xx_nucleo.h.

Referenced by **BSP_LED_Init()**.

STM32F4xx-Nucleo BSP User Manual

Drivers Directory Reference

Directories

directory	**BSP**

STM32F4xx-Nucleo BSP User Manual

BSP Directory Reference

Directories

directory	**STM32F4xx-Nucleo**

STM32F4xx-Nucleo BSP User Manual

STM32F4xx-Nucleo Directory Reference

Files

file	**stm32f4xx_nucleo.c** [code] This file provides set of firmware functions to manage:
file	**stm32f4xx_nucleo.h** [code] This file contains definitions for:

STM32F4xx-Nucleo BSP User Manual

stm32f4xx_nucleo.h

Go to the documentation of this file.

```
00001 /**
00002   ******************************************************************************
00003   * @file    stm32f4xx_nucleo.h
00004   * @author  MCD Application Team
00005   * @version V1.2.6
00006   * @date    31-January-2017
00007   * @brief   This file contains definitions for:
00008   *          - LEDs and push-button available on STM32F4XX-Nucleo Kit
00009   *            from STMicroelectronics
00010   *          - LCD, joystick and microSD available on Adafruit 1.8" TFT LCD
00011   *            shield (reference ID 802)
00012   ******************************************************************************
00013   * @attention
00014   *
00015   * <h2><center>&copy; COPYRIGHT(c) 2017 STMicroelectronics</center></h2>
00016   *
00017   * Redistribution and use in source and binary forms, with or without modification,
```

```
00018   * are permitted provided that the followin
g conditions are met:
00019   *   1. Redistributions of source code must
 retain the above copyright notice,
00020   *      this list of conditions and the fol
lowing disclaimer.
00021   *   2. Redistributions in binary form must
 reproduce the above copyright notice,
00022   *      this list of conditions and the fol
lowing disclaimer in the documentation
00023   *      and/or other materials provided wit
h the distribution.
00024   *   3. Neither the name of STMicroelectron
ics nor the names of its contributors
00025   *      may be used to endorse or promote p
roducts derived from this software
00026   *      without specific prior written perm
ission.
00027   *
00028   * THIS SOFTWARE IS PROVIDED BY THE COPYRIG
HT HOLDERS AND CONTRIBUTORS "AS IS"
00029   * AND ANY EXPRESS OR IMPLIED WARRANTIES, I
NCLUDING, BUT NOT LIMITED TO, THE
00030   * IMPLIED WARRANTIES OF MERCHANTABILITY AN
D FITNESS FOR A PARTICULAR PURPOSE ARE
00031   * DISCLAIMED. IN NO EVENT SHALL THE COPYRI
GHT HOLDER OR CONTRIBUTORS BE LIABLE
00032   * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP
ECIAL, 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, W
HETHER IN CONTRACT, STRICT LIABILITY,
00036   * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI
SE) 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 __STM32F4XX_NUCLEO_H
00044 #define __STM32F4XX_NUCLEO_H
00045
00046 #ifdef __cplusplus
00047  extern "C" {
00048 #endif
00049
00050 /* Includes ------------------------------
-----------------------------------*/
00051 #include "stm32f4xx_hal.h"
00052
00053 /* To be defined only if the board is provid
ed with the related shield */
00054 /* https://www.adafruit.com/products/802 */
00055 #define ADAFRUIT_TFT_JOY_SD_ID802
00056
00057 /** @addtogroup BSP
00058   * @{
00059   */
00060
00061 /** @addtogroup STM32F4XX_NUCLEO
00062   * @{
00063   */
00064
00065 /** @addtogroup STM32F4XX_NUCLEO_LOW_LEVEL
00066   * @{
00067   */
00068
```

```
00069 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Exp
orted_Types STM32F4XX NUCLEO LOW LEVEL Exported Ty
pes
00070   * @{
00071   */
00072 typedef enum
00073 {
00074   LED2 = 0
00075 }Led_TypeDef;
00076
00077 typedef enum
00078 {
00079   BUTTON_USER = 0,
00080   /* Alias */
00081   BUTTON_KEY  = BUTTON_USER
00082 } Button_TypeDef;
00083
00084 typedef enum
00085 {
00086   BUTTON_MODE_GPIO = 0,
00087   BUTTON_MODE_EXTI = 1
00088 }ButtonMode_TypeDef;
00089
00090 typedef enum
00091 {
00092   JOY_NONE  = 0,
00093   JOY_SEL   = 1,
00094   JOY_DOWN  = 2,
00095   JOY_LEFT  = 3,
00096   JOY_RIGHT = 4,
00097   JOY_UP    = 5
00098 }JOYState_TypeDef;
00099
00100 /**
00101   * @}
00102   */
00103
```

```
00104 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Exp
orted_Constants STM32F4XX NUCLEO LOW LEVEL Exporte
d Constants
00105   * @{
00106   */
00107
00108 /**
00109   * @brief Define for STM32F4XX_NUCLEO board

00110   */
00111 #if !defined (USE_STM32F4XX_NUCLEO)
00112  #define USE_STM32F4XX_NUCLEO
00113 #endif
00114
00115 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_LED
 STM32F4XX NUCLEO LOW LEVEL LED
00116   * @{
00117   */
00118 #define LEDn
    1
00119
00120 #define LED2_PIN
    GPIO_PIN_5
00121 #define LED2_GPIO_PORT
    GPIOA
00122 #define LED2_GPIO_CLK_ENABLE()
    __HAL_RCC_GPIOA_CLK_ENABLE()
00123 #define LED2_GPIO_CLK_DISABLE()
    __HAL_RCC_GPIOA_CLK_DISABLE()
00124
00125 #define LEDx_GPIO_CLK_ENABLE(__INDEX__)
    LED2_GPIO_CLK_ENABLE()
00126 #define LEDx_GPIO_CLK_DISABLE(__INDEX__)
    LED2_GPIO_CLK_DISABLE()
00127 /**
00128   * @}
00129   */
```

```
00130
00131 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_BUT
TON STM32F4XX NUCLEO LOW LEVEL BUTTON
00132   * @{
00133   */
00134 #define BUTTONn
    1
00135
00136 /**
00137   * @brief Key push-button
00138   */
00139 #define USER_BUTTON_PIN
    GPIO_PIN_13
00140 #define USER_BUTTON_GPIO_PORT
    GPIOC
00141 #define USER_BUTTON_GPIO_CLK_ENABLE()
    __HAL_RCC_GPIOC_CLK_ENABLE()
00142 #define USER_BUTTON_GPIO_CLK_DISABLE()
    __HAL_RCC_GPIOC_CLK_DISABLE()
00143 #define USER_BUTTON_EXTI_LINE
    GPIO_PIN_13
00144 #define USER_BUTTON_EXTI_IRQn
    EXTI15_10_IRQn
00145
00146 #define BUTTONx_GPIO_CLK_ENABLE(__INDEX__)
     USER_BUTTON_GPIO_CLK_ENABLE()
00147 #define BUTTONx_GPIO_CLK_DISABLE(__INDEX__)
     USER_BUTTON_GPIO_CLK_DISABLE()
00148
00149 /* Aliases */
00150 #define KEY_BUTTON_PIN
 USER_BUTTON_PIN
00151 #define KEY_BUTTON_GPIO_PORT
 USER_BUTTON_GPIO_PORT
00152 #define KEY_BUTTON_GPIO_CLK_ENABLE()
 USER_BUTTON_GPIO_CLK_ENABLE()
00153 #define KEY_BUTTON_GPIO_CLK_DISABLE()
```

```
 USER_BUTTON_GPIO_CLK_DISABLE()
00154 #define KEY_BUTTON_EXTI_LINE
 USER_BUTTON_EXTI_LINE
00155 #define KEY_BUTTON_EXTI_IRQn
 USER_BUTTON_EXTI_IRQn
00156
00157 /**
00158   * @}
00159   */
00160
00161 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_BUS
 STM32F4XX NUCLEO LOW LEVEL BUS
00162   * @{
00163   */
00164 /*################################## SPI1 #####
###################################*/
00165 #ifdef HAL_SPI_MODULE_ENABLED
00166
00167 #define NUCLEO_SPIx
            SPI1
00168 #define NUCLEO_SPIx_CLK_ENABLE()
            __HAL_RCC_SPI1_CLK_ENABLE()
00169
00170 #define NUCLEO_SPIx_SCK_AF
            GPIO_AF5_SPI1
00171 #define NUCLEO_SPIx_SCK_GPIO_PORT
            GPIOA
00172 #define NUCLEO_SPIx_SCK_PIN
            GPIO_PIN_5
00173 #define NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE()
            __HAL_RCC_GPIOA_CLK_ENABLE()
00174 #define NUCLEO_SPIx_SCK_GPIO_CLK_DISABLE()
            __HAL_RCC_GPIOA_CLK_DISABLE()
00175
00176 #define NUCLEO_SPIx_MISO_MOSI_AF
            GPIO_AF5_SPI1
00177 #define NUCLEO_SPIx_MISO_MOSI_GPIO_PORT
```

```
            GPIOA
00178 #define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABL
E()         __HAL_RCC_GPIOA_CLK_ENABLE()
00179 #define NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_DISAB
LE()        __HAL_RCC_GPIOA_CLK_DISABLE()
00180 #define NUCLEO_SPIx_MISO_PIN
            GPIO_PIN_6
00181 #define NUCLEO_SPIx_MOSI_PIN
            GPIO_PIN_7
00182 /* Maximum Timeout values for flags waiting
loops. These timeouts are not based
00183    on accurate values, they just guarantee t
hat the application will not remain
00184    stuck if the SPI communication is corrupt
ed.
00185    You may modify these timeout values depen
ding on CPU frequency and application
00186    conditions (interrupts routines ...). */

00187 #define NUCLEO_SPIx_TIMEOUT_MAX
      1000
00188
00189 /**
00190   * @brief  SD Control Lines management
00191   */
00192 #define SD_CS_LOW()       HAL_GPIO_WritePin(
SD_CS_GPIO_PORT, SD_CS_PIN, GPIO_PIN_RESET)
00193 #define SD_CS_HIGH()      HAL_GPIO_WritePin(
SD_CS_GPIO_PORT, SD_CS_PIN, GPIO_PIN_SET)
00194
00195 /**
00196   * @brief  LCD Control Lines management
00197   */
00198 #define LCD_CS_LOW()      HAL_GPIO_WritePin(
LCD_CS_GPIO_PORT, LCD_CS_PIN, GPIO_PIN_RESET)
00199 #define LCD_CS_HIGH()     HAL_GPIO_WritePin(
LCD_CS_GPIO_PORT, LCD_CS_PIN, GPIO_PIN_SET)
```

```
00200 #define LCD_DC_LOW()      HAL_GPIO_WritePin(
LCD_DC_GPIO_PORT, LCD_DC_PIN, GPIO_PIN_RESET)
00201 #define LCD_DC_HIGH()     HAL_GPIO_WritePin(
LCD_DC_GPIO_PORT, LCD_DC_PIN, GPIO_PIN_SET)
00202
00203 /**
00204   * @brief  SD Control Interface pins (shiel
d D4)
00205   */
00206 #define SD_CS_PIN
      GPIO_PIN_5
00207 #define SD_CS_GPIO_PORT
      GPIOB
00208 #define SD_CS_GPIO_CLK_ENABLE()
    __HAL_RCC_GPIOB_CLK_ENABLE()
00209 #define SD_CS_GPIO_CLK_DISABLE()
    __HAL_RCC_GPIOB_CLK_DISABLE()
00210
00211 /**
00212   * @brief  LCD Control Interface pins (shie
ld D10)
00213   */
00214 #define LCD_CS_PIN
       GPIO_PIN_6
00215 #define LCD_CS_GPIO_PORT
       GPIOB
00216 #define LCD_CS_GPIO_CLK_ENABLE()
     __HAL_RCC_GPIOB_CLK_ENABLE()
00217 #define LCD_CS_GPIO_CLK_DISABLE()
     __HAL_RCC_GPIOB_CLK_DISABLE()
00218
00219 /**
00220   * @brief  LCD Data/Command Interface pins
(shield D8)
00221   */
00222 #define LCD_DC_PIN
       GPIO_PIN_9
```

```
00223 #define LCD_DC_GPIO_PORT
       GPIOA
00224 #define LCD_DC_GPIO_CLK_ENABLE()
     __HAL_RCC_GPIOA_CLK_ENABLE()
00225 #define LCD_DC_GPIO_CLK_DISABLE()
     __HAL_RCC_GPIOA_CLK_DISABLE()
00226
00227 #endif /* HAL_SPI_MODULE_ENABLED */
00228
00229 /*################################## ADC1 ####
####################################*/
00230 /**
00231   * @brief  ADC Interface pins
00232   *         used to detect motion of Joystic
k available on Adafruit 1.8" TFT shield
00233   */
00234
00235 #ifdef HAL_ADC_MODULE_ENABLED
00236
00237 #define NUCLEO_ADCx
       ADC1
00238 #define NUCLEO_ADCx_CLK_ENABLE()
      __HAL_RCC_ADC1_CLK_ENABLE()
00239 #define NUCLEO_ADCx_CLK_DISABLE()
      __HAL_RCC_ADC1_CLK_DISABLE()
00240
00241 #define NUCLEO_ADCx_CHANNEL
      ADC_CHANNEL_8
00242
00243 #define NUCLEO_ADCx_GPIO_PORT
       GPIOB
00244 #define NUCLEO_ADCx_GPIO_PIN
       GPIO_PIN_0
00245 #define NUCLEO_ADCx_GPIO_CLK_ENABLE()
      __HAL_RCC_GPIOB_CLK_ENABLE()
00246 #define NUCLEO_ADCx_GPIO_CLK_DISABLE()
      __HAL_RCC_GPIOB_CLK_DISABLE()
```

```
00247 #endif /* HAL_ADC_MODULE_ENABLED */
00248 /**
00249   * @}
00250   */
00251
00252 /**
00253   * @}
00254   */
00255
00256 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Exp
orted_Macros STM32F4XX NUCLEO LOW LEVEL Exported M
acros
00257   * @{
00258   */
00259 /**
00260   * @}
00261   */
00262
00263 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Exp
orted_Functions STM32F4XX NUCLEO LOW LEVEL Exporte
d Functions
00264   * @{
00265   */
00266 uint32_t         BSP_GetVersion(void);
00267 void             BSP_LED_Init(Led_TypeDef Le
d);
00268 void             BSP_LED_DeInit(Led_TypeDef
Led);
00269 void             BSP_LED_On(Led_TypeDef Led)
;
00270 void             BSP_LED_Off(Led_TypeDef Led
);
00271 void             BSP_LED_Toggle(Led_TypeDef
Led);
00272 void             BSP_PB_Init(Button_TypeDef
Button, ButtonMode_TypeDef ButtonMode);
00273 void             BSP_PB_DeInit(Button_TypeDef
```

```
 Button);
00274 uint32_t         BSP_PB_GetState(Button_Type
Def Button);
00275 #ifdef HAL_ADC_MODULE_ENABLED
00276 uint8_t          BSP_JOY_Init(void);
00277 JOYState_TypeDef BSP_JOY_GetState(void);
00278 void             BSP_JOY_DeInit(void);
00279 #endif /* HAL_ADC_MODULE_ENABLED */
00280
00281
00282 /**
00283   * @}
00284   */
00285
00286 /**
00287   * @}
00288   */
00289
00290 /**
00291   * @}
00292   */
00293
00294 /**
00295   * @}
00296   */
00297
00298 #ifdef __cplusplus
00299 }
00300 #endif
00301
00302 #endif /* __STM32F4XX_NUCLEO_H */
00303
00304 /************************ (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/
```

User Manual by  1.7.6.1

STM32F4xx-Nucleo BSP User Manual

stm32f4xx_nucleo.c

Go to the documentation of this file.

```
00001 /**
00002   ******************************************************************************
00003   * @file    stm32f4xx_nucleo.c
00004   * @author  MCD Application Team
00005   * @version V1.2.6
00006   * @date    31-January-2017
00007   * @brief   This file provides set of firmware functions to manage:
00008   *          - LEDs and push-button available on STM32F4XX-Nucleo Kit
00009   *            from STMicroelectronics
00010   *          - LCD, joystick and microSD available on Adafruit 1.8" TFT LCD
00011   *            shield (reference ID 802)
00012   ******************************************************************************
00013   * @attention
00014   *
00015   * <h2><center>&copy; COPYRIGHT(c) 2017 STMicroelectronics</center></h2>
00016   *
00017   * Redistribution and use in source and binary forms, with or without modification,
```

```
00018   * are permitted provided that the followin
g conditions are met:
00019   *   1. Redistributions of source code must
 retain the above copyright notice,
00020   *      this list of conditions and the fol
lowing disclaimer.
00021   *   2. Redistributions in binary form must
 reproduce the above copyright notice,
00022   *      this list of conditions and the fol
lowing disclaimer in the documentation
00023   *      and/or other materials provided wit
h the distribution.
00024   *   3. Neither the name of STMicroelectron
ics nor the names of its contributors
00025   *      may be used to endorse or promote p
roducts derived from this software
00026   *      without specific prior written perm
ission.
00027   *
00028   * THIS SOFTWARE IS PROVIDED BY THE COPYRIG
HT HOLDERS AND CONTRIBUTORS "AS IS"
00029   * AND ANY EXPRESS OR IMPLIED WARRANTIES, I
NCLUDING, BUT NOT LIMITED TO, THE
00030   * IMPLIED WARRANTIES OF MERCHANTABILITY AN
D FITNESS FOR A PARTICULAR PURPOSE ARE
00031   * DISCLAIMED. IN NO EVENT SHALL THE COPYRI
GHT HOLDER OR CONTRIBUTORS BE LIABLE
00032   * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP
ECIAL, 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, W
HETHER IN CONTRACT, STRICT LIABILITY,
00036   * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI
SE) 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 "stm32f4xx_nucleo.h"
00044
00045 /** @defgroup BSP BSP
00046   * @{
00047   */
00048
00049 /** @defgroup STM32F4XX_NUCLEO STM32F4XX NUC
LEO
00050   * @{
00051   */
00052
00053 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL STM
32F4XX NUCLEO LOW LEVEL
00054   * @brief This file provides set of firmwar
e functions to manage Leds and push-button
00055   *        available on STM32F4xx-Nucleo Kit
 from STMicroelectronics.
00056   * @{
00057   */
00058
00059 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Pri
vate_TypesDefinitions STM32F4XX NUCLEO LOW LEVEL P
rivate TypesDefinitions
00060   * @{
00061   */
00062 /**
00063   * @}
00064   */
```

```
00065
00066
00067 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Pri
vate_Defines STM32F4XX NUCLEO LOW LEVEL Private De
fines
00068   * @{
00069   */
00070
00071 /**
00072   * @brief STM32F4xx NUCLEO BSP Driver versi
on number V1.2.6
00073   */
00074 #define __STM32F4xx_NUCLEO_BSP_VERSION_MAIN
  (0x01) /*!< [31:24] main version */
00075 #define __STM32F4xx_NUCLEO_BSP_VERSION_SUB1
  (0x02) /*!< [23:16] sub1 version */
00076 #define __STM32F4xx_NUCLEO_BSP_VERSION_SUB2
  (0x06) /*!< [15:8]  sub2 version */
00077 #define __STM32F4xx_NUCLEO_BSP_VERSION_RC
  (0x00) /*!< [7:0]  release candidate */
00078 #define __STM32F4xx_NUCLEO_BSP_VERSION
  ((__STM32F4xx_NUCLEO_BSP_VERSION_MAIN << 24)\
00079
 |(__STM32F4xx_NUCLEO_BSP_VERSION_SUB1 << 16)\
00080
 |(__STM32F4xx_NUCLEO_BSP_VERSION_SUB2 << 8 )\
00081
 |(__STM32F4xx_NUCLEO_BSP_VERSION_RC))
00082
00083 /**
00084   * @brief LINK SD Card
00085   */
00086 #define SD_DUMMY_BYTE            0xFF
00087 #define SD_NO_RESPONSE_EXPECTED  0x80
00088
00089 /**
00090   * @}
```

```
00091   */
00092
00093 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Pri
vate_Macros STM32F4XX NUCLEO LOW LEVEL Private Mac
ros
00094   * @{
00095   */
00096 /**
00097   * @}
00098   */
00099
00100 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Pri
vate_Variables STM32F4XX NUCLEO LOW LEVEL Private
Variables
00101   * @{
00102   */
00103 GPIO_TypeDef* GPIO_PORT[LEDn] = {LED2_GPIO_P
ORT};
00104
00105 const uint16_t GPIO_PIN[LEDn] = {LED2_PIN};
00106
00107 GPIO_TypeDef* BUTTON_PORT[BUTTONn] = {KEY_BU
TTON_GPIO_PORT};
00108 const uint16_t BUTTON_PIN[BUTTONn] = {KEY_BU
TTON_PIN};
00109 const uint8_t BUTTON_IRQn[BUTTONn] = {KEY_BU
TTON_EXTI_IRQn};
00110
00111 /**
00112  * @brief BUS variables
00113  */
00114
00115 #ifdef ADAFRUIT_TFT_JOY_SD_ID802
00116 #ifdef HAL_SPI_MODULE_ENABLED
00117 uint32_t SpixTimeout = NUCLEO_SPIx_TIMEOUT_M
AX; /*<! Value of Timeout when SPI communication f
ails */
```

```
00118 static SPI_HandleTypeDef hnucleo_Spi;
00119 #endif /* HAL_SPI_MODULE_ENABLED */
00120
00121 #ifdef HAL_ADC_MODULE_ENABLED
00122 static ADC_HandleTypeDef hnucleo_Adc;
00123 /* ADC channel configuration structure decla
ration */
00124 static ADC_ChannelConfTypeDef sConfig;
00125 #endif /* HAL_ADC_MODULE_ENABLED */
00126 #endif /* ADAFRUIT_TFT_JOY_SD_ID802 */
00127
00128 /**
00129   * @}
00130   */
00131
00132 /** @defgroup STM32F4XX_NUCLEO_LOW_LEVEL_Pri
vate_FunctionPrototypes STM32F4XX NUCLEO LOW LEVEL
 Private FunctionPrototypes
00133   * @{
00134   */
00135 #ifdef ADAFRUIT_TFT_JOY_SD_ID802
00136
00137 #ifdef HAL_SPI_MODULE_ENABLED
00138 static void       SPIx_Init(void);
00139 static void       SPIx_Write(uint8_t Value);
00140 static void       SPIx_WriteReadData(const u
int8_t *DataIn, uint8_t *DataOut, uint16_t DataLeg
nth);
00141 static void       SPIx_Error(void);
00142 static void       SPIx_MspInit(SPI_HandleTyp
eDef *hspi);
00143
00144 /* SD IO functions */
00145 void              SD_IO_Init(void);
00146 void              SD_IO_CSState(uint8_t stat
e);
00147 void              SD_IO_WriteReadData(const
```

```
uint8_t *DataIn, uint8_t *DataOut, uint16_t DataLe
ngth);
00148 uint8_t           SD_IO_WriteByte(uint8_t Da
ta);
00149
00150 /* LCD IO functions */
00151 void              LCD_IO_Init(void);
00152 void              LCD_IO_WriteData(uint8_t D
ata);
00153 void              LCD_IO_WriteMultipleData(u
int8_t *pData, uint32_t Size);
00154 void              LCD_IO_WriteReg(uint8_t LC
DReg);
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_HandleTyp
eDef *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 STM32F4XX_NUCLEO_LOW_LEVEL_Pri
vate_Functions STM32F4XX NUCLEO LOW LEVEL Private
Functions
00172   * @{
00173   */
00174
```

```
00175 /**
00176   * @brief  This method returns the STM32F4x
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 __STM32F4xx_NUCLEO_BSP_VERSION;
00182 }
00183
00184 /**
00185   * @brief  Configures LED GPIO.
00186   * @param  Led: Specifies the Led to be con
figured.
00187   *   This parameter can be one of following
 parameters:
00188   *     @arg LED2
00189   */
00190 void BSP_LED_Init(Led_TypeDef Led)
00191 {
00192   GPIO_InitTypeDef  GPIO_InitStruct;
00193
00194   /* Enable the GPIO_LED Clock */
00195   LEDx_GPIO_CLK_ENABLE(Led);
00196
00197   /* Configure the GPIO_LED pin */
00198   GPIO_InitStruct.Pin = GPIO_PIN[Led];
00199   GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP
;
00200   GPIO_InitStruct.Pull = GPIO_NOPULL;
00201   GPIO_InitStruct.Speed = GPIO_SPEED_FAST;
00202
00203   HAL_GPIO_Init(GPIO_PORT[Led], &GPIO_InitSt
ruct);
00204
00205   HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN
```

```
[Led], GPIO_PIN_RESET);
00206 }
00207
00208 /**
00209   * @brief  DeInit LEDs.
00210   * @param  Led: LED to be de-init.
00211   *   This parameter can be one of the follo
wing values:
00212   *     @arg  LED2
00213   * @note Led DeInit does not disable the GP
IO clock nor disable the Mfx
00214   */
00215 void BSP_LED_DeInit(Led_TypeDef Led)
00216 {
00217   GPIO_InitTypeDef  gpio_init_structure;
00218
00219   /* Turn off LED */
00220   HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN
[Led], GPIO_PIN_RESET);
00221   /* DeInit the GPIO_LED pin */
00222   gpio_init_structure.Pin = GPIO_PIN[Led];
00223   HAL_GPIO_DeInit(GPIO_PORT[Led], gpio_init_
structure.Pin);
00224 }
00225
00226 /**
00227   * @brief  Turns selected LED On.
00228   * @param  Led: Specifies the Led to be set
 on.
00229   *   This parameter can be one of following
 parameters:
00230   *     @arg LED2
00231   */
00232 void BSP_LED_On(Led_TypeDef Led)
00233 {
00234   HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN
[Led], GPIO_PIN_SET);
```

```
00235 }
00236
00237 /**
00238   * @brief  Turns selected LED Off.
00239   * @param  Led: Specifies the Led to be set
 off.
00240   *   This parameter can be one of following
 parameters:
00241   *     @arg LED2
00242   */
00243 void BSP_LED_Off(Led_TypeDef Led)
00244 {
00245   HAL_GPIO_WritePin(GPIO_PORT[Led], GPIO_PIN
[Led], GPIO_PIN_RESET);
00246 }
00247
00248 /**
00249   * @brief  Toggles the selected LED.
00250   * @param  Led: Specifies the Led to be tog
gled.
00251   *   This parameter can be one of following
 parameters:
00252   *     @arg LED2
00253   */
00254 void BSP_LED_Toggle(Led_TypeDef Led)
00255 {
00256   HAL_GPIO_TogglePin(GPIO_PORT[Led], GPIO_PIN
[Led]);
00257 }
00258
00259 /**
00260   * @brief  Configures Button GPIO and EXTI
Line.
00261   * @param  Button: Specifies the Button to
be configured.
00262   *   This parameter should be: BUTTON_KEY
00263   * @param  ButtonMode: Specifies Button mod
```

```
e.
00264   *   This parameter can be one of following
 parameters:
00265   *     @arg BUTTON_MODE_GPIO: Button will b
e used as simple IO
00266   *     @arg BUTTON_MODE_EXTI: Button will b
e connected to EXTI line with interrupt
00267   *                            generation ca
pability
00268   */
00269 void BSP_PB_Init(Button_TypeDef Button, Butt
onMode_TypeDef ButtonMode)
00270 {
00271   GPIO_InitTypeDef GPIO_InitStruct;
00272
00273   /* Enable the BUTTON Clock */
00274   BUTTONx_GPIO_CLK_ENABLE(Button);
00275
00276   if(ButtonMode == BUTTON_MODE_GPIO)
00277   {
00278     /* Configure Button pin as input */
00279     GPIO_InitStruct.Pin = BUTTON_PIN[Button]
;
00280     GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
00281     GPIO_InitStruct.Pull = GPIO_PULLDOWN;
00282     GPIO_InitStruct.Speed = GPIO_SPEED_FAST;
00283     HAL_GPIO_Init(BUTTON_PORT[Button], &GPIO
_InitStruct);
00284   }
00285
00286   if(ButtonMode == BUTTON_MODE_EXTI)
00287   {
00288     /* Configure Button pin as input with Ex
ternal interrupt */
00289     GPIO_InitStruct.Pin = BUTTON_PIN[Button]
;
00290     GPIO_InitStruct.Pull = GPIO_NOPULL;
```

```
00291     GPIO_InitStruct.Mode = GPIO_MODE_IT_FALL
ING;
00292     HAL_GPIO_Init(BUTTON_PORT[Button], &GPIO
_InitStruct);
00293
00294     /* Enable and set Button EXTI Interrupt
to the lowest priority */
00295     HAL_NVIC_SetPriority((IRQn_Type)(BUTTON_
IRQn[Button]), 0x0F, 0x00);
00296     HAL_NVIC_EnableIRQ((IRQn_Type)(BUTTON_IR
Qn[Button]));
00297   }
00298 }
00299
00300 /**
00301   * @brief  Push Button DeInit.
00302   * @param  Button: Button to be configured
00303   *   This parameter should be: BUTTON_KEY
00304   * @note PB DeInit does not disable the GPI
O clock
00305   */
00306 void BSP_PB_DeInit(Button_TypeDef Button)
00307 {
00308     GPIO_InitTypeDef gpio_init_structure;
00309
00310     gpio_init_structure.Pin = BUTTON_PIN[But
ton];
00311     HAL_NVIC_DisableIRQ((IRQn_Type)(BUTTON_I
RQn[Button]));
00312     HAL_GPIO_DeInit(BUTTON_PORT[Button], gpi
o_init_structure.Pin);
00313 }
00314
00315 /**
00316   * @brief  Returns the selected Button stat
e.
00317   * @param  Button: Specifies the Button to
```

```
be checked.
00318   *   This parameter should be: BUTTON_KEY
00319   * @retval The Button GPIO pin value.
00320   */
00321 uint32_t BSP_PB_GetState(Button_TypeDef Butt
on)
00322 {
00323   return HAL_GPIO_ReadPin(BUTTON_PORT[Button
], BUTTON_PIN[Button]);
00324 }
00325
00326 /**********************************************
**********************************
00327                                     BUS OPERATIONS
00328 **********************************************
**********************************/
00329 #ifdef ADAFRUIT_TFT_JOY_SD_ID802
00330
00331 /******************************** SPI *******
************************/
00332 #ifdef HAL_SPI_MODULE_ENABLED
00333
00334 /**
00335   * @brief  Initializes SPI MSP.
00336   */
00337 static void SPIx_MspInit(SPI_HandleTypeDef *
hspi)
00338 {
00339   GPIO_InitTypeDef  GPIO_InitStruct;
00340
00341   /*** Configure the GPIOs ***/
00342   /* Enable GPIO clock */
00343   NUCLEO_SPIx_SCK_GPIO_CLK_ENABLE();
00344   NUCLEO_SPIx_MISO_MOSI_GPIO_CLK_ENABLE();
00345
00346   /* Configure SPI SCK */
00347   GPIO_InitStruct.Pin = NUCLEO_SPIx_SCK_PIN;
```

```
00348   GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
00349   GPIO_InitStruct.Pull  = GPIO_PULLUP;
00350   GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
00351   GPIO_InitStruct.Alternate = NUCLEO_SPIx_SC
K_AF;
00352   HAL_GPIO_Init(NUCLEO_SPIx_SCK_GPIO_PORT, &
GPIO_InitStruct);
00353
00354   /* Configure SPI MISO and MOSI */
00355   GPIO_InitStruct.Pin = NUCLEO_SPIx_MOSI_PIN
;
00356   GPIO_InitStruct.Alternate = NUCLEO_SPIx_MI
SO_MOSI_AF;
00357   GPIO_InitStruct.Pull  = GPIO_PULLDOWN;
00358   HAL_GPIO_Init(NUCLEO_SPIx_MISO_MOSI_GPIO_P
ORT, &GPIO_InitStruct);
00359
00360   GPIO_InitStruct.Pin = NUCLEO_SPIx_MISO_PIN
;
00361   GPIO_InitStruct.Pull  = GPIO_PULLDOWN;
00362   HAL_GPIO_Init(NUCLEO_SPIx_MISO_MOSI_GPIO_P
ORT, &GPIO_InitStruct);
00363
00364   /*** Configure the SPI peripheral ***/
00365   /* Enable SPI clock */
00366   NUCLEO_SPIx_CLK_ENABLE();
00367 }
00368
00369 /**
00370   * @brief  Initializes SPI HAL.
00371   */
00372 static void SPIx_Init(void)
00373 {
00374   if(HAL_SPI_GetState(&hnucleo_Spi) == HAL_S
PI_STATE_RESET)
00375   {
00376     /* SPI Config */
```

```
00377     hnucleo_Spi.Instance = NUCLEO_SPIx;
00378       /* SPI baudrate is set to 12,5 MHz max
imum (APB1/SPI_BaudRatePrescaler = 100/8 = 12,5 MH
z)
00379        to verify these constraints:
00380           - ST7735 LCD SPI interface max bau
drate is 15MHz for write and 6.66MHz for read
00381             Since the provided driver doesn'
t use read capability from LCD, only constraint
00382             on write baudrate is considered.
00383           - SD card SPI interface max baudra
te is 25MHz for write/read
00384           - PCLK2 max frequency is 100 MHz
00385        */
00386     hnucleo_Spi.Init.BaudRatePrescaler = SPI
_BAUDRATEPRESCALER_8;
00387     hnucleo_Spi.Init.Direction = SPI_DIRECTI
ON_2LINES;
00388     hnucleo_Spi.Init.CLKPhase = SPI_PHASE_2E
DGE;
00389     hnucleo_Spi.Init.CLKPolarity = SPI_POLAR
ITY_HIGH;
00390     hnucleo_Spi.Init.CRCCalculation = SPI_CR
CCALCULATION_DISABLED;
00391     hnucleo_Spi.Init.CRCPolynomial = 7;
00392     hnucleo_Spi.Init.DataSize = SPI_DATASIZE
_8BIT;
00393     hnucleo_Spi.Init.FirstBit = SPI_FIRSTBIT
_MSB;
00394     hnucleo_Spi.Init.NSS = SPI_NSS_SOFT;
00395     hnucleo_Spi.Init.TIMode = SPI_TIMODE_DIS
ABLED;
00396     hnucleo_Spi.Init.Mode = SPI_MODE_MASTER;
00397
00398     SPIx_MspInit(&hnucleo_Spi);
00399     HAL_SPI_Init(&hnucleo_Spi);
00400   }
```

```
00401 }
00402
00403 /**
00404   * @brief  SPI Write a byte to device
00405   * @param  DataIn: value to be written
00406   * @param  DataOut: value to be read
00407   * @param  DataLegnth: length of data
00408   */
00409 static void SPIx_WriteReadData(const uint8_t
 *DataIn, uint8_t *DataOut, uint16_t DataLegnth)
00410 {
00411   HAL_StatusTypeDef status = HAL_OK;
00412
00413   status = HAL_SPI_TransmitReceive(&hnucleo_
Spi, (uint8_t*) DataIn, DataOut, DataLegnth, SpixT
imeout);
00414
00415   /* Check the communication status */
00416   if(status != HAL_OK)
00417   {
00418     /* Execute user timeout callback */
00419     SPIx_Error();
00420   }
00421 }
00422
00423 /**
00424   * @brief  SPI Write a byte to device.
00425   * @param  Value: value to be written
00426   */
00427 static void SPIx_Write(uint8_t Value)
00428 {
00429   HAL_StatusTypeDef status = HAL_OK;
00430   uint8_t data;
00431
00432   status = HAL_SPI_TransmitReceive(&hnucleo_
Spi, (uint8_t*) &Value, &data, 1, SpixTimeout);
00433
```

```
00434   /* Check the communication status */
00435   if(status != HAL_OK)
00436   {
00437     /* Execute user timeout callback */
00438     SPIx_Error();
00439   }
00440 }
00441
00442 /**
00443   * @brief  SPI error treatment function.
00444   */
00445 static void SPIx_Error (void)
00446 {
00447   /* De-initialize the SPI communication BUS
 */
00448   HAL_SPI_DeInit(&hnucleo_Spi);
00449
00450   /* Re-Initiaize the SPI communication BUS
*/
00451   SPIx_Init();
00452 }
00453
00454 /**********************************************
**********************************
00455                                       LINK OPERATIONS
00456 **********************************************
**********************************/
00457
00458 /********************************** LINK SD *
**********************************/
00459 /**
00460   * @brief  Initializes the SD Card and put
it into StandBy State (Ready for
00461   *         data transfer).
00462   */
00463 void SD_IO_Init(void)
00464 {
```

```
00465   GPIO_InitTypeDef  GPIO_InitStruct;
00466   uint8_t counter;
00467
00468   /* SD_CS_GPIO Periph clock enable */
00469   SD_CS_GPIO_CLK_ENABLE();
00470
00471   /* Configure SD_CS_PIN pin: SD Card CS pin
 */
00472   GPIO_InitStruct.Pin = SD_CS_PIN;
00473   GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP
;
00474   GPIO_InitStruct.Pull = GPIO_PULLUP;
00475   GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
00476   HAL_GPIO_Init(SD_CS_GPIO_PORT, &GPIO_InitS
truct);
00477
00478   /*------------Put SD in SPI mode----------
----*/
00479   /* SD SPI Config */
00480   SPIx_Init();
00481
00482   /* SD chip select high */
00483   SD_CS_HIGH();
00484
00485   /* Send dummy byte 0xFF, 10 times with CS
high */
00486   /* Rise CS and MOSI for 80 clocks cycles */

00487   for (counter = 0; counter <= 9; counter++)
00488   {
00489     /* Send dummy byte 0xFF */
00490     SD_IO_WriteByte(SD_DUMMY_BYTE);
00491   }
00492 }
00493
00494 /**
00495   * @brief Set the SD_CS pin.
```

```
00496   * @param val: pin value.
00497   */
00498 void SD_IO_CSState(uint8_t val)
00499 {
00500   if(val == 1)
00501   {
00502     SD_CS_HIGH();
00503   }
00504   else
00505   {
00506     SD_CS_LOW();
00507   }
00508 }
00509
00510 /**
00511   * @brief Write a byte on the SD.
00512   * @param  DataIn: value to be written
00513   * @param  DataOut: value to be read
00514   * @param  DataLength: length of data
00515   */
00516 void SD_IO_WriteReadData(const uint8_t *Data
In, uint8_t *DataOut, uint16_t DataLength)
00517 {
00518   /* Send the byte */
00519   SPIx_WriteReadData(DataIn, DataOut, DataLe
ngth);
00520 }
00521
00522 /**
00523   * @brief  Writes a byte on the SD.
00524   * @param  Data: byte to send.
00525   */
00526 uint8_t SD_IO_WriteByte(uint8_t Data)
00527 {
00528   uint8_t tmp;
00529   /* Send the byte */
00530   SPIx_WriteReadData(&Data,&tmp,1);
```

```
00531   return tmp;
00532 }
00533
00534 /********************************** LINK LCD
**********************************/
00535 /**
00536   * @brief  Initializes the LCD.
00537   */
00538 void LCD_IO_Init(void)
00539 {
00540   GPIO_InitTypeDef  GPIO_InitStruct;
00541
00542   /* LCD_CS_GPIO and LCD_DC_GPIO Periph cloc
k enable */
00543   LCD_CS_GPIO_CLK_ENABLE();
00544   LCD_DC_GPIO_CLK_ENABLE();
00545
00546   /* Configure LCD_CS_PIN pin: LCD Card CS p
in */
00547   GPIO_InitStruct.Pin = LCD_CS_PIN;
00548   GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP
;
00549   GPIO_InitStruct.Pull = GPIO_NOPULL;
00550   GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
00551   HAL_GPIO_Init(LCD_CS_GPIO_PORT, &GPIO_Init
Struct);
00552
00553   /* Configure LCD_DC_PIN pin: LCD Card DC p
in */
00554   GPIO_InitStruct.Pin = LCD_DC_PIN;
00555   HAL_GPIO_Init(LCD_DC_GPIO_PORT, &GPIO_Init
Struct);
00556
00557   /* LCD chip select high */
00558   LCD_CS_HIGH();
00559
00560   /* LCD SPI Config */
```

```
00561   SPIx_Init();
00562 }
00563
00564 /**
00565   * @brief  Writes command to select the LCD
 register.
00566   * @param  LCDReg: Address of the selected
register.
00567   */
00568 void LCD_IO_WriteReg(uint8_t LCDReg)
00569 {
00570   /* Reset LCD control line CS */
00571   LCD_CS_LOW();
00572
00573   /* Set LCD data/command line DC to Low */
00574   LCD_DC_LOW();
00575
00576   /* Send Command */
00577   SPIx_Write(LCDReg);
00578
00579   /* Deselect : Chip Select high */
00580   LCD_CS_HIGH();
00581 }
00582
00583 /**
00584   * @brief  Writes data to select the LCD re
gister.
00585   *         This function must be used after
 st7735_WriteReg() function
00586   * @param  Data: data to write to the selec
ted register.
00587   */
00588 void LCD_IO_WriteData(uint8_t Data)
00589 {
00590   /* Reset LCD control line CS */
00591   LCD_CS_LOW();
00592
```

```
00593   /* Set LCD data/command line DC to High */
00594   LCD_DC_HIGH();
00595
00596   /* Send Data */
00597   SPIx_Write(Data);
00598
00599   /* Deselect : Chip Select high */
00600   LCD_CS_HIGH();
00601 }
00602
00603 /**
00604   * @brief  Writes register value.
00605   * @param  pData: Pointer on the register v
alue
00606   * @param  Size: Size of byte to transmit t
o the register
00607   */
00608 void LCD_IO_WriteMultipleData(uint8_t *pData
, uint32_t Size)
00609 {
00610   uint32_t counter = 0;
00611   __IO uint32_t data = 0;
00612
00613   /* Reset LCD control line CS */
00614   LCD_CS_LOW();
00615
00616   /* Set LCD data/command line DC to High */
00617   LCD_DC_HIGH();
00618
00619   if (Size == 1)
00620   {
00621     /* Only 1 byte to be sent to LCD - gener
al interface can be used */
00622     /* Send Data */
00623     SPIx_Write(*pData);
00624   }
00625   else
```

```
00626   {
00627     /* Several data should be sent in a raw
*/
00628     /* Direct SPI accesses for optimization
*/
00629     for (counter = Size; counter != 0; count
er--)
00630     {
00631       while(((hnucleo_Spi.Instance->SR) & SP
I_FLAG_TXE) != SPI_FLAG_TXE)
00632       {
00633       }
00634       /* Need to invert bytes for LCD*/
00635       *((__IO uint8_t*)&hnucleo_Spi.Instance
->DR) = *(pData+1);
00636
00637       while(((hnucleo_Spi.Instance->SR) & SP
I_FLAG_TXE) != SPI_FLAG_TXE)
00638       {
00639       }
00640       *((__IO uint8_t*)&hnucleo_Spi.Instance
->DR) = *pData;
00641       counter--;
00642       pData += 2;
00643       }
00644
00645     /* Wait until the bus is ready before re
leasing Chip select */
00646     while(((hnucleo_Spi.Instance->SR) & SPI_
FLAG_BSY) != RESET)
00647     {
00648     }
00649   }
00650
00651   /* Empty the Rx fifo */
00652   data = *(&hnucleo_Spi.Instance->DR);
00653   UNUSED(data);
```

```
00654
00655   /* Deselect : Chip Select high */
00656   LCD_CS_HIGH();
00657 }
00658
00659 /**
00660   * @brief  Wait for loop in ms.
00661   * @param  Delay in ms.
00662   */
00663 void LCD_Delay(uint32_t Delay)
00664 {
00665   HAL_Delay(Delay);
00666 }
00667 #endif /* HAL_SPI_MODULE_ENABLED */
00668
00669 /****************************** ADC driver
****************************/
00670 #ifdef HAL_ADC_MODULE_ENABLED
00671
00672 /**
00673   * @brief  Initializes ADC MSP.
00674   */
00675 static void ADCx_MspInit(ADC_HandleTypeDef *
hadc)
00676 {
00677   GPIO_InitTypeDef  GPIO_InitStruct;
00678
00679   /*** Configure the GPIOs ***/
00680   /* Enable GPIO clock */
00681   NUCLEO_ADCx_GPIO_CLK_ENABLE();
00682
00683   /* Configure the selected ADC Channel as a
nalog input */
00684   GPIO_InitStruct.Pin = NUCLEO_ADCx_GPIO_PIN
 ;
00685   GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
00686   GPIO_InitStruct.Pull = GPIO_NOPULL;
```

```
00687   HAL_GPIO_Init(NUCLEO_ADCx_GPIO_PORT, &GPIO
_InitStruct);
00688
00689   /*** Configure the ADC peripheral ***/
00690   /* Enable ADC clock */
00691   NUCLEO_ADCx_CLK_ENABLE();
00692 }
00693
00694 /**
00695   * @brief  DeInitializes ADC MSP.
00696   * @note ADC DeInit does not disable the GP
IO clock
00697   */
00698 static void ADCx_MspDeInit(ADC_HandleTypeDef
 *hadc)
00699 {
00700   GPIO_InitTypeDef  GPIO_InitStruct;
00701
00702   /*** DeInit the ADC peripheral ***/
00703   /* Disable ADC clock */
00704   NUCLEO_ADCx_CLK_DISABLE();
00705
00706   /* Configure the selected ADC Channel as a
nalog input */
00707   GPIO_InitStruct.Pin = NUCLEO_ADCx_GPIO_PIN
 ;
00708   HAL_GPIO_DeInit(NUCLEO_ADCx_GPIO_PORT, GPI
O_InitStruct.Pin);
00709
00710   /* Disable GPIO clock has to be done by th
e application*/
00711   /* NUCLEO_ADCx_GPIO_CLK_DISABLE(); */
00712 }
00713
00714 /**
00715   * @brief  Initializes ADC HAL.
00716   */
```

```
00717 static void ADCx_Init(void)
00718 {
00719   if(HAL_ADC_GetState(&hnucleo_Adc) == HAL_A
DC_STATE_RESET)
00720   {
00721     /* ADC Config */
00722     hnucleo_Adc.Instance                   =
NUCLEO_ADCx;
00723     hnucleo_Adc.Init.ClockPrescaler        =
 ADC_CLOCKPRESCALER_PCLK_DIV4; /* (must not exceed
 36MHz) */
00724     hnucleo_Adc.Init.Resolution            =
 ADC_RESOLUTION12b;
00725     hnucleo_Adc.Init.DataAlign             =
 ADC_DATAALIGN_RIGHT;
00726     hnucleo_Adc.Init.ContinuousConvMode    =
 DISABLE;
00727     hnucleo_Adc.Init.DiscontinuousConvMode =
 DISABLE;
00728     hnucleo_Adc.Init.ExternalTrigConvEdge  =
 ADC_EXTERNALTRIGCONVEDGE_NONE;
00729     hnucleo_Adc.Init.EOCSelection          =
 EOC_SINGLE_CONV;
00730     hnucleo_Adc.Init.NbrOfConversion       =
 1;
00731     hnucleo_Adc.Init.DMAContinuousRequests =
 DISABLE;
00732
00733     ADCx_MspInit(&hnucleo_Adc);
00734     HAL_ADC_Init(&hnucleo_Adc);
00735   }
00736 }
00737
00738 /**
00739   * @brief  Initializes ADC HAL.
00740   */
00741 static void ADCx_DeInit(void)
```

```
00742 {
00743     hnucleo_Adc.Instance   = NUCLEO_ADCx;
00744
00745     HAL_ADC_DeInit(&hnucleo_Adc);
00746     ADCx_MspDeInit(&hnucleo_Adc);
00747 }
00748
00749 /******************************** LINK JOYSTI
CK ********************************/
00750
00751 /**
00752   * @brief  Configures joystick available on
 adafruit 1.8" TFT shield
00753   *         managed through ADC to detect mo
tion.
00754   * @retval Joystickstatus (0=> success, 1=>
 fail)
00755   */
00756 uint8_t BSP_JOY_Init(void)
00757 {
00758   uint8_t status = HAL_ERROR;
00759
00760   ADCx_Init();
00761
00762   /* Select the ADC Channel to be converted
*/
00763   sConfig.Channel = NUCLEO_ADCx_CHANNEL;
00764   sConfig.SamplingTime = ADC_SAMPLETIME_3CYC
LES;
00765   sConfig.Rank = 1;
00766   status = HAL_ADC_ConfigChannel(&hnucleo_Adc
, &sConfig);
00767
00768   /* Return Joystick initialization status */

00769   return status;
00770 }
```

```
00771
00772 /**
00773   * @brief  DeInit joystick GPIOs.
00774   * @note   JOY DeInit does not disable the
Mfx, just set the Mfx pins in Off modee.
00775   */
00776 void BSP_JOY_DeInit(void)
00777 {
00778   ADCx_DeInit();
00779 }
00780
00781 /**
00782   * @brief  Returns the Joystick key pressed.

00783   * @note   To know which Joystick key is pr
essed we need to detect the voltage
00784   *         level on each key output
00785   *           - None  : 3.3 V / 4095
00786   *           - SEL   : 1.055 V / 1308
00787   *           - DOWN  : 0.71 V / 88
00788   *           - LEFT  : 3.0 V / 3720
00789   *           - RIGHT : 0.595 V / 737
00790   *           - UP    : 1.65 V / 2046
00791   * @retval JOYState_TypeDef: Code of the Jo
ystick key pressed.
00792   */
00793 JOYState_TypeDef BSP_JOY_GetState(void)
00794 {
00795   JOYState_TypeDef state;
00796   uint16_t  keyconvertedvalue = 0;
00797
00798   /* Start the conversion process */
00799   HAL_ADC_Start(&hnucleo_Adc);
00800
00801   /* Wait for the end of conversion */
00802   HAL_ADC_PollForConversion(&hnucleo_Adc, 10
);
```

```
00803
00804   /* Check if the continuous conversion of r
egular channel is finished */
00805   if(((HAL_ADC_GetState(&hnucleo_Adc) & HAL_
ADC_STATE_EOC_REG) == HAL_ADC_STATE_EOC_REG))
00806   {
00807     /* Get the converted value of regular ch
annel */
00808     keyconvertedvalue = HAL_ADC_GetValue(&hn
ucleo_Adc);
00809   }
00810
00811   if((keyconvertedvalue > 2010) && (keyconve
rtedvalue < 2090))
00812   {
00813     state = JOY_UP;
00814   }
00815   else if((keyconvertedvalue > 680) && (keyc
onvertedvalue < 780))
00816   {
00817     state = JOY_RIGHT;
00818   }
00819   else if((keyconvertedvalue > 1270) && (key
convertedvalue < 1350))
00820   {
00821     state = JOY_SEL;
00822   }
00823   else if((keyconvertedvalue > 50) && (keyco
nvertedvalue < 130))
00824   {
00825     state = JOY_DOWN;
00826   }
00827   else if((keyconvertedvalue > 3680) && (key
convertedvalue < 3760))
00828   {
00829     state = JOY_LEFT;
00830   }
```

```
00831   else
00832   {
00833     state = JOY_NONE;
00834   }
00835
00836   /* Loop while a key is pressed */
00837   if(state != JOY_NONE)
00838   {
00839     keyconvertedvalue = HAL_ADC_GetValue(&hn
ucleo_Adc);
00840   }
00841   /* Return the code of the Joystick key pre
ssed */
00842   return state;
00843 }
00844 #endif /* HAL_ADC_MODULE_ENABLED */
00845
00846 #endif /* ADAFRUIT_TFT_JOY_SD_ID802 */
00847
00848
00849 /**
00850   * @}
00851   */
00852
00853 /**
00854   * @}
00855   */
00856
00857 /**
00858   * @}
00859   */
00860
00861 /**
00862   * @}
00863   */
00864
00865 /************************ (C) COPYRIGHT STMi
```

```
croelectronics *****END OF FILE****/
```

STM32F4xx-Nucleo BSP User Manual

BSP

Modules

STM32F4XX NUCLEO

Generated on Fri Jan 20 2017 15:36:09 for STM32F4xx-Nucleo BSP User Manual by doxygen 1.7.6.1

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO

BSP

Modules

STM32F4XX NUCLEO LOW LEVEL

This file provides set of firmware functions to manage Leds and push-button available on STM32F4xx-Nucleo Kit from STMicroelectronics.

STM32F4xx-Nucleo BSP User Manual

STM32F4XX NUCLEO LOW LEVEL Exported Constants

STM32F4XX NUCLEO LOW LEVEL

Modules

STM32F4XX NUCLEO LOW LEVEL LED

Define for STM32F4XX_NUCLEO board.

STM32F4XX NUCLEO LOW LEVEL BUTTON

STM32F4XX NUCLEO LOW LEVEL BUS