

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_LOW_LEVEL Private Types Definitions

[STM32746G_DISCOVERY_LOW_LEVEL](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_LOW_LEVEL Private Macros

[STM32746G_DISCOVERY_LOW_LEVEL](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_LOW_LEVEL Private Function Prototypes

[STM32746G_DISCOVERY_LOW_LEVEL](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_LOW_LEVEL Exported Macros

[STM32746G_DISCOVERY_LOW_LEVEL](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY AUDIO Private Types

[STM32746G_DISCOVERY AUDIO](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY AUDIO Private Defines

[STM32746G_DISCOVERY AUDIO](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY AUDIO Private Macros

[STM32746G_DISCOVERY AUDIO](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY AUDIO Private Function Prototypes

[STM32746G_DISCOVERY AUDIO](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_AUDIO Exported Types

[STM32746G_DISCOVERY AUDIO](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Data Structures	Data Structure Index	Data Fields	Data Fields
QSPI_Info Struct Reference			Data Fields
STM32746G_DISCOVERY_QSPI Exported Types			

```
#include <stm32746g_discovery_qspi.h>
```

Data Fields

uint32_t	FlashSize
uint32_t	EraseSectorSize
uint32_t	EraseSectorsNumber
uint32_t	ProgPageSize
uint32_t	ProgPagesNumber

Detailed Description

Definition at line [119](#) of file `stm32746g_discovery_qspi.h`.

Field Documentation

`uint32_t QSPI_Info::EraseSectorSize`

Size of sectors for the erase operation

Definition at line [121](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_GetInfo()`.

`uint32_t QSPI_Info::EraseSectorsNumber`

Number of sectors for the erase operation

Definition at line [122](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_GetInfo()`.

`uint32_t QSPI_Info::FlashSize`

Size of the flash

Definition at line [120](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_GetInfo()`.

`uint32_t QSPI_Info::ProgPageSize`

Size of pages for the program operation

Definition at line [123](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_GetInfo()`.

uint32_t QSPI_Info::ProgPagesNumber

Number of pages for the program operation

Definition at line **124** of file **stm32746g_discovery_qspi.h**.

Referenced by **BSP_QSPI_GetInfo()**.

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

- **stm32746g_discovery_qspi.h**
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [**doxygen**](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SD Private Types Definitions

[STM32746G_DISCOVERY_SD](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SD Private Defines

[STM32746G_DISCOVERY_SD](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SD Private Macros

[STM32746G_DISCOVERY_SD](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SD Private Function Prototypes

[STM32746G_DISCOVERY_SD](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SD Exported Macro

[STM32746G_DISCOVERY_SD](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SDRAM Private Types Definitions

[STM32746G_DISCOVERY_SDRAM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SDRAM Private Defines

[STM32746G_DISCOVERY_SDRAM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SDRAM Private Macros

[STM32746G_DISCOVERY_SDRAM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SDRAM Private Function Prototypes

[STM32746G_DISCOVERY_SDRAM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_SDRAM Exported Macro

[STM32746G_DISCOVERY_SDRAM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_TS Types Definitions

[STM32746G_DISCOVERY_TS](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_TS Types Defines

[STM32746G_DISCOVERY_TS](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_TS Private Macros

[STM32746G_DISCOVERY_TS](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_TS Private Function Prototypes

[STM32746G_DISCOVERY_TS](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Data Structures	Data Structure Index	Data Fields	Data Fields
TS_StateTypeDef Struct Reference STM32746G_DISCOVERY_TS Exported Types			

TS_StateTypeDef Define TS State structure. [More...](#)

```
#include <stm32746g_discovery_ts.h>
```

Data Fields

```
uint8_t touchDetected  
uint16_t touchX [TS_MAX_NB_TOUCH]  
uint16_t touchY [TS_MAX_NB_TOUCH]  
uint8_t touchWeight [TS_MAX_NB_TOUCH]  
uint8_t touchEventId [TS_MAX_NB_TOUCH]  
uint8_t touchArea [TS_MAX_NB_TOUCH]  
uint32_t gestureId
```

Detailed Description

TS_StateTypeDef Define TS State structure.

Definition at line **91** of file **stm32746g_discovery_ts.h**.

Field Documentation

`uint32_t TS_StateTypeDef::gestureId`

type of gesture detected : take value of type `TS_GestureIdTypeDef`

Definition at line [101](#) of file `stm32746g_discovery_ts.h`.

Referenced by `BSP_TS_GetGestureId()`, and
`BSP_TS_ResetTouchData()`.

`uint8_t TS_StateTypeDef::touchArea[TS_MAX_NB_TOUCH]`

`Touch_Area[0], Touch_Area[1]` : touch area of each touch

Definition at line [100](#) of file `stm32746g_discovery_ts.h`.

Referenced by `BSP_TS_GetState()`, and
`BSP_TS_ResetTouchData()`.

`uint8_t TS_StateTypeDef::touchDetected`

Total number of active touches detected at last scan

Definition at line [93](#) of file `stm32746g_discovery_ts.h`.

Referenced by `BSP_TS_GetState()`, and
`BSP_TS_ResetTouchData()`.

`uint8_t TS_StateTypeDef::touchEventId[TS_MAX_NB_TOUCH]`

`Touch_EventId[0], Touch_EventId[1]` : take value of type
`TS_TouchEventTypeDef`

Definition at line **99** of file [stm32746g_discovery_ts.h](#).

Referenced by [BSP_TS_GetState\(\)](#), and
[BSP_TS_ResetTouchData\(\)](#).

uint8_t TS_StateTypeDef::touchWeight[TS_MAX_NB_TOUCH]

Touch_Weight[0], Touch_Weight[1] : weight property of touches

Definition at line **98** of file [stm32746g_discovery_ts.h](#).

Referenced by [BSP_TS_GetState\(\)](#), and
[BSP_TS_ResetTouchData\(\)](#).

uint16_t TS_StateTypeDef::touchX[TS_MAX_NB_TOUCH]

Touch X[0], X[1] coordinates on 12 bits

Definition at line **94** of file [stm32746g_discovery_ts.h](#).

Referenced by [BSP_TS_GetState\(\)](#), and
[BSP_TS_ResetTouchData\(\)](#).

uint16_t TS_StateTypeDef::touchY[TS_MAX_NB_TOUCH]

Touch Y[0], Y[1] coordinates on 12 bits

Definition at line **95** of file [stm32746g_discovery_ts.h](#).

Referenced by [BSP_TS_GetState\(\)](#), and
[BSP_TS_ResetTouchData\(\)](#).

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

- **stm32746g_discovery_ts.h**
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_CAMERA Private Types Definitions

[STM32746G_DISCOVERY_CAMERA](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_CAMERA Private Macros

[STM32746G_DISCOVERY_CAMERA](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_CAMERA Private Function Prototypes

[STM32746G_DISCOVERY_CAMERA](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_EEPROM Private Types

[STM32746G_DISCOVERY_EEPROM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_EEPROM Private Defines

[STM32746G_DISCOVERY_EEPROM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_EEPROM Private Macros

[STM32746G_DISCOVERY_EEPROM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_EEPROM Private Function Prototypes

[STM32746G_DISCOVERY_EEPROM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_EEPROM Exported Types

[STM32746G_DISCOVERY_EEPROM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_EEPROM Exported Macros

[STM32746G_DISCOVERY_EEPROM](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_LCD Private Types Definitions

[STM32746G_DISCOVERY_LCD](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

STM32746G_DISCOVERY_LCD Private Function Prototypes

[STM32746G_DISCOVERY_LCD](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Data Structures	Data Structure Index	Data Fields	Data Fields	
<h2>LCD_DrawPropTypeDef Struct Reference</h2> <p>STM32746G_DISCOVERY_LCD Exported Types</p>				

```
#include <stm32746g\_discovery\_lcd.h>
```

Data Fields

uint32_t	TextColor
uint32_t	BackColor
sFONT *	pFont

Detailed Description

Definition at line [73](#) of file [stm32746g_discovery_lcd.h](#).

Field Documentation

`uint32_t LCD_DrawPropTypeDef::BackColor`

Definition at line [76](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_ClearStringLine()`,
`BSP_LCD_GetBackColor()`, `BSP_LCD_LayerDefaultInit()`,
`BSP_LCD_LayerRgb565Init()`, and `BSP_LCD_SetBackColor()`.

`sFONT* LCD_DrawPropTypeDef::pFont`

Definition at line [77](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_DisplayChar()`,
`BSP_LCD_DisplayStringAt()`, `BSP_LCD_GetFont()`,
`BSP_LCD_LayerDefaultInit()`, `BSP_LCD_LayerRgb565Init()`,
`BSP_LCD_SetFont()`, and `DrawChar()`.

`uint32_t LCD_DrawPropTypeDef::TextColor`

Definition at line [75](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_ClearStringLine()`,
`BSP_LCD_GetTextColor()`, `BSP_LCD_LayerDefaultInit()`,
`BSP_LCD_LayerRgb565Init()`, and `BSP_LCD_SetTextColor()`.

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

- `stm32746g_discovery_lcd.h`

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Data Structures	Data Structure Index	Data Fields	Data Fields
Point Struct Reference STM32746G_DISCOVERY_LCD Exported Types			

```
#include <stm32746g_discovery_lcd.h>
```

Data Fields

int16_t X

int16_t Y

Detailed Description

Definition at line **80** of file [stm32746g_discovery_lcd.h](#).

Field Documentation

`int16_t Point::X`

Definition at line [82](#) of file [stm32746g_discovery_lcd.h](#).

Referenced by [BSP_LCD_DrawPolygon\(\)](#), and
[BSP_LCD_FillPolygon\(\)](#).

`int16_t Point::Y`

Definition at line [83](#) of file [stm32746g_discovery_lcd.h](#).

Referenced by [BSP_LCD_DrawPolygon\(\)](#), and
[BSP_LCD_FillPolygon\(\)](#).

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

- [stm32746g_discovery_lcd.h](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Data Structures	Data Structure Index	Data Fields	
All	Variables		

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

- BackColor : [LCD_DrawPropTypeDef](#)
- EraseSectorSize : [QSPI_Info](#)
- EraseSectorsNumber : [QSPI_Info](#)
- FlashSize : [QSPI_Info](#)
- gestureId : [TS_StateTypeDef](#)
- pFont : [LCD_DrawPropTypeDef](#)
- ProgPageSize : [QSPI_Info](#)
- ProgPagesNumber : [QSPI_Info](#)
- TextColor : [LCD_DrawPropTypeDef](#)
- touchArea : [TS_StateTypeDef](#)
- touchDetected : [TS_StateTypeDef](#)
- touchEventId : [TS_StateTypeDef](#)
- touchWeight : [TS_StateTypeDef](#)
- touchX : [TS_StateTypeDef](#)
- touchY : [TS_StateTypeDef](#)
- X : [Point](#)
- Y : [Point](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Data Structures	Data Structure Index	Data Fields	
All	Variables		

- BackColor : [LCD_DrawPropTypeDef](#)
- EraseSectorSize : [QSPI_Info](#)
- EraseSectorsNumber : [QSPI_Info](#)
- FlashSize : [QSPI_Info](#)
- gestureId : [TS_StateTypeDef](#)
- pFont : [LCD_DrawPropTypeDef](#)
- ProgPageSize : [QSPI_Info](#)
- ProgPagesNumber : [QSPI_Info](#)
- TextColor : [LCD_DrawPropTypeDef](#)
- touchArea : [TS_StateTypeDef](#)
- touchDetected : [TS_StateTypeDef](#)
- touchEventId : [TS_StateTypeDef](#)
- touchWeight : [TS_StateTypeDef](#)
- touchX : [TS_StateTypeDef](#)
- touchY : [TS_StateTypeDef](#)
- X : [Point](#)
- Y : [Point](#)

STM32746G-Discovery BSP User Manual

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

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

- - -

- `__DMAx_CLK_DISABLE` : [stm32746g_discovery_sram.h](#)
- `__DMAx_CLK_ENABLE` : [stm32746g_discovery_sram.h](#)
- `__DMAx_TxRx_CLK_ENABLE` : [stm32746g_discovery_sd.h](#)
- `__STM32746G_DISCO_BSP_VERSION` :
[stm32746g_discovery.c](#)
- `__STM32746G_DISCO_BSP_VERSION_MAIN` :
[stm32746g_discovery.c](#)
- `__STM32746G_DISCO_BSP_VERSION_RC` :
[stm32746g_discovery.c](#)
- `__STM32746G_DISCO_BSP_VERSION_SUB1` :
[stm32746g_discovery.c](#)
- `__STM32746G_DISCO_BSP_VERSION_SUB2` :
[stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

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

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

- a -

- ABS : [stm32746g_discovery_lcd.c](#)
- ActiveLayer : [stm32746g_discovery_lcd.c](#)
- audio_drv : [stm32746g_discovery_audio.c](#)
- AUDIO_ERROR : [stm32746g_discovery_audio.h](#)
- AUDIO_I2C_ADDRESS : [stm32746g_discovery.h](#)
- AUDIO_IN_INT_GPIO_ENABLE :
[stm32746g_discovery_audio.h](#)
- AUDIO_IN_INT_GPIO_PIN : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_INT_GPIO_PORT : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_INT_IRQ : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_INT_IRQHandler : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_IRQ_PREPRIO : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx_CLK_DISABLE :
[stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx_CLK_ENABLE :
[stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx_DMAX_CHANNEL :
[stm32746g_discovery_audio.h](#)

- AUDIO_IN_SAIx_DMAx_CLK_ENABLE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_DMAx_IRQ : [**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_DMAx_IRQHandler :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_DMAx_MEM_DATA_SIZE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_DMAx_PERIPH_DATA_SIZE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_DMAx_STREAM :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_SD_AF : [**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_SD_ENABLE : [**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_SD_GPIO_PORT :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_SD_PIN : [**stm32746g_discovery_audio.h**](#)
- AUDIO_IO_DelInit() : [**stm32746g_discovery.c**](#)
- AUDIO_IO_Delay() : [**stm32746g_discovery.c**](#)
- AUDIO_IO_Init() : [**stm32746g_discovery.c**](#)
- AUDIO_IO_Read() : [**stm32746g_discovery.c**](#)
- AUDIO_IO_Write() : [**stm32746g_discovery.c**](#)
- AUDIO_OK : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_IRQ_PREPRIO : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_CLK_DISABLE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_CLK_ENABLE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_CHANNEL :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_CLK_ENABLE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_IRQ : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_IRQHandler :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_MEM_DATA_SIZE :
[**stm32746g_discovery_audio.h**](#)

- AUDIO_OUT_SAIx_DMAx_PERIPH_DATA_SIZE :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_DMAx_STREAM :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_FS_ENABLE :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_FS_GPIO_PORT :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_FS_PIN : [`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_FS_SD_MCLK_AF :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_MCLK_ENABLE :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_MCLK_GPIO_PORT :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_MCLK_PIN : [`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_SCK_AF : [`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_SCK_PIN : [`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_SCK_SD_ENABLE :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT :
[`stm32746g_discovery_audio.h`](#)
- AUDIO_OUT_SAIx_SD_PIN : [`stm32746g_discovery_audio.h`](#)
- AUDIO_TIMEOUT : [`stm32746g_discovery_audio.h`](#)
- AUDIODATA_SIZE : [`stm32746g_discovery_audio.h`](#)
- AudioInVolume : [`stm32746g_discovery_audio.c`](#) ,
[`stm32746g_discovery_audio.h`](#)

STM32746G-Discovery BSP User Manual

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

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

- b -

- `BSP_AUDIO_IN_DeInit()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_Error_CallBack()` : [stm32746g_discovery_audio.h](#) , [stm32746g_discovery_audio.c](#)
- `BSP_AUDIO_IN_HalfTransfer_CallBack()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_Init()` : [stm32746g_discovery_audio.h](#) , [stm32746g_discovery_audio.c](#)
- `BSP_AUDIO_IN_InitEx()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_MspDeInit()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_MsplInit()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_OUT_Init()` : [stm32746g_discovery_audio.h](#) , [stm32746g_discovery_audio.c](#)
- `BSP_AUDIO_IN_Pause()` : [stm32746g_discovery_audio.c](#) ,

stm32746g_discovery_audio.h

- **BSP_AUDIO_IN_Record()** : **stm32746g_discovery_audio.c** ,
stm32746g_discovery_audio.h
- **BSP_AUDIO_IN_Resume()** : **stm32746g_discovery_audio.c** ,
stm32746g_discovery_audio.h
- **BSP_AUDIO_IN_SetVolume()** : **stm32746g_discovery_audio.c** ,
stm32746g_discovery_audio.h
- **BSP_AUDIO_IN_Stop()** : **stm32746g_discovery_audio.c** ,
stm32746g_discovery_audio.h
- **BSP_AUDIO_IN_TransferComplete_CallBack()** :
stm32746g_discovery_audio.c ,
stm32746g_discovery_audio.h
- **BSP_AUDIO_OUT_ChangeBuffer()** :
stm32746g_discovery_audio.c
- **BSP_AUDIO_OUT_ClockConfig()** :
stm32746g_discovery_audio.c
- **BSP_AUDIO_OUT_DeInit()** : **stm32746g_discovery_audio.c**
- **BSP_AUDIO_OUT_Error_CallBack()** :
stm32746g_discovery_audio.c
- **BSP_AUDIO_OUT_HalfTransfer_CallBack()** :
stm32746g_discovery_audio.c
- **BSP_AUDIO_OUT_Init()** : **stm32746g_discovery_audio.c**
- **BSP_AUDIO_OUT_MspDeInit()** : **stm32746g_discovery_audio.c**
- **BSP_AUDIO_OUT_MspInit()** : **stm32746g_discovery_audio.c**
- **BSP_AUDIO_OUT_Pause()** : **stm32746g_discovery_audio.c**
- **BSP_AUDIO_OUT_Play()** : **stm32746g_discovery_audio.c**
- **BSP_AUDIO_OUT_Resume()** : **stm32746g_discovery_audio.c**
- **BSP_AUDIO_OUT_SetAudioFrameSlot()** :
stm32746g_discovery_audio.c
- **BSP_AUDIO_OUT_SetFrequency()** :
stm32746g_discovery_audio.c
- **BSP_AUDIO_OUT_SetMute()** : **stm32746g_discovery_audio.c**
- **BSP_AUDIO_OUT_SetOutputMode()** :
stm32746g_discovery_audio.c
- **BSP_AUDIO_OUT_SetVolume()** :
stm32746g_discovery_audio.c
- **BSP_AUDIO_OUT_Stop()** : **stm32746g_discovery_audio.c**

- `BSP_AUDIO_OUT_TransferComplete_CallBack()` :
`stm32746g_discovery_audio.c`
- `BSP_CAMERA_BlackWhiteConfig()` :
`stm32746g_discovery_camera.c`
- `BSP_CAMERA_ColorEffectConfig()` :
`stm32746g_discovery_camera.c`
- `BSP_CAMERA_ContinuousStart()` :
`stm32746g_discovery_camera.c`
- `BSP_CAMERA_ContrastBrightnessConfig()` :
`stm32746g_discovery_camera.c`
- `BSP_CAMERA_DelInit()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_DMA_IRQHandler` :
`stm32746g_discovery_camera.h`
- `BSP_CAMERA_ErrorCallback()` :
`stm32746g_discovery_camera.c`
- `BSP_CAMERA_FrameEventCallback()` :
`stm32746g_discovery_camera.c`
- `BSP_CAMERA_Init()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_IRQHandler` : `stm32746g_discovery_camera.h`
- `BSP_CAMERA_LineEventCallback()` :
`stm32746g_discovery_camera.c`
- `BSP_CAMERA_MspDelInit()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_MspInit()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_PwrDown()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_PwrUp()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_Resume()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_SnapshotStart()` :
`stm32746g_discovery_camera.c`
- `BSP_CAMERA_Stop()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_Suspend()` : `stm32746g_discovery_camera.c`
- `BSP_CAMERA_VsyncEventCallback()` :
`stm32746g_discovery_camera.c`
- `BSP_COM_DelInit()` : `stm32746g_discovery.c`
- `BSP_COM_Init()` : `stm32746g_discovery.c`
- `BSP_EEPROM_DelInit()` : `stm32746g_discovery_eeprom.c`
- `BSP_EEPROM_Init()` : `stm32746g_discovery_eeprom.c`
- `BSP_EEPROM_ReadBuffer()` : `stm32746g_discovery_eeprom.c`

- `BSP_EEPROM_TIMEOUT_UserCallback()` :
[`stm32746g_discovery_eeprom.c`](#)
- `BSP_EEPROM_WaitEepromStandbyState()` :
[`stm32746g_discovery_eeprom.c`](#)
- `BSP_EEPROM_WriteBuffer()` : [`stm32746g_discovery_eeprom.c`](#)
- `BSP_EEPROM_WritePage()` : [`stm32746g_discovery_eeprom.c`](#)
- `BSP_GetVersion()` : [`stm32746g_discovery.c`](#)
- `BSP_LCD_Clear()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_ClearStringLine()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_ClockConfig()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DelInit()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DisplayChar()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DisplayOff()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DisplayOn()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DisplayStringAt()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DisplayStringAtLine()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawBitmap()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawCircle()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawEllipse()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawHLine()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawLine()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawPixel()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawPolygon()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawRect()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_DrawVLine()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_FillCircle()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_FillEllipse()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_FillPolygon()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_FillRect()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_GetBackColor()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_GetFont()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_GetTextColor()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_GetXSize()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_GetYSize()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_Init()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_LayerDefaultInit()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_LayerRgb565Init()` : [`stm32746g_discovery_lcd.c`](#)

- `BSP_LCD_MspDeInit()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_MspInit()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_ReadPixel()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_Reload()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_ResetColorKeying()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_ResetColorKeying_NoReload()` :
`stm32746g_discovery_lcd.c`
- `BSP_LCD_SelectLayer()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetBackColor()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetColorKeying()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetColorKeying_NoReload()` :
`stm32746g_discovery_lcd.c`
- `BSP_LCD_SetFont()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetLayerAddress()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetLayerAddress_NoReload()` :
`stm32746g_discovery_lcd.c`
- `BSP_LCD_SetLayerVisible()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetLayerVisible_NoReload()` :
`stm32746g_discovery_lcd.c`
- `BSP_LCD_SetLayerWindow()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetLayerWindow_NoReload()` :
`stm32746g_discovery_lcd.c`
- `BSP_LCD_SetTextColor()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetTransparency()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetTransparency_NoReload()` :
`stm32746g_discovery_lcd.c`
- `BSP_LCD_SetXSize()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_SetYSize()` : `stm32746g_discovery_lcd.c`
- `BSP_LED_DeInit()` : `stm32746g_discovery.c`
- `BSP_LED_Init()` : `stm32746g_discovery.c`
- `BSP_LED_Off()` : `stm32746g_discovery.c`
- `BSP_LED_On()` : `stm32746g_discovery.c`
- `BSP_LED_Toggle()` : `stm32746g_discovery.c`
- `BSP_PB_DeInit()` : `stm32746g_discovery.c`
- `BSP_PB_GetState()` : `stm32746g_discovery.c`
- `BSP_PB_Init()` : `stm32746g_discovery.c`
- `BSP_QSPI_DeInit()` : `stm32746g_discovery_qspi.c`

- `BSP_QSPI_EnableMemoryMappedMode()` :
`stm32746g_discovery_qspi.c`
- `BSP_QSPI_Erase_Block()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_Erase_Chip()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_GetInfo()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_GetStatus()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_Init()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_MemoryMappedMode` :
`stm32746g_discovery_qspi.h`
- `BSP_QSPI_MspDeInit()` : `stm32746g_discovery_qspi.h` ,
`stm32746g_discovery_qspi.c`
- `BSP_QSPI_MspInit()` : `stm32746g_discovery_qspi.h` ,
`stm32746g_discovery_qspi.c`
- `BSP_QSPI_Read()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_Write()` : `stm32746g_discovery_qspi.c`
- `BSP_SD_AbortCallback()` : `stm32746g_discovery_sd.c`
- `BSP_SD_CardInfo` : `stm32746g_discovery_sd.h`
- `BSP_SD_DeInit()` : `stm32746g_discovery_sd.c`
- `BSP_SD_Detect_MspInit()` : `stm32746g_discovery_sd.c`
- `BSP_SD_Erase()` : `stm32746g_discovery_sd.c`
- `BSP_SD_GetCardInfo()` : `stm32746g_discovery_sd.c`
- `BSP_SD_GetCardState()` : `stm32746g_discovery_sd.c`
- `BSP_SD_Init()` : `stm32746g_discovery_sd.c`
- `BSP_SD_IsDetected()` : `stm32746g_discovery_sd.c`
- `BSP_SD_ITConfig()` : `stm32746g_discovery_sd.c`
- `BSP_SD_MspDeInit()` : `stm32746g_discovery_sd.c`
- `BSP_SD_MspInit()` : `stm32746g_discovery_sd.c`
- `BSP_SD_ReadBlocks()` : `stm32746g_discovery_sd.c`
- `BSP_SD_ReadBlocks_DMA()` : `stm32746g_discovery_sd.c`
- `BSP_SD_ReadCpltCallback()` : `stm32746g_discovery_sd.c`
- `BSP_SD_WriteBlocks()` : `stm32746g_discovery_sd.c`
- `BSP_SD_WriteBlocks_DMA()` : `stm32746g_discovery_sd.c`
- `BSP_SD_WriteCpltCallback()` : `stm32746g_discovery_sd.c`
- `BSP_SDMMC_DMA_Rx_IRQHandler` :
`stm32746g_discovery_sd.h`
- `BSP_SDMMC_DMA_Tx_IRQHandler` :
`stm32746g_discovery_sd.h`

- BSP_SDMMC_IRQHandler : [stm32746g_discovery_sd.h](#)
- BSP_SDRAM_DeInit() : [stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_DMA_IRQHandler :

[stm32746g_discovery_sdram.h](#)
- BSP_SDRAM_Init() : [stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_Initialization_Sequence() :

[stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_MspDeInit() : [stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_MspInit() : [stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_ReadData() : [stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_ReadData_DMA() :

[stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_SendCmd() : [stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_WriteData() : [stm32746g_discovery_sdram.c](#)
- BSP_SDRAM_WriteData_DMA() :

[stm32746g_discovery_sdram.c](#)
- BSP_TS_DeInit() : [stm32746g_discovery_ts.c](#)
- BSP_TS_Get_GestureId() : [stm32746g_discovery_ts.c](#)
- BSP_TS_GetState() : [stm32746g_discovery_ts.c](#)
- BSP_TS_Init() : [stm32746g_discovery_ts.c](#)
- BSP_TS_ITClear() : [stm32746g_discovery_ts.c](#)
- BSP_TS_ITConfig() : [stm32746g_discovery_ts.c](#)
- BSP_TS_ITGetStatus() : [stm32746g_discovery_ts.c](#)
- BSP_TS_ResetTouchData() : [stm32746g_discovery_ts.c](#) ,

[stm32746g_discovery_ts.h](#)
- BUTTON_IRQn : [stm32746g_discovery.c](#)
- BUTTON_KEY : [stm32746g_discovery.h](#)
- BUTTON_MODE_EXTI : [stm32746g_discovery.h](#)
- BUTTON_MODE_GPIO : [stm32746g_discovery.h](#)
- BUTTON_PIN : [stm32746g_discovery.c](#)
- BUTTON_PORT : [stm32746g_discovery.c](#)
- BUTTON_TAMPER : [stm32746g_discovery.h](#)
- Button_TypeDef : [stm32746g_discovery.h](#)
- BUTTON_WAKEUP : [stm32746g_discovery.h](#)
- ButtonMode_TypeDef : [stm32746g_discovery.h](#)
- BUTTONN : [stm32746g_discovery.h](#)
- BUTTONx_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)

- BUTTONx_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

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

- C -

- CAMERA_480x272_RES_X : [stm32746g_discovery_camera.c](#)
- CAMERA_480x272_RES_Y : [stm32746g_discovery_camera.c](#)
- CAMERA_Delay() : [stm32746g_discovery.c](#)
- camera_drv : [stm32746g_discovery_camera.c](#)
- CAMERA_ERROR : [stm32746g_discovery_camera.h](#)
- CAMERA_I2C_ADDRESS : [stm32746g_discovery.h](#)
- CAMERA_IO_Init() : [stm32746g_discovery.c](#)
- CAMERA_IO_Read() : [stm32746g_discovery.c](#)
- CAMERA_IO_Write() : [stm32746g_discovery.c](#)
- CAMERA_NOT_DETECTED : [stm32746g_discovery_camera.h](#)
- CAMERA_NOT_SUPPORTED :
[stm32746g_discovery_camera.h](#)
- CAMERA_OK : [stm32746g_discovery_camera.h](#)
- CAMERA_QQVGA_RES_X : [stm32746g_discovery_camera.c](#)
- CAMERA_QQVGA_RES_Y : [stm32746g_discovery_camera.c](#)
- CAMERA_QVGA_RES_X : [stm32746g_discovery_camera.c](#)
- CAMERA_QVGA_RES_Y : [stm32746g_discovery_camera.c](#)
- Camera_StatusTypeDef : [stm32746g_discovery_camera.h](#)
- CAMERA_TIMEOUT : [stm32746g_discovery_camera.h](#)

- CAMERA_VGA_RES_X : [stm32746g_discovery_camera.c](#)
- CAMERA_VGA_RES_Y : [stm32746g_discovery_camera.c](#)
- CameraCurrentResolution : [stm32746g_discovery_camera.c](#)
- CameraHwAddress : [stm32746g_discovery_camera.c](#)
- CENTER_MODE : [stm32746g_discovery_lcd.h](#)
- CODEC_AUDIOFRAME_SLOT_0123 :
[stm32746g_discovery_audio.h](#)
- CODEC_AUDIOFRAME_SLOT_02 :
[stm32746g_discovery_audio.h](#)
- CODEC_AUDIOFRAME_SLOT_13 :
[stm32746g_discovery_audio.h](#)
- CODEC_RESET_DELAY : [stm32746g_discovery_audio.h](#)
- COM1 : [stm32746g_discovery.h](#)
- COM2 : [stm32746g_discovery.h](#)
- COM_RX_AF : [stm32746g_discovery.c](#)
- COM_RX_PIN : [stm32746g_discovery.c](#)
- COM_RX_PORT : [stm32746g_discovery.c](#)
- COM_TX_AF : [stm32746g_discovery.c](#)
- COM_TX_PIN : [stm32746g_discovery.c](#)
- COM_TX_PORT : [stm32746g_discovery.c](#)
- COM_TypeDef : [stm32746g_discovery.h](#)
- COM_USART : [stm32746g_discovery.c](#)
- Command : [stm32746g_discovery_sram.c](#)
- COMn : [stm32746g_discovery.h](#)

STM32746G-Discovery BSP User Manual

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

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

- d -

- DEFAULT_AUDIO_IN_BIT_RESOLUTION :
[stm32746g_discovery_audio.h](#)
- DEFAULT_AUDIO_IN_CHANNEL_NBR :
[stm32746g_discovery_audio.h](#)
- DEFAULT_AUDIO_IN_FREQ : [stm32746g_discovery_audio.h](#)
- DEFAULT_AUDIO_IN_VOLUME :
[stm32746g_discovery_audio.h](#)
- DISCOVERY_AUDIO_DMAX_CLK_ENABLE :
[stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_CLK_ENABLE :
[stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_ER IRQn : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_EV IRQn : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_FORCE_RESET :
[stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_RELEASE_RESET :
[stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_SCL_PIN : [stm32746g_discovery.h](#)

- DISCOVERY_AUDIO_I2Cx_SCL_SDA_AF :
[**stm32746g_discovery.h**](#)
- DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_CLK_ENABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PORT :
[**stm32746g_discovery.h**](#)
- DISCOVERY_AUDIO_I2Cx_SDA_PIN : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1 : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_CLK_DISABLE : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_CLK_ENABLE : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_IRQn : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_RX_AF : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_RX_GPIO_CLK_DISABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_RX_GPIO_CLK_ENABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_RX_GPIO_PORT : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_RX_PIN : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_TX_AF : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_TX_GPIO_CLK_DISABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_TX_GPIO_CLK_ENABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_TX_GPIO_PORT : [**stm32746g_discovery.h**](#)
- DISCOVERY_COM1_TX_PIN : [**stm32746g_discovery.h**](#)
- DISCOVERY_COMx_CLK_DISABLE : [**stm32746g_discovery.h**](#)
- DISCOVERY_COMx_CLK_ENABLE : [**stm32746g_discovery.h**](#)
- DISCOVERY_COMx_RX_GPIO_CLK_DISABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_COMx_RX_GPIO_CLK_ENABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_COMx_TX_GPIO_CLK_DISABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_COMx_TX_GPIO_CLK_ENABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_DMAx_CLK_ENABLE :
[**stm32746g_discovery.h**](#)

- DISCOVERY_EXT_I2Cx : [stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_CLK_ENABLE :
[stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_ER IRQn : [stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_EV IRQn : [stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_FORCE_RESET :
[stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_RELEASE_RESET :
[stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_SCL_PIN : [stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_SCL_SDA_AF :
[stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_CLK_ENABLE :
[stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_PORT :
[stm32746g_discovery.h](#)
- DISCOVERY_EXT_I2Cx_SDA_PIN : [stm32746g_discovery.h](#)
- DMA_MAX : [stm32746g_discovery_audio.h](#)
- DMA_MAX_SZE : [stm32746g_discovery_audio.h](#)
- DrawChar() : [stm32746g_discovery_lcd.c](#)
- DrawProp : [stm32746g_discovery_lcd.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

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

- e -

- EEPROM_FAIL : [stm32746g_discovery_eeprom.h](#)
- EEPROM_I2C_ADDRESS_A01 : [stm32746g_discovery.h](#)
- EEPROM_I2C_ADDRESS_A02 : [stm32746g_discovery.h](#)
- EEPROM_IO_Init() : [stm32746g_discovery.c](#) , [stm32746g_discovery_eeprom.h](#)
- EEPROM_IO_IsDeviceReady() : [stm32746g_discovery.c](#) , [stm32746g_discovery_eeprom.h](#)
- EEPROM_IO_ReadData() : [stm32746g_discovery.c](#) , [stm32746g_discovery_eeprom.h](#)
- EEPROM_IO_WriteData() : [stm32746g_discovery.c](#) , [stm32746g_discovery_eeprom.h](#)
- EEPROM_MAX_SIZE : [stm32746g_discovery_eeprom.h](#)
- EEPROM_MAX_TRIALS : [stm32746g_discovery_eeprom.h](#)
- EEPROM_OK : [stm32746g_discovery_eeprom.h](#)
- EEPROM_PAGESIZE : [stm32746g_discovery_eeprom.h](#)
- EEPROM_TIMEOUT : [stm32746g_discovery_eeprom.h](#)
- EEPROMAddress : [stm32746g_discovery_eeprom.c](#)
- EEPROMDataRead : [stm32746g_discovery_eeprom.c](#)
- EEPROMDataWrite : [stm32746g_discovery_eeprom.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

The screenshot shows a navigation bar with tabs: Main Page, Modules, Data Structures, and Files (selected). Below the tabs, there's a 'Directories' section and a 'File List' tab. The main content area has several tabs: All, Functions, Variables, Typedefs, Enumerations, Enumerator, and Defines. At the bottom, there's a scrollable list of letters from a to w.

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

- f -

- FillTriangle() : [stm32746g_discovery_lcd.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

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

- g -

- GEST_ID_MOVE_DOWN : [stm32746g_discovery_ts.h](#)
- GEST_ID_MOVE_LEFT : [stm32746g_discovery_ts.h](#)
- GEST_ID_MOVE_RIGHT : [stm32746g_discovery_ts.h](#)
- GEST_ID_MOVE_UP : [stm32746g_discovery_ts.h](#)
- GEST_ID_NB_MAX : [stm32746g_discovery_ts.h](#)
- GEST_ID_NO_GESTURE : [stm32746g_discovery_ts.h](#)
- GEST_ID_ZOOM_IN : [stm32746g_discovery_ts.h](#)
- GEST_ID_ZOOM_OUT : [stm32746g_discovery_ts.h](#)
- GetSize() : [stm32746g_discovery_camera.c](#)
- GPIO_PIN : [stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

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

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

- h -

- HAL_DCMI_ErrorCallback() : [stm32746g_discovery_camera.c](#)
- HAL_DCMI_FrameEventCallback() : [stm32746g_discovery_camera.c](#)
- HAL_DCMI_LineEventCallback() : [stm32746g_discovery_camera.c](#)
- HAL_DCMI_VsyncEventCallback() : [stm32746g_discovery_camera.c](#)
- HAL_SAI_ErrorCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SAI_RxCpltCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SAI_RxHalfCpltCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SAI_TxCpltCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SAI_TxHalfCpltCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SD_AbortCallback() : [stm32746g_discovery_sd.c](#)
- HAL_SD_RxCpltCallback() : [stm32746g_discovery_sd.c](#)
- HAL_SD_TxCpltCallback() : [stm32746g_discovery_sd.c](#)
- haudio_in_sai : [stm32746g_discovery_audio.c](#)
- haudio_out_sai : [stm32746g_discovery_audio.c](#)
- haudio_tim : [stm32746g_discovery_audio.c](#)
- hDcmiHandler : [stm32746g_discovery_camera.c](#)

- hDma2dHandler : **stm32746g_discovery_lcd.c**
 - hI2cAudioHandler : **stm32746g_discovery.c**
 - hI2cExtHandler : **stm32746g_discovery.c**
 - hLtdcHandler : **stm32746g_discovery_lcd.c**
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

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

- i -

- I2cAddress : [stm32746g_discovery_ts.c](#)
- I2Cx_Error() : [stm32746g_discovery.c](#)
- I2Cx_Init() : [stm32746g_discovery.c](#)
- I2Cx_IsDeviceReady() : [stm32746g_discovery.c](#)
- I2Cx_MspInit() : [stm32746g_discovery.c](#)
- I2Cx_ReadMultiple() : [stm32746g_discovery.c](#)
- I2Cx_WriteMultiple() : [stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files																	
Directories																				
File List	Globals																			
All	Functions	Variables	Typedefs	Enumerations																
Defines																				
-	a	b	c	d	e	f	g	h	i	k	l	m	o	p	q	r	s	t	u	w

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

- k -

- KEY_BUTTON_EXTI_IRQn : [stm32746g_discovery.h](#)
- KEY_BUTTON_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- KEY_BUTTON_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- KEY_BUTTON_GPIO_PORT : [stm32746g_discovery.h](#)
- KEY_BUTTON_PIN : [stm32746g_discovery.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files																	
Directories																				
File List	Globals																			
All	Functions	Variables	Typedefs	Enumerations	Enumerator															
Defines																				
-	a	b	c	d	e	f	g	h	i	k	I	m	o	p	q	r	s	t	u	w

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

- I -

- LCD_BL_CTRL_GPIO_CLK_DISABLE :
[stm32746g_discovery_lcd.h](#)
- LCD_BL_CTRL_GPIO_CLK_ENABLE :
[stm32746g_discovery_lcd.h](#)
- LCD_BL_CTRL_GPIO_PORT : [stm32746g_discovery_lcd.h](#)
- LCD_BL_CTRL_PIN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_BLACK : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_BLUE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_BROWN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_CYAN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKBLUE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKCYAN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKGRAY : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKGREEN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKMAGENTA : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKRED : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKYELLOW : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_GRAY : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_GREEN : [stm32746g_discovery_lcd.h](#)

- LCD_COLOR_LIGHTBLUE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTCYAN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTGRAY : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTGREEN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTMAGENTA : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTRED : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTYELLOW : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_MAGENTA : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_ORANGE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_RED : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_TRANSPARENT : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_WHITE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_YELLOW : [stm32746g_discovery_lcd.h](#)
- LCD_DEFAULT_FONT : [stm32746g_discovery_lcd.h](#)
- LCD_DISP_GPIO_CLK_DISABLE : [stm32746g_discovery_lcd.h](#)
- LCD_DISP_GPIO_CLK_ENABLE : [stm32746g_discovery_lcd.h](#)
- LCD_DISP_GPIO_PORT : [stm32746g_discovery_lcd.h](#)
- LCD_DISP_PIN : [stm32746g_discovery_lcd.h](#)
- LCD_ERROR : [stm32746g_discovery_lcd.h](#)
- LCD_FB_START_ADDRESS : [stm32746g_discovery_lcd.h](#)
- LCD_I2C_ADDRESS : [stm32746g_discovery.h](#)
- LCD_LayerCfgTypeDef : [stm32746g_discovery_lcd.h](#)
- LCD_OK : [stm32746g_discovery_lcd.h](#)
- LCD_RELOAD_IMMEDIATE : [stm32746g_discovery_lcd.h](#)
- LCD_RELOAD_VERTICAL_BLANKING :
[stm32746g_discovery_lcd.h](#)
- LCD_TIMEOUT : [stm32746g_discovery_lcd.h](#)
- LED1 : [stm32746g_discovery.h](#)
- LED1_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- LED1_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- LED1_GPIO_PORT : [stm32746g_discovery.h](#)
- LED1_PIN : [stm32746g_discovery.h](#)
- LED_GREEN : [stm32746g_discovery.h](#)
- Led_TypeDef : [stm32746g_discovery.h](#)
- LEDn : [stm32746g_discovery.h](#)
- LEFT_MODE : [stm32746g_discovery_lcd.h](#)
- LL_ConvertLineToARGB8888() : [stm32746g_discovery_lcd.c](#)

- LL_FillBuffer() : [stm32746g_discovery_lcd.c](#)
 - LTDC_ACTIVE_LAYER : [stm32746g_discovery_lcd.h](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

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

- m -

- MAX_LAYER_NUMBER : [stm32746g_discovery_lcd.h](#)
- MSD_ERROR : [stm32746g_discovery_sd.h](#)
- MSD_ERROR_SD_NOT_PRESENT :
[stm32746g_discovery_sd.h](#)
- MSD_OK : [stm32746g_discovery_sd.h](#)

STM32746G-Discovery BSP User Manual

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

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

- o -

- OUTPUT_DEVICE_HEADPHONE1 :
[stm32746g_discovery_audio.h](#)
- OUTPUT_DEVICE_HEADPHONE2 :
[stm32746g_discovery_audio.h](#)

STM32746G-Discovery BSP User Manual

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

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

- p -

- POLY_X : [stm32746g_discovery_lcd.c](#)
- POLY_Y : [stm32746g_discovery_lcd.c](#)
- pPoint : [stm32746g_discovery_lcd.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files																	
Directories																				
File List	Globals																			
All	Functions	Variables	Typedefs	Enumerations																
Defines																				
-	a	b	c	d	e	f	g	h	i	k	l	m	o	p	q	r	s	t	u	w

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

- q -

- `QSPI_AutoPollingMemReady()` : [stm32746g_discovery_qspi.c](#)
- `QSPI_BUSY` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_DISABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CS_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CS_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CS_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D0_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D0_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D0_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D1_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D1_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D1_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D2_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D2_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D2_PIN` : [stm32746g_discovery_qspi.h](#)

- QSPI_D3_GPIO_CLK_ENABLE : [stm32746g_discovery_qspi.h](#)
 - QSPI_D3_GPIO_PORT : [stm32746g_discovery_qspi.h](#)
 - QSPI_D3_PIN : [stm32746g_discovery_qspi.h](#)
 - QSPI_DummyCyclesCfg() : [stm32746g_discovery_qspi.c](#)
 - QSPI_ERROR : [stm32746g_discovery_qspi.h](#)
 - QSPI_FLASH_SIZE : [stm32746g_discovery_qspi.h](#)
 - QSPI_FORCE_RESET : [stm32746g_discovery_qspi.h](#)
 - QSPI_NOT_SUPPORTED : [stm32746g_discovery_qspi.h](#)
 - QSPI_OK : [stm32746g_discovery_qspi.h](#)
 - QSPI_PAGE_SIZE : [stm32746g_discovery_qspi.h](#)
 - QSPI_RELEASE_RESET : [stm32746g_discovery_qspi.h](#)
 - QSPI_ResetMemory() : [stm32746g_discovery_qspi.c](#)
 - QSPI_SUSPENDED : [stm32746g_discovery_qspi.h](#)
 - QSPI_WriteEnable() : [stm32746g_discovery_qspi.c](#)
 - QSPIHandle : [stm32746g_discovery_qspi.c](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

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

- r -

- REFRESH_COUNT : [stm32746g_discovery_sram.h](#)
- RESOLUTION_R160x120 : [stm32746g_discovery_camera.h](#)
- RESOLUTION_R320x240 : [stm32746g_discovery_camera.h](#)
- RESOLUTION_R480x272 : [stm32746g_discovery_camera.h](#)
- RESOLUTION_R640x480 : [stm32746g_discovery_camera.h](#)
- RIGHT_MODE : [stm32746g_discovery_lcd.h](#)

STM32746G-Discovery BSP User Manual

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

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

- S -

- SAIx_In_DeInit() : [stm32746g_discovery_audio.c](#)
- SAIx_In_Init() : [stm32746g_discovery_audio.c](#)
- SAIx_Out_DeInit() : [stm32746g_discovery_audio.c](#)
- SAIx_Out_Init() : [stm32746g_discovery_audio.c](#)
- SD_DATATIMEOUT : [stm32746g_discovery_sd.h](#)
- SD_DETECT_EXTI_IRQn : [stm32746g_discovery.h](#)
- SD_DETECT_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- SD_DETECT_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- SD_DETECT_GPIO_PORT : [stm32746g_discovery.h](#)
- SD_DETECT_PIN : [stm32746g_discovery.h](#)
- SD_DetectIRQHandler : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Rx_CHANNEL : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Rx_IRQn : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Rx_STREAM : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Tx_CHANNEL : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Tx_IRQn : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Tx_STREAM : [stm32746g_discovery_sd.h](#)
- SD_NOT_PRESENT : [stm32746g_discovery_sd.h](#)
- SD_PRESENT : [stm32746g_discovery_sd.h](#)

- SD_TRANSFER_BUSY : [stm32746g_discovery_sd.h](#)
 - SD_TRANSFER_OK : [stm32746g_discovery_sd.h](#)
 - SDCLOCK_PERIOD : [stm32746g_discovery_sdram.h](#)
 - SDRAM_DEVICE_ADDR : [stm32746g_discovery_sdram.h](#)
 - SDRAM_DEVICE_SIZE : [stm32746g_discovery_sdram.h](#)
 - SDRAM_DMax_CHANNEL : [stm32746g_discovery_sdram.h](#)
 - SDRAM_DMax_IRQn : [stm32746g_discovery_sdram.h](#)
 - SDRAM_DMax_STREAM : [stm32746g_discovery_sdram.h](#)
 - SDRAM_ERROR : [stm32746g_discovery_sdram.h](#)
 - SDRAM_MEMORY_WIDTH : [stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_BURST_LENGTH_1 :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_BURST_LENGTH_2 :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_BURST_LENGTH_4 :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_BURST_LENGTH_8 :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_BURST_TYPE_INTERLEAVED :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_BURST_TYPE_SEQUENTIAL :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_CAS_LATENCY_2 :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_CAS_LATENCY_3 :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_OPERATING_MODE_STANDARD :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_WRITEBURST_MODE_PROGRAMMED :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_MODEREG_WRITEBURST_MODE_SINGLE :
[stm32746g_discovery_sdram.h](#)
 - SDRAM_OK : [stm32746g_discovery_sdram.h](#)
 - SDRAM_TIMEOUT : [stm32746g_discovery_sdram.h](#)
 - sdramHandle : [stm32746g_discovery_sdram.c](#)
 - SIGNALn : [stm32746g_discovery.h](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

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

- t -

- TAMPER_BUTTON_EXTI_IRQn : [stm32746g_discovery.h](#)
- TAMPER_BUTTON_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- TAMPER_BUTTON_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- TAMPER_BUTTON_GPIO_PORT : [stm32746g_discovery.h](#)
- TAMPER_BUTTON_PIN : [stm32746g_discovery.h](#)
- Text_AlignModeTypdef : [stm32746g_discovery_lcd.h](#)
- Timing : [stm32746g_discovery_sdram.c](#)
- TOUCH_EVENT_CONTACT : [stm32746g_discovery_ts.h](#)
- TOUCH_EVENT_LIFT_UP : [stm32746g_discovery_ts.h](#)
- TOUCH_EVENT_NB_MAX : [stm32746g_discovery_ts.h](#)
- TOUCH_EVENT_NO_EVT : [stm32746g_discovery_ts.h](#)
- TOUCH_EVENT_PRESS_DOWN : [stm32746g_discovery_ts.h](#)
- TS_DEVICE_NOT_FOUND : [stm32746g_discovery_ts.h](#)
- TS_ERROR : [stm32746g_discovery_ts.h](#)
- ts_event_string_tab : [stm32746g_discovery_ts.h](#)
- ts_gesture_id_string_tab : [stm32746g_discovery_ts.h](#)
- TS_GestureIdTypeDef : [stm32746g_discovery_ts.h](#)

- TS_I2C_ADDRESS : [stm32746g_discovery.h](#)
- TS_INT_EXTI_IRQn : [stm32746g_discovery.h](#)
- TS_INT_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- TS_INT_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- TS_INT_GPIO_PORT : [stm32746g_discovery.h](#)
- TS_INT_PIN : [stm32746g_discovery.h](#)
- TS_IO_Delay() : [stm32746g_discovery.c](#)
- TS_IO_Init() : [stm32746g_discovery.c](#)
- TS_IO_Read() : [stm32746g_discovery.c](#)
- TS_IO_Write() : [stm32746g_discovery.c](#)
- TS_IRQ_PENDING : [stm32746g_discovery_ts.h](#)
- TS_MAX_NB_TOUCH : [stm32746g_discovery_ts.h](#)
- TS_NO_IRQ_PENDING : [stm32746g_discovery_ts.h](#)
- TS_OK : [stm32746g_discovery_ts.h](#)
- TS_StatusTypeDef : [stm32746g_discovery_ts.h](#)
- TS_SWAP_NONE : [stm32746g_discovery_ts.h](#)
- TS_SWAP_X : [stm32746g_discovery_ts.h](#)
- TS_SWAP_XY : [stm32746g_discovery_ts.h](#)
- TS_SWAP_Y : [stm32746g_discovery_ts.h](#)
- TS_TIMEOUT : [stm32746g_discovery_ts.h](#)
- TS_TouchEventTypeDef : [stm32746g_discovery_ts.h](#)
- tsDriver : [stm32746g_discovery_ts.c](#)
- tsOrientation : [stm32746g_discovery_ts.c](#)
- tsXBoundary : [stm32746g_discovery_ts.c](#)
- tsYBoundary : [stm32746g_discovery_ts.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files																	
Directories																				
File List	Globals																			
All	Functions	Variables	Typedefs	Enumerations																
Defines																				
-	a	b	c	d	e	f	g	h	i	k	l	m	o	p	q	r	s	t	u	w

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

- u -

- uSdHandle : [stm32746g_discovery_sd.c](#)

STM32746G-Discovery BSP User Manual

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

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

- W -

- WAKEUP_BUTTON_EXTI_IRQn : [stm32746g_discovery.h](#)
- WAKEUP_BUTTON_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- WAKEUP_BUTTON_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- WAKEUP_BUTTON_GPIO_PORT : [stm32746g_discovery.h](#)
- WAKEUP_BUTTON_PIN : [stm32746g_discovery.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations								
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- a -

- AUDIO_IO_Delnit() : [stm32746g_discovery.c](#)
- AUDIO_IO_Delay() : [stm32746g_discovery.c](#)
- AUDIO_IO_Init() : [stm32746g_discovery.c](#)
- AUDIO_IO_Read() : [stm32746g_discovery.c](#)
- AUDIO_IO_Write() : [stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations								
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- b -

- `BSP_AUDIO_IN_DeInit()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_Error_CallBack()` : [stm32746g_discovery_audio.h](#) , [stm32746g_discovery_audio.c](#)
- `BSP_AUDIO_IN_HalfTransfer_CallBack()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_Init()` : [stm32746g_discovery_audio.h](#) , [stm32746g_discovery_audio.c](#)
- `BSP_AUDIO_IN_InitEx()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_MspDeInit()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_MspInit()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)
- `BSP_AUDIO_IN_OUT_Init()` : [stm32746g_discovery_audio.h](#) , [stm32746g_discovery_audio.c](#)
- `BSP_AUDIO_IN_Pause()` : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)

- `BSP_AUDIO_IN_Record()` : [`stm32746g_discovery_audio.c`](#) , [`stm32746g_discovery_audio.h`](#)
- `BSP_AUDIO_IN_Resume()` : [`stm32746g_discovery_audio.c`](#) , [`stm32746g_discovery_audio.h`](#)
- `BSP_AUDIO_IN_SetVolume()` : [`stm32746g_discovery_audio.c`](#) , [`stm32746g_discovery_audio.h`](#)
- `BSP_AUDIO_IN_Stop()` : [`stm32746g_discovery_audio.c`](#) , [`stm32746g_discovery_audio.h`](#)
- `BSP_AUDIO_IN_TransferComplete_CallBack()` :
[`stm32746g_discovery_audio.c`](#) ,
[`stm32746g_discovery_audio.h`](#)
- `BSP_AUDIO_OUT_ChangeBuffer()` :
[`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_ClockConfig()` :
[`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_DeInit()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_Error_CallBack()` :
[`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_HalfTransfer_CallBack()` :
[`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_Init()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_MspDeInit()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_MspInit()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_Pause()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_Play()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_Resume()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_SetAudioFrameSlot()` :
[`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_SetFrequency()` :
[`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_SetMute()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_SetOutputMode()` :
[`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_SetVolume()` :
[`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_Stop()` : [`stm32746g_discovery_audio.c`](#)
- `BSP_AUDIO_OUT_TransferComplete_CallBack()` :

stm32746g_discovery_audio.c

- **BSP_CAMERA_BlackWhiteConfig()** :
stm32746g_discovery_camera.c
- **BSP_CAMERA_ColorEffectConfig()** :
stm32746g_discovery_camera.c
- **BSP_CAMERA_ContinuousStart()** :
stm32746g_discovery_camera.c
- **BSP_CAMERA_ContrastBrightnessConfig()** :
stm32746g_discovery_camera.c
- **BSP_CAMERA_DelInit()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_ErrorCallback()** :
stm32746g_discovery_camera.c
- **BSP_CAMERA_FrameEventCallback()** :
stm32746g_discovery_camera.c
- **BSP_CAMERA_Init()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_LineEventCallback()** :
stm32746g_discovery_camera.c
- **BSP_CAMERA_MspDelInit()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_MspInit()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_PwrDown()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_PwrUp()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_Resume()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_SnapshotStart()** :
stm32746g_discovery_camera.c
- **BSP_CAMERA_Stop()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_Suspend()** : **stm32746g_discovery_camera.c**
- **BSP_CAMERA_VsyncEventCallback()** :
stm32746g_discovery_camera.c
- **BSP_COM_DelInit()** : **stm32746g_discovery.c**
- **BSP_COM_Init()** : **stm32746g_discovery.c**
- **BSP_EEPROM_DelInit()** : **stm32746g_discovery_eeprom.c**
- **BSP_EEPROM_Init()** : **stm32746g_discovery_eeprom.c**
- **BSP_EEPROM_ReadBuffer()** : **stm32746g_discovery_eeprom.c**
- **BSP_EEPROM_TIMEOUT_UserCallback()** :
stm32746g_discovery_eeprom.c
- **BSP_EEPROM_WaitEepromStandbyState()** :
stm32746g_discovery_eeprom.c

- `BSP_EEPROM_WriteBuffer()` : `stm32746g_discovery_eeprom.c`
- `BSP_EEPROM_WritePage()` : `stm32746g_discovery_eeprom.c`
- `BSP_GetVersion()` : `stm32746g_discovery.c`
- `BSP_LCD_Clear()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_ClearStringLine()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_ClockConfig()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DeInit()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DisplayChar()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DisplayOff()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DisplayOn()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DisplayStringAt()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DisplayStringAtLine()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawBitmap()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawCircle()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawEllipse()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawHLine()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawLine()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawPixel()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawPolygon()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawRect()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_DrawVLine()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_FillCircle()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_FillEllipse()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_FillPolygon()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_FillRect()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_GetBackColor()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_GetFont()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_GetTextColor()` : `stm32746g_discovery_lcd.c`
- `BSP_LCDGetXSize()` : `stm32746g_discovery_lcd.c`
- `BSP_LCDGetYSize()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_Init()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_LayerDefaultInit()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_LayerRgb565Init()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_MspDeInit()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_MspInit()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_ReadPixel()` : `stm32746g_discovery_lcd.c`
- `BSP_LCD_Reload()` : `stm32746g_discovery_lcd.c`

- `BSP_LCD_ResetColorKeying()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_ResetColorKeying_NoReload()` :
[`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SelectLayer()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetBackColor()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetColorKeying()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetColorKeying_NoReload()` :
[`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetFont()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetLayerAddress()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetLayerAddress_NoReload()` :
[`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetLayerVisible()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetLayerVisible_NoReload()` :
[`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetLayerWindow()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetLayerWindow_NoReload()` :
[`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetTextColor()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetTransparency()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetTransparency_NoReload()` :
[`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetXSize()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LCD_SetYSize()` : [`stm32746g_discovery_lcd.c`](#)
- `BSP_LED_DeInit()` : [`stm32746g_discovery.c`](#)
- `BSP_LED_Init()` : [`stm32746g_discovery.c`](#)
- `BSP_LED_Off()` : [`stm32746g_discovery.c`](#)
- `BSP_LED_On()` : [`stm32746g_discovery.c`](#)
- `BSP_LED_Toggle()` : [`stm32746g_discovery.c`](#)
- `BSP_PB_DeInit()` : [`stm32746g_discovery.c`](#)
- `BSP_PB_GetState()` : [`stm32746g_discovery.c`](#)
- `BSP_PB_Init()` : [`stm32746g_discovery.c`](#)
- `BSP_QSPI_DeInit()` : [`stm32746g_discovery_qspi.c`](#)
- `BSP_QSPI_EnableMemoryMappedMode()` :
[`stm32746g_discovery_qspi.c`](#)
- `BSP_QSPI_Erase_Block()` : [`stm32746g_discovery_qspi.c`](#)
- `BSP_QSPI_Erase_Chip()` : [`stm32746g_discovery_qspi.c`](#)

- `BSP_QSPI_GetInfo()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_GetStatus()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_Init()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_MspDeInit()` : `stm32746g_discovery_qspi.h` ,
`stm32746g_discovery_qspi.c`
- `BSP_QSPI_MspInit()` : `stm32746g_discovery_qspi.c` ,
`stm32746g_discovery_qspi.h`
- `BSP_QSPI_Read()` : `stm32746g_discovery_qspi.c`
- `BSP_QSPI_Write()` : `stm32746g_discovery_qspi.c`
- `BSP_SD_AbortCallback()` : `stm32746g_discovery_sd.c`
- `BSP_SD_DeInit()` : `stm32746g_discovery_sd.c`
- `BSP_SD_Detect_MspInit()` : `stm32746g_discovery_sd.c`
- `BSP_SD_Erase()` : `stm32746g_discovery_sd.c`
- `BSP_SD_GetCardInfo()` : `stm32746g_discovery_sd.c`
- `BSP_SD_GetCardState()` : `stm32746g_discovery_sd.c`
- `BSP_SD_Init()` : `stm32746g_discovery_sd.c`
- `BSP_SD_IsDetected()` : `stm32746g_discovery_sd.c`
- `BSP_SD_ITConfig()` : `stm32746g_discovery_sd.c`
- `BSP_SD_MspDeInit()` : `stm32746g_discovery_sd.c`
- `BSP_SD_MspInit()` : `stm32746g_discovery_sd.c`
- `BSP_SD_ReadBlocks()` : `stm32746g_discovery_sd.c`
- `BSP_SD_ReadBlocks_DMA()` : `stm32746g_discovery_sd.c`
- `BSP_SD_ReadCpltCallback()` : `stm32746g_discovery_sd.c`
- `BSP_SD_WriteBlocks()` : `stm32746g_discovery_sd.c`
- `BSP_SD_WriteBlocks_DMA()` : `stm32746g_discovery_sd.c`
- `BSP_SD_WriteCpltCallback()` : `stm32746g_discovery_sd.c`
- `BSP_SDRAM_DeInit()` : `stm32746g_discovery_sdram.c`
- `BSP_SDRAM_Init()` : `stm32746g_discovery_sdram.c`
- `BSP_SDRAM_Initialization_sequence()` :
`stm32746g_discovery_sdram.c`
- `BSP_SDRAM_MspDeInit()` : `stm32746g_discovery_sdram.c`
- `BSP_SDRAM_MspInit()` : `stm32746g_discovery_sdram.c`
- `BSP_SDRAM_ReadData()` : `stm32746g_discovery_sdram.c`
- `BSP_SDRAM_ReadData_DMA()` :
`stm32746g_discovery_sdram.c`
- `BSP_SDRAM_Sendcmd()` : `stm32746g_discovery_sdram.c`
- `BSP_SDRAM_WriteData()` : `stm32746g_discovery_sdram.c`

- `BSP_SDRAM_WriteData_DMA()` :
[`stm32746g_discovery_sdram.c`](#)
 - `BSP_TS_DeInit()` : [`stm32746g_discovery_ts.c`](#)
 - `BSP_TS_Get_GestureId()` : [`stm32746g_discovery_ts.c`](#)
 - `BSP_TS_GetState()` : [`stm32746g_discovery_ts.c`](#)
 - `BSP_TS_Init()` : [`stm32746g_discovery_ts.c`](#)
 - `BSP_TS_ITClear()` : [`stm32746g_discovery_ts.c`](#)
 - `BSP_TS_ITConfig()` : [`stm32746g_discovery_ts.c`](#)
 - `BSP_TS_ITGetStatus()` : [`stm32746g_discovery_ts.c`](#)
 - `BSP_TS_ResetTouchData()` : [`stm32746g_discovery_ts.c`](#) ,
[`stm32746g_discovery_ts.h`](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

- C -

- CAMERA_Delay() : [stm32746g_discovery.c](#)
- CAMERA_IO_Init() : [stm32746g_discovery.c](#)
- CAMERA_IO_Read() : [stm32746g_discovery.c](#)
- CAMERA_IO_Write() : [stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations	Enumerator							
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- d -

- DrawChar() : [stm32746g_discovery_lcd.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

- e -

- EEPROM_IO_Init() : [stm32746g_discovery.c](#) , [stm32746g_discovery_eeprom.h](#)
- EEPROM_IO_IsDeviceReady() : [stm32746g_discovery_eeprom.h](#) , [stm32746g_discovery.c](#)
- EEPROM_IO_ReadData() : [stm32746g_discovery.c](#) , [stm32746g_discovery_eeprom.h](#)
- EEPROM_IO_WriteData() : [stm32746g_discovery_eeprom.h](#) , [stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations								
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- f -

- FillTriangle() : [stm32746g_discovery_lcd.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations	Enumerator							
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- g -

- GetSize() : [stm32746g_discovery_camera.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations								
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- h -

- HAL_DCMI_ErrorCallback() : [stm32746g_discovery_camera.c](#)
- HAL_DCMI_FrameEventCallback() : [stm32746g_discovery_camera.c](#)
- HAL_DCMI_LineEventCallback() : [stm32746g_discovery_camera.c](#)
- HAL_DCMI_VsyncEventCallback() : [stm32746g_discovery_camera.c](#)
- HAL_SAI_ErrorCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SAI_RxCpltCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SAI_RxHalfCpltCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SAI_TxCpltCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SAI_TxHalfCpltCallback() : [stm32746g_discovery_audio.c](#)
- HAL_SD_AbortCallback() : [stm32746g_discovery_sd.c](#)
- HAL_SD_RxCpltCallback() : [stm32746g_discovery_sd.c](#)
- HAL_SD_TxCpltCallback() : [stm32746g_discovery_sd.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations	Enumerator							
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- i -

- I2Cx_Error() : [stm32746g_discovery.c](#)
- I2Cx_Init() : [stm32746g_discovery.c](#)
- I2Cx_IsDeviceReady() : [stm32746g_discovery.c](#)
- I2Cx_MspInit() : [stm32746g_discovery.c](#)
- I2Cx_ReadMultiple() : [stm32746g_discovery.c](#)
- I2Cx_WriteMultiple() : [stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations								
Defines												
a	b	c	d	e	f	g	h	i	I	q	s	t

- I -

- LL_ConvertLineToARGB8888() : [stm32746g_discovery_lcd.c](#)
- LL_FillBuffer() : [stm32746g_discovery_lcd.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations								
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- q -

- `QSPI_AutoPollingMemReady()` : [stm32746g_discovery_qspi.c](#)
- `QSPI_DummyCyclesCfg()` : [stm32746g_discovery_qspi.c](#)
- `QSPI_ResetMemory()` : [stm32746g_discovery_qspi.c](#)
- `QSPI_WriteEnable()` : [stm32746g_discovery_qspi.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations								
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- S -

- SAIx_In_DelInit() : [stm32746g_discovery_audio.c](#)
- SAIx_In_Init() : [stm32746g_discovery_audio.c](#)
- SAIx_Out_DelInit() : [stm32746g_discovery_audio.c](#)
- SAIx_Out_Init() : [stm32746g_discovery_audio.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files									
Directories												
File List	Globals											
All	Functions	Variables	Typedefs	Enumerations	Enumerator							
Defines												
a	b	c	d	e	f	g	h	i	j	q	s	t

- t -

- TS_IO_Delay() : [stm32746g_discovery.c](#)
- TS_IO_Init() : [stm32746g_discovery.c](#)
- TS_IO_Read() : [stm32746g_discovery.c](#)
- TS_IO_Write() : [stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

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

- a -

- ActiveLayer : [stm32746g_discovery_lcd.c](#)
- audio_drv : [stm32746g_discovery_audio.c](#)
- AudioInVolume : [stm32746g_discovery_audio.c](#) , [stm32746g_discovery_audio.h](#)

- b -

- BUTTON_IRQn : [stm32746g_discovery.c](#)
- BUTTON_PIN : [stm32746g_discovery.c](#)
- BUTTON_PORT : [stm32746g_discovery.c](#)

- c -

- camera_drv : [stm32746g_discovery_camera.c](#)
- CameraCurrentResolution : [stm32746g_discovery_camera.c](#)
- CameraHwAddress : [stm32746g_discovery_camera.c](#)
- COM_RX_AF : [stm32746g_discovery.c](#)
- COM_RX_PIN : [stm32746g_discovery.c](#)
- COM_RX_PORT : [stm32746g_discovery.c](#)
- COM_TX_AF : [stm32746g_discovery.c](#)

- COM_TX_PIN : [stm32746g_discovery.c](#)
- COM_TX_PORT : [stm32746g_discovery.c](#)
- COM_USART : [stm32746g_discovery.c](#)
- Command : [stm32746g_discovery_sram.c](#)

- d -

- DrawProp : [stm32746g_discovery_lcd.c](#)

- e -

- EEPROMAddress : [stm32746g_discovery_eeprom.c](#)
- EEPROMDataRead : [stm32746g_discovery_eeprom.c](#)
- EEPROMDataWrite : [stm32746g_discovery_eeprom.c](#)

- g -

- GPIO_PIN : [stm32746g_discovery.c](#)

- h -

- haudio_in_sai : [stm32746g_discovery_audio.c](#)
- haudio_out_sai : [stm32746g_discovery_audio.c](#)
- haudio_tim : [stm32746g_discovery_audio.c](#)
- hDcmiHandler : [stm32746g_discovery_camera.c](#)
- hDma2dHandler : [stm32746g_discovery_lcd.c](#)
- hI2cAudioHandler : [stm32746g_discovery.c](#)
- hI2cExtHandler : [stm32746g_discovery.c](#)
- hLcdcHandler : [stm32746g_discovery_lcd.c](#)

- i -

- I2cAddress : [stm32746g_discovery_ts.c](#)

- q -

- QSPIHandle : [stm32746g_discovery_qspi.c](#)

- s -

- sdramHandle : [stm32746g_discovery_sdram.c](#)

- t -

- Timing : [stm32746g_discovery_sdram.c](#)
- ts_event_string_tab : [stm32746g_discovery_ts.h](#)
- ts_gesture_id_string_tab : [stm32746g_discovery_ts.h](#)
- tsDriver : [stm32746g_discovery_ts.c](#)
- tsOrientation : [stm32746g_discovery_ts.c](#)
- tsXBoundary : [stm32746g_discovery_ts.c](#)
- tsYBoundary : [stm32746g_discovery_ts.c](#)

- u -

- uSdHandle : [stm32746g_discovery_sd.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

- pPoint : [stm32746g_discovery_lcd.h](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

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

- Button_TypeDef : [stm32746g_discovery.h](#)
- ButtonMode_TypeDef : [stm32746g_discovery.h](#)
- Camera_StatusTypeDef : [stm32746g_discovery_camera.h](#)
- COM_TypeDef : [stm32746g_discovery.h](#)
- Led_TypeDef : [stm32746g_discovery.h](#)
- Text_AlignModeTypdef : [stm32746g_discovery_lcd.h](#)
- TS_GestureIdTypeDef : [stm32746g_discovery_ts.h](#)
- TS_StatusTypeDef : [stm32746g_discovery_ts.h](#)
- TS_TouchEventTypeDef : [stm32746g_discovery_ts.h](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files		
Directories					
File List	Globals				
All	Functions	Variables	Typedefs	Enumerations	Enumerator
Defines					
b	c	g	l	r	t

- b -

- BUTTON_KEY : [stm32746g_discovery.h](#)
- BUTTON_MODE_EXTI : [stm32746g_discovery.h](#)
- BUTTON_MODE_GPIO : [stm32746g_discovery.h](#)
- BUTTON_TAMPER : [stm32746g_discovery.h](#)
- BUTTON_WAKEUP : [stm32746g_discovery.h](#)

- c -

- CAMERA_ERROR : [stm32746g_discovery_camera.h](#)
- CAMERA_NOT_DETECTED : [stm32746g_discovery_camera.h](#)
- CAMERA_NOT_SUPPORTED :
[stm32746g_discovery_camera.h](#)
- CAMERA_OK : [stm32746g_discovery_camera.h](#)
- CAMERA_TIMEOUT : [stm32746g_discovery_camera.h](#)
- CENTER_MODE : [stm32746g_discovery_lcd.h](#)
- COM1 : [stm32746g_discovery.h](#)
- COM2 : [stm32746g_discovery.h](#)

- g -

- GEST_ID_MOVE_DOWN : [stm32746g_discovery_ts.h](#)
- GEST_ID_MOVE_LEFT : [stm32746g_discovery_ts.h](#)
- GEST_ID_MOVE_RIGHT : [stm32746g_discovery_ts.h](#)
- GEST_ID_MOVE_UP : [stm32746g_discovery_ts.h](#)
- GEST_ID_NB_MAX : [stm32746g_discovery_ts.h](#)
- GEST_ID_NO_GESTURE : [stm32746g_discovery_ts.h](#)
- GEST_ID_ZOOM_IN : [stm32746g_discovery_ts.h](#)
- GEST_ID_ZOOM_OUT : [stm32746g_discovery_ts.h](#)

- l -

- LED1 : [stm32746g_discovery.h](#)
- LED_GREEN : [stm32746g_discovery.h](#)
- LEFT_MODE : [stm32746g_discovery_lcd.h](#)

- r -

- RIGHT_MODE : [stm32746g_discovery_lcd.h](#)

- t -

- TOUCH_EVENT_CONTACT : [stm32746g_discovery_ts.h](#)
- TOUCH_EVENT_LIFT_UP : [stm32746g_discovery_ts.h](#)
- TOUCH_EVENT_NB_MAX : [stm32746g_discovery_ts.h](#)
- TOUCH_EVENT_NO_EVT : [stm32746g_discovery_ts.h](#)
- TOUCH_EVENT_PRESS_DOWN : [stm32746g_discovery_ts.h](#)
- TS_DEVICE_NOT_FOUND : [stm32746g_discovery_ts.h](#)
- TS_ERROR : [stm32746g_discovery_ts.h](#)
- TS_OK : [stm32746g_discovery_ts.h](#)
- TS_TIMEOUT : [stm32746g_discovery_ts.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- __DMAx_CLK_DISABLE : [stm32746g_discovery_sdram.h](#)
- __DMAx_CLK_ENABLE : [stm32746g_discovery_sdram.h](#)
- __DMAx_TxRx_CLK_ENABLE : [stm32746g_discovery_sd.h](#)
- __STM32746G_DISCO_BSP_VERSION :
[stm32746g_discovery.c](#)
- __STM32746G_DISCO_BSP_VERSION_MAIN :
[stm32746g_discovery.c](#)
- __STM32746G_DISCO_BSP_VERSION_RC :
[stm32746g_discovery.c](#)
- __STM32746G_DISCO_BSP_VERSION_SUB1 :
[stm32746g_discovery.c](#)
- __STM32746G_DISCO_BSP_VERSION_SUB2 :
[stm32746g_discovery.c](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
_	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- a -

- ABS : [stm32746g_discovery_lcd.c](#)
- AUDIO_ERROR : [stm32746g_discovery_audio.h](#)
- AUDIO_I2C_ADDRESS : [stm32746g_discovery.h](#)
- AUDIO_IN_INT_GPIO_ENABLE :
[stm32746g_discovery_audio.h](#)
- AUDIO_IN_INT_GPIO_PIN : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_INT_GPIO_PORT : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_INT_IRQ : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_INT_IRQHandler : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_IRQ_PREPRIO : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx : [stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx_CLK_DISABLE :
[stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx_CLK_ENABLE :
[stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx_DMAX_CHANNEL :
[stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx_DMAX_CLK_ENABLE :
[stm32746g_discovery_audio.h](#)
- AUDIO_IN_SAIx_DMAX_IRQ : [stm32746g_discovery_audio.h](#)

- AUDIO_IN_SAIx_DMAx_IRQHandler :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_DMAx_MEM_DATA_SIZE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_DMAx_PERIPH_DATA_SIZE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_DMAx_STREAM :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_SD_AF : [**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_SD_ENABLE : [**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_SD_GPIO_PORT :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_IN_SAIx_SD_PIN : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OK : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_IRQ_PREPRIO : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_CLK_DISABLE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_CLK_ENABLE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_CHANNEL :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_CLK_ENABLE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_IRQHandler : [**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_IRQHandler :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_MEM_DATA_SIZE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_PERIPH_DATA_SIZE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_DMAx_STREAM :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_FS_ENABLE :
[**stm32746g_discovery_audio.h**](#)
- AUDIO_OUT_SAIx_FS_GPIO_PORT :
[**stm32746g_discovery_audio.h**](#)

- AUDIO_OUT_SAIx_FS_PIN : [stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_FS_SD_MCLK_AF :
[stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_MCLK_ENABLE :
[stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_MCLK_GPIO_PORT :
[stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_MCLK_PIN : [stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_SCK_AF : [stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_SCK_PIN : [stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_SCK_SD_ENABLE :
[stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT :
[stm32746g_discovery_audio.h](#)
 - AUDIO_OUT_SAIx_SD_PIN : [stm32746g_discovery_audio.h](#)
 - AUDIO_TIMEOUT : [stm32746g_discovery_audio.h](#)
 - AUDIODATA_SIZE : [stm32746g_discovery_audio.h](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- b -

- `BSP_CAMERA_DMA_IRQHandler` : [stm32746g_discovery_camera.h](#)
- `BSP_CAMERA_IRQHandler` : [stm32746g_discovery_camera.h](#)
- `BSP_QSPI_MemoryMappedMode` : [stm32746g_discovery_qspi.h](#)
- `BSP_SD_CardInfo` : [stm32746g_discovery_sd.h](#)
- `BSP_SDMMC_DMA_Rx_IRQHandler` : [stm32746g_discovery_sd.h](#)
- `BSP_SDMMC_DMA_Tx_IRQHandler` : [stm32746g_discovery_sd.h](#)
- `BSP_SDMMC_IRQHandler` : [stm32746g_discovery_sd.h](#)
- `BSP_SDRAM_DMA_IRQHandler` : [stm32746g_discovery_sdram.h](#)
- `BUTTONN` : [stm32746g_discovery.h](#)
- `BUTTONx_GPIO_CLK_DISABLE` : [stm32746g_discovery.h](#)
- `BUTTONx_GPIO_CLK_ENABLE` : [stm32746g_discovery.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- C -

- CAMERA_480x272_RES_X : [stm32746g_discovery_camera.c](#)
- CAMERA_480x272_RES_Y : [stm32746g_discovery_camera.c](#)
- CAMERA_I2C_ADDRESS : [stm32746g_discovery.h](#)
- CAMERA_QQVGA_RES_X : [stm32746g_discovery_camera.c](#)
- CAMERA_QQVGA_RES_Y : [stm32746g_discovery_camera.c](#)
- CAMERA_QVGA_RES_X : [stm32746g_discovery_camera.c](#)
- CAMERA_QVGA_RES_Y : [stm32746g_discovery_camera.c](#)
- CAMERA_VGA_RES_X : [stm32746g_discovery_camera.c](#)
- CAMERA_VGA_RES_Y : [stm32746g_discovery_camera.c](#)
- CODEC_AUDIOFRAME_SLOT_0123 :
[stm32746g_discovery_audio.h](#)
- CODEC_AUDIOFRAME_SLOT_02 :
[stm32746g_discovery_audio.h](#)
- CODEC_AUDIOFRAME_SLOT_13 :
[stm32746g_discovery_audio.h](#)
- CODEC_RESET_DELAY : [stm32746g_discovery_audio.h](#)
- COMn : [stm32746g_discovery.h](#)

BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- d -

- DEFAULT_AUDIO_IN_BIT_RESOLUTION : [stm32746g_discovery_audio.h](#)
- DEFAULT_AUDIO_IN_CHANNEL_NBR : [stm32746g_discovery_audio.h](#)
- DEFAULT_AUDIO_IN_FREQ : [stm32746g_discovery_audio.h](#)
- DEFAULT_AUDIO_IN_VOLUME : [stm32746g_discovery_audio.h](#)
- DISCOVERY_AUDIO_DMAX_CLK_ENABLE : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_CLK_ENABLE : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_ER IRQn : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_EV IRQn : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_FORCE_RESET : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_RELEASE_RESET : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_SCL_PIN : [stm32746g_discovery.h](#)
- DISCOVERY_AUDIO_I2Cx_SDA_AF :

stm32746g_discovery.h

- DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_CLK_ENABLE :
stm32746g_discovery.h
- DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PORT :
stm32746g_discovery.h
- DISCOVERY_AUDIO_I2Cx_SDA_PIN : **stm32746g_discovery.h**
- DISCOVERY_COM1 : **stm32746g_discovery.h**
- DISCOVERY_COM1_CLK_DISABLE : **stm32746g_discovery.h**
- DISCOVERY_COM1_CLK_ENABLE : **stm32746g_discovery.h**
- DISCOVERY_COM1_IRQn : **stm32746g_discovery.h**
- DISCOVERY_COM1_RX_AF : **stm32746g_discovery.h**
- DISCOVERY_COM1_RX_GPIO_CLK_DISABLE :
stm32746g_discovery.h
- DISCOVERY_COM1_RX_GPIO_CLK_ENABLE :
stm32746g_discovery.h
- DISCOVERY_COM1_RX_GPIO_PORT : **stm32746g_discovery.h**
- DISCOVERY_COM1_RX_PIN : **stm32746g_discovery.h**
- DISCOVERY_COM1_TX_AF : **stm32746g_discovery.h**
- DISCOVERY_COM1_TX_GPIO_CLK_DISABLE :
stm32746g_discovery.h
- DISCOVERY_COM1_TX_GPIO_CLK_ENABLE :
stm32746g_discovery.h
- DISCOVERY_COM1_TX_GPIO_PORT : **stm32746g_discovery.h**
- DISCOVERY_COM1_TX_PIN : **stm32746g_discovery.h**
- DISCOVERY_COMx_CLK_DISABLE : **stm32746g_discovery.h**
- DISCOVERY_COMx_CLK_ENABLE : **stm32746g_discovery.h**
- DISCOVERY_COMx_RX_GPIO_CLK_DISABLE :
stm32746g_discovery.h
- DISCOVERY_COMx_RX_GPIO_CLK_ENABLE :
stm32746g_discovery.h
- DISCOVERY_COMx_TX_GPIO_CLK_DISABLE :
stm32746g_discovery.h
- DISCOVERY_COMx_TX_GPIO_CLK_ENABLE :
stm32746g_discovery.h
- DISCOVERY_EXT_DMAx_CLK_ENABLE :
stm32746g_discovery.h
- DISCOVERY_EXT_I2Cx : **stm32746g_discovery.h**

- DISCOVERY_EXT_I2Cx_CLK_ENABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_ER IRQn : [**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_EV IRQn : [**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_FORCE_RESET :
[**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_RELEASE_RESET :
[**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_SCL_PIN : [**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_SCL_SDA_AF :
[**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_CLK_ENABLE :
[**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_PORT :
[**stm32746g_discovery.h**](#)
- DISCOVERY_EXT_I2Cx_SDA_PIN : [**stm32746g_discovery.h**](#)
- DMA_MAX : [**stm32746g_discovery_audio.h**](#)
- DMA_MAX_SZE : [**stm32746g_discovery_audio.h**](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [**doxygen**](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- e -

- EEPROM_FAIL : [stm32746g_discovery_eeprom.h](#)
- EEPROM_I2C_ADDRESS_A01 : [stm32746g_discovery.h](#)
- EEPROM_I2C_ADDRESS_A02 : [stm32746g_discovery.h](#)
- EEPROM_MAX_SIZE : [stm32746g_discovery_eeprom.h](#)
- EEPROM_MAX_TRIALS : [stm32746g_discovery_eeprom.h](#)
- EEPROM_OK : [stm32746g_discovery_eeprom.h](#)
- EEPROM_PAGESIZE : [stm32746g_discovery_eeprom.h](#)
- EEPROM_TIMEOUT : [stm32746g_discovery_eeprom.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- k -

- KEY_BUTTON_EXTI_IRQn : [stm32746g_discovery.h](#)
- KEY_BUTTON_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- KEY_BUTTON_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- KEY_BUTTON_GPIO_PORT : [stm32746g_discovery.h](#)
- KEY_BUTTON_PIN : [stm32746g_discovery.h](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	I	m	o	p	q	r	s	t	w

- I -

- LCD_BL_CTRL_GPIO_CLK_DISABLE : [stm32746g_discovery_lcd.h](#)
- LCD_BL_CTRL_GPIO_CLK_ENABLE : [stm32746g_discovery_lcd.h](#)
- LCD_BL_CTRL_GPIO_PORT : [stm32746g_discovery_lcd.h](#)
- LCD_BL_CTRL_PIN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_BLACK : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_BLUE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_BROWN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_CYAN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKBLUE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKCYAN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKGRAY : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKGREEN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKMAGENTA : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKRED : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_DARKYELLOW : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_GRAY : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_GREEN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTBLUE : [stm32746g_discovery_lcd.h](#)

- LCD_COLOR_LIGHTCYAN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTGRAY : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTGREEN : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTMAGENTA : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTRED : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_LIGHTYELLOW : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_MAGENTA : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_ORANGE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_RED : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_TRANSPARENT : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_WHITE : [stm32746g_discovery_lcd.h](#)
- LCD_COLOR_YELLOW : [stm32746g_discovery_lcd.h](#)
- LCD_DEFAULT_FONT : [stm32746g_discovery_lcd.h](#)
- LCD_DISP_GPIO_CLK_DISABLE : [stm32746g_discovery_lcd.h](#)
- LCD_DISP_GPIO_CLK_ENABLE : [stm32746g_discovery_lcd.h](#)
- LCD_DISP_GPIO_PORT : [stm32746g_discovery_lcd.h](#)
- LCD_DISP_PIN : [stm32746g_discovery_lcd.h](#)
- LCD_ERROR : [stm32746g_discovery_lcd.h](#)
- LCD_FB_START_ADDRESS : [stm32746g_discovery_lcd.h](#)
- LCD_I2C_ADDRESS : [stm32746g_discovery.h](#)
- LCD_LayerCfgTypeDef : [stm32746g_discovery_lcd.h](#)
- LCD_OK : [stm32746g_discovery_lcd.h](#)
- LCD_RELOAD_IMMEDIATE : [stm32746g_discovery_lcd.h](#)
- LCD_RELOAD_VERTICAL_BLANKING :
[stm32746g_discovery_lcd.h](#)
- LCD_TIMEOUT : [stm32746g_discovery_lcd.h](#)
- LED1_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- LED1_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- LED1_GPIO_PORT : [stm32746g_discovery.h](#)
- LED1_PIN : [stm32746g_discovery.h](#)
- LEDn : [stm32746g_discovery.h](#)
- LTDC_ACTIVE_LAYER : [stm32746g_discovery_lcd.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- m -

- MAX_LAYER_NUMBER : [stm32746g_discovery_lcd.h](#)
- MSD_ERROR : [stm32746g_discovery_sd.h](#)
- MSD_ERROR_SD_NOT_PRESENT :
[stm32746g_discovery_sd.h](#)
- MSD_OK : [stm32746g_discovery_sd.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- o -

- OUTPUT_DEVICE_HEADPHONE1 :
[stm32746g_discovery_audio.h](#)
- OUTPUT_DEVICE_HEADPHONE2 :
[stm32746g_discovery_audio.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- p -

- POLY_X : [stm32746g_discovery_lcd.c](#)
- POLY_Y : [stm32746g_discovery_lcd.c](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- q -

- `QSPI_BUSY` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_DISABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CLK_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CS_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CS_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_CS_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D0_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D0_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D0_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D1_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D1_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D1_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D2_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D2_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D2_PIN` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D3_GPIO_CLK_ENABLE` : [stm32746g_discovery_qspi.h](#)
- `QSPI_D3_GPIO_PORT` : [stm32746g_discovery_qspi.h](#)

- QSPI_D3_PIN : [stm32746g_discovery_qspi.h](#)
 - QSPI_ERROR : [stm32746g_discovery_qspi.h](#)
 - QSPI_FLASH_SIZE : [stm32746g_discovery_qspi.h](#)
 - QSPI_FORCE_RESET : [stm32746g_discovery_qspi.h](#)
 - QSPI_NOT_SUPPORTED : [stm32746g_discovery_qspi.h](#)
 - QSPI_OK : [stm32746g_discovery_qspi.h](#)
 - QSPI_PAGE_SIZE : [stm32746g_discovery_qspi.h](#)
 - QSPI_RELEASE_RESET : [stm32746g_discovery_qspi.h](#)
 - QSPI_SUSPENDED : [stm32746g_discovery_qspi.h](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- r -

- REFRESH_COUNT : [stm32746g_discovery_sdram.h](#)
- RESOLUTION_R160x120 : [stm32746g_discovery_camera.h](#)
- RESOLUTION_R320x240 : [stm32746g_discovery_camera.h](#)
- RESOLUTION_R480x272 : [stm32746g_discovery_camera.h](#)
- RESOLUTION_R640x480 : [stm32746g_discovery_camera.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- S -

- SD_DATATIMEOUT : [stm32746g_discovery_sd.h](#)
- SD_DETECT_EXTI_IRQn : [stm32746g_discovery.h](#)
- SD_DETECT_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- SD_DETECT_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- SD_DETECT_GPIO_PORT : [stm32746g_discovery.h](#)
- SD_DETECT_PIN : [stm32746g_discovery.h](#)
- SD_DetectIRQHandler : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Rx_CHANNEL : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Rx_IRQn : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Rx_STREAM : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Tx_CHANNEL : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Tx_IRQn : [stm32746g_discovery_sd.h](#)
- SD_DMAx_Tx_STREAM : [stm32746g_discovery_sd.h](#)
- SD_NOT_PRESENT : [stm32746g_discovery_sd.h](#)
- SD_PRESENT : [stm32746g_discovery_sd.h](#)
- SD_TRANSFER_BUSY : [stm32746g_discovery_sd.h](#)
- SD_TRANSFER_OK : [stm32746g_discovery_sd.h](#)
- SDCLOCK_PERIOD : [stm32746g_discovery_sdram.h](#)
- SDRAM_DEVICE_ADDR : [stm32746g_discovery_sdram.h](#)
- SDRAM_DEVICE_SIZE : [stm32746g_discovery_sdram.h](#)

- SDRAM_DMAX_CHANNEL : [stm32746g_discovery_sdram.h](#)
- SDRAM_DMAX_IRQn : [stm32746g_discovery_sdram.h](#)
- SDRAM_DMAX_STREAM : [stm32746g_discovery_sdram.h](#)
- SDRAM_ERROR : [stm32746g_discovery_sdram.h](#)
- SDRAM_MEMORY_WIDTH : [stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_BURST_LENGTH_1 :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_BURST_LENGTH_2 :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_BURST_LENGTH_4 :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_BURST_LENGTH_8 :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_BURST_TYPE_INTERLEAVED :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_BURST_TYPE_SEQUENTIAL :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_CAS_LATENCY_2 :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_CAS_LATENCY_3 :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_OPERATING_MODE_STANDARD :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_WRITEBURST_MODE_PROGRAMMED :
[stm32746g_discovery_sdram.h](#)
- SDRAM_MODEREG_WRITEBURST_MODE_SINGLE :
[stm32746g_discovery_sdram.h](#)
- SDRAM_OK : [stm32746g_discovery_sdram.h](#)
- SDRAM_TIMEOUT : [stm32746g_discovery_sdram.h](#)
- SIGNALn : [stm32746g_discovery.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- t -

- TAMPER_BUTTON_EXTI_IRQn : [stm32746g_discovery.h](#)
- TAMPER_BUTTON_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- TAMPER_BUTTON_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- TAMPER_BUTTON_GPIO_PORT : [stm32746g_discovery.h](#)
- TAMPER_BUTTON_PIN : [stm32746g_discovery.h](#)
- TS_I2C_ADDRESS : [stm32746g_discovery.h](#)
- TS_INT_EXTI_IRQn : [stm32746g_discovery.h](#)
- TS_INT_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- TS_INT_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- TS_INT_GPIO_PORT : [stm32746g_discovery.h](#)
- TS_INT_PIN : [stm32746g_discovery.h](#)
- TS_IRQ_PENDING : [stm32746g_discovery_ts.h](#)
- TS_MAX_NB_TOUCH : [stm32746g_discovery_ts.h](#)
- TS_NO_IRQ_PENDING : [stm32746g_discovery_ts.h](#)
- TS_SWAP_NONE : [stm32746g_discovery_ts.h](#)
- TS_SWAP_X : [stm32746g_discovery_ts.h](#)
- TS_SWAP_XY : [stm32746g_discovery_ts.h](#)
- TS_SWAP_Y : [stm32746g_discovery_ts.h](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files												
Directories															
File List	Globals														
All	Functions	Variables	Typedefs	Enumerations	Enumerator										
Defines															
-	a	b	c	d	e	k	l	m	o	p	q	r	s	t	w

- W -

- WAKEUP_BUTTON_EXTI_IRQn : [stm32746g_discovery.h](#)
- WAKEUP_BUTTON_GPIO_CLK_DISABLE : [stm32746g_discovery.h](#)
- WAKEUP_BUTTON_GPIO_CLK_ENABLE : [stm32746g_discovery.h](#)
- WAKEUP_BUTTON_GPIO_PORT : [stm32746g_discovery.h](#)
- WAKEUP_BUTTON_PIN : [stm32746g_discovery.h](#)

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Defines Functions Variables

stm32746g_discovery.c File Reference

This file provides a set of firmware functions to manage LEDs, push-buttons and COM ports available on STM32746G-Discovery board(MB1191) from STMicroelectronics. [More...](#)

```
#include "stm32746g_discovery.h"
```

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

Defines

```
#define __STM32746G_DISCO_BSP_VERSION_MAIN (0x02)
    STM32746G DISCOVERY BSP Driver version number
    V2.0.0.

#define __STM32746G_DISCO_BSP_VERSION_SUB1 (0x00)
#define __STM32746G_DISCO_BSP_VERSION_SUB2 (0x00)
#define __STM32746G_DISCO_BSP_VERSION_RC (0x00)
#define __STM32746G_DISCO_BSP_VERSION
```

Functions

static void	I2Cx_MsInit (I2C_HandleTypeDef *i2c_handler) Initializes I2C MSP.
static void	I2Cx_Init (I2C_HandleTypeDef *i2c_handler) Initializes I2C HAL.
static HAL_StatusTypeDef	I2Cx_ReadMultiple (I2C_HandleTypeDef *i2c_handler, uint8_t Addr, uint16_t Reg, uint16_t MemAddress, uint8_t *Buffer, uint16_t Length) Reads multiple data.
static HAL_StatusTypeDef	I2Cx_WriteMultiple (I2C_HandleTypeDef *i2c_handler, uint8_t Addr, uint16_t Reg, uint16_t MemAddress, uint8_t *Buffer, uint16_t Length) Writes a value in a register of the device through BUS in using DMA mode.
static HAL_StatusTypeDef	I2Cx_IsDeviceReady (I2C_HandleTypeDef *i2c_handler, uint16_t DevAddress, uint32_t Trials) Checks if target device is ready for communication.
static void	I2Cx_Error (I2C_HandleTypeDef *i2c_handler, uint8_t Addr) Manages error callback by re-initializing I2C.
void	AUDIO_IO_Init (void) Initializes Audio low level.
void	AUDIO_IO_DelInit (void) Deinitializes Audio low level.
void	AUDIO_IO_Write (uint8_t Addr, uint16_t Reg, uint16_t Value)

		Writes a single data.
uint16_t	AUDIO_IO_Read (uint8_t Addr, uint16_t Reg)	Reads a single data.
void	AUDIO_IO_Delay (uint32_t Delay)	AUDIO Codec delay.
void	TS_IO_Init (void)	Initializes Touchscreen low level.
void	TS_IO_Write (uint8_t Addr, uint8_t Reg, uint8_t Value)	Writes a single data.
uint8_t	TS_IO_Read (uint8_t Addr, uint8_t Reg)	Reads a single data.
void	TS_IO_Delay (uint32_t Delay)	TS delay.
void	CAMERA_IO_Init (void)	Initializes Camera low level.
void	CAMERA_Delay (uint32_t Delay)	Camera delay.
void	CAMERA_IO_Write (uint8_t Addr, uint8_t Reg, uint8_t Value)	Camera writes single data.
uint8_t	CAMERA_IO_Read (uint8_t Addr, uint8_t Reg)	Camera reads single data.
void	EEPROM_IO_Init (void)	Initializes peripherals used by the I2C EEPROM driver.
HAL_StatusTypeDef	EEPROM_IO_WriteData (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize)	Write data to I2C EEPROM driver in using DMA channel.

HAL_StatusTypeDef	EEPROM_IO_ReadData (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize) Read data from I2C EEPROM driver in using DMA channel.
HAL_StatusTypeDef	EEPROM_IO_IsDeviceReady (uint16_t DevAddress, uint32_t Trials) Checks if target device is ready for communication.
uint32_t	BSP_GetVersion (void) This method returns the STM32746G DISCOVERY BSP Driver revision.
void	BSP_LED_Init (Led_TypeDef Led) Configures LED on GPIO.
void	BSP_LED_DelInit (Led_TypeDef Led) DelInit 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_DelInit (Button_TypeDef Button) Push Button DelInit.
uint32_t	BSP_PB_GetState (Button_TypeDef Button) Returns the selected button state.
void	BSP_COM_Init (COM_TypeDef COM, UART_HandleTypeDef *huart) Configures COM port.

```
void BSP_COM_DeInit (COM_TypeDef COM,  
                      UART_HandleTypeDef *huart)  
DeInit COM port.
```

Variables

```
const uint32_t GPIO_PIN [LEDn] = {LED1_PIN}
GPIO_TypeDef * BUTTON_PORT [BUTTONn]
const uint16_t BUTTON_PIN [BUTTONn]
const uint16_t BUTTON IRQn [BUTTONn]
USART_TypeDef * COM_USART [COMn] =
{DISCOVERY_COM1}
GPIO_TypeDef * COM_TX_PORT [COMn] =
{DISCOVERY_COM1_TX_GPIO_PORT}
GPIO_TypeDef * COM_RX_PORT [COMn] =
{DISCOVERY_COM1_RX_GPIO_PORT}
const uint16_t COM_TX_PIN [COMn] =
{DISCOVERY_COM1_TX_PIN}
const uint16_t COM_RX_PIN [COMn] =
{DISCOVERY_COM1_RX_PIN}
const uint16_t COM_TX_AF [COMn] =
{DISCOVERY_COM1_TX_AF}
const uint16_t COM_RX_AF [COMn] =
{DISCOVERY_COM1_RX_AF}
static I2C_HandleTypeDef hI2cAudioHandler = {0}
static I2C_HandleTypeDef hI2cExtHandler = {0}
```

Detailed Description

This file provides a set of firmware functions to manage LEDs, push-buttons and COM ports available on STM32746G-Discovery board(MB1191) from STMicroelectronics.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	
Defines Enumerations Functions			
<h2>stm32746g_discovery.h File Reference</h2>			

This file contains definitions for STM32746G_DISCOVERY's LEDs, push-buttons and COM ports hardware resources. [More...](#)

```
#include "stm32f7xx_hal.h"
```

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

Defines

```
#define LEDn ((uint8_t)1)
#define LED1_GPIO_PORT GPIOI
#define LED1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()
#define LED1_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DISABLE()
#define LED1_PIN GPIO_PIN_1
#define BUTTONn ((uint8_t)3)
#define WAKEUP_BUTTON_PIN GPIO_PIN_11
Wakeup push-button.

#define WAKEUP_BUTTON_GPIO_PORT GPIOI
#define WAKEUP_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()
#define WAKEUP_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DISABLE()
#define WAKEUP_BUTTON_EXTI_IRQn EXTI15_10_IRQn
#define TAMPER_BUTTON_PIN GPIO_PIN_11
Tamper push-button.

#define TAMPER_BUTTON_GPIO_PORT GPIOI
#define TAMPER_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()
#define TAMPER_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DISABLE()
#define TAMPER_BUTTON_EXTI_IRQn EXTI15_10_IRQn
#define KEY_BUTTON_PIN GPIO_PIN_11
Key push-button.

#define KEY_BUTTON_GPIO_PORT GPIOI
#define KEY_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()
#define KEY_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DISABLE()
#define KEY_BUTTON_EXTI_IRQn EXTI15_10_IRQn
#define BUTTONx_GPIO_CLK_ENABLE(__INDEX__)
#define BUTTONx_GPIO_CLK_DISABLE(__INDEX__)
#define SIGNALn ((uint8_t)1)
#define SD_DETECT_PIN GPIO_PIN_13
SD-detect signal.

#define SD_DETECT_GPIO_PORT GPIOC
#define SD_DETECT_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC_CLK_ENABLE()
```

```

#define SD_DETECT_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC
#define SD_DETECT EXTI IRQn EXTI15_10_IRQHandler
#define TS_INT_PIN GPIO_PIN_13
Touch screen interrupt signal.

#define TS_INT_GPIO_PORT GPIOI
#define TS_INT_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()
#define TS_INT_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DISABLE()
#define TS_INT_EXTI_IRQn EXTI15_10_IRQHandler
#define COMn ((uint8_t)1)
#define DISCOVERY_COM1 USART1
Definition for COM port1, connected to USART1.

#define DISCOVERY_COM1_CLK_ENABLE() __HAL_RCC_USART1
#define DISCOVERY_COM1_CLK_DISABLE() __HAL_RCC_USART1
#define DISCOVERY_COM1_TX_PIN GPIO_PIN_9
#define DISCOVERY_COM1_TX_GPIO_PORT GPIOA
#define DISCOVERY_COM1_TX_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA
#define DISCOVERY_COM1_TX_GPIO_CLK_DISABLE() __HAL_RCC_GPIOA
#define DISCOVERY_COM1_TX_AF GPIO_AF7_USART1
#define DISCOVERY_COM1_RX_PIN GPIO_PIN_7
#define DISCOVERY_COM1_RX_GPIO_PORT GPIOB
#define DISCOVERY_COM1_RX_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB
#define DISCOVERY_COM1_RX_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB
#define DISCOVERY_COM1_RX_AF GPIO_AF7_USART1
#define DISCOVERY_COM1_IRQn USART1_IRQHandler
#define DISCOVERY_COMx_CLK_ENABLE(__INDEX__) do { if((__INDEX__ < 1) || (__INDEX__ > 15)) break; __HAL_RCC_GPIOC_CLK_ENABLE(); } while(0)
#define DISCOVERY_COMx_CLK_DISABLE(__INDEX__) (((__INDEX__ < 1) || (__INDEX__ > 15)) ? 0 : __HAL_RCC_GPIOC_CLK_DISABLE())
#define DISCOVERY_COMx_TX_GPIO_CLK_ENABLE(__INDEX__)
#define DISCOVERY_COM1_TX_GPIO_CLK_ENABLE(); } while(0)
#define DISCOVERY_COMx_TX_GPIO_CLK_DISABLE(__INDEX__)
#define DISCOVERY_COM1_TX_GPIO_CLK_DISABLE(); } while(0)
#define DISCOVERY_COMx_RX_GPIO_CLK_ENABLE(__INDEX__)
#define DISCOVERY_COM1_RX_GPIO_CLK_ENABLE(); } while(0)

```

```
#define DISCOVERY_COMx_RX_GPIO_CLK_DISABLE(__INDEX__)
DISCOVERY_COM1_RX_GPIO_CLK_DISABLE() : 0)
#define LCD_I2C_ADDRESS ((uint16_t)0x70)
#define CAMERA_I2C_ADDRESS ((uint16_t)0x60)
#define AUDIO_I2C_ADDRESS ((uint16_t)0x34)
#define EEPROM_I2C_ADDRESS_A01 ((uint16_t)0xA0)
#define EEPROM_I2C_ADDRESS_A02 ((uint16_t)0xA6)
#define TS_I2C_ADDRESS ((uint16_t)0x70)
#define DISCOVERY_AUDIO_I2Cx I2C3
#define DISCOVERY_AUDIO_I2Cx_CLK_ENABLE() __HAL_RCC_I
#define DISCOVERY_AUDIO_DMAX_CLK_ENABLE() __HAL_RCC_
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_CLK_ENABLE
#define DISCOVERY_AUDIO_I2Cx_FORCE_RESET() __HAL_RCC_
#define DISCOVERY_AUDIO_I2Cx_RELEASE_RESET() __HAL_RC
#define DISCOVERY_AUDIO_I2Cx_SCL_PIN GPIO_PIN_7
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PORT GPIOH
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_AF GPIO_AF4_I2C3
#define DISCOVERY_AUDIO_I2Cx_SDA_PIN GPIO_PIN_8
#define DISCOVERY_AUDIO_I2Cx_EV IRQn I2C3_EV_IRQn
#define DISCOVERY_AUDIO_I2Cx_ER IRQn I2C3_ER_IRQn
#define DISCOVERY_EXT_I2Cx I2C1
#define DISCOVERY_EXT_I2Cx_CLK_ENABLE() __HAL_RCC_I2C
#define DISCOVERY_EXT_DMAX_CLK_ENABLE() __HAL_RCC_DI
#define DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_CLK_ENABLE()
#define DISCOVERY_EXT_I2Cx_FORCE_RESET() __HAL_RCC_I2
#define DISCOVERY_EXT_I2Cx_RELEASE_RESET() __HAL_RCC_
#define DISCOVERY_EXT_I2Cx_SCL_PIN GPIO_PIN_8
#define DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_PORT GPIOB
#define DISCOVERY_EXT_I2Cx_SCL_SDA_AF GPIO_AF4_I2C1
#define DISCOVERY_EXT_I2Cx_SDA_PIN GPIO_PIN_9
#define DISCOVERY_EXT_I2Cx_EV IRQn I2C1_EV_IRQn
#define DISCOVERY_EXT_I2Cx_ER IRQn I2C1_ER_IRQn
```

Enumerations

```
enum  Led_TypeDef { LED1 = 0, LED_GREEN = LED1 }
enum  Button_TypeDef { BUTTON_WAKEUP = 0,
                      BUTTON_TAMPER = 1, BUTTON_KEY = 2 }
enum  ButtonMode_TypeDef { BUTTON_MODE_GPIO = 0,
                           BUTTON_MODE_EXTI = 1 }
enum  COM_TypeDef { COM1 = 0, COM2 = 1 }
```

Functions

uint32_t **BSP_GetVersion** (void)

This method returns the STM32746G DISCOVERY BSP Driver revision.

void **BSP_LED_Init** (**Led_TypeDef** Led)

Configures LED on 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.

void **BSP_COM_Init** (**COM_TypeDef** COM,

UART_HandleTypeDef *huart)

Configures COM port.

void **BSP_COM_DeInit** (**COM_TypeDef** COM,

UART_HandleTypeDef *huart)

DeInit COM port.

Detailed Description

This file contains definitions for STM32746G_DISCOVERY's LEDs, push-buttons and COM ports hardware resources.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Functions Variables

stm32746g_discovery_audio.c File Reference

This file provides the Audio driver for the STM32746G-Discovery board.
[More...](#)

```
#include "stm32746g_discovery_audio.h"
```

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

Functions

static void	SAIx_Out_Init (uint32_t AudioFreq)	Initializes the output Audio Codec audio interface (SAI).
static void	SAIx_Out_DelInit (void)	Deinitializes the output Audio Codec audio interface (SAI).
static void	SAIx_In_Init (uint32_t SaiOutMode, uint32_t SlotActive, uint32_t AudioFreq)	Initializes the input Audio Codec audio interface (SAI).
static void	SAIx_In_DelInit (void)	Deinitializes the output Audio Codec audio interface (SAI).
uint8_t	BSP_AUDIO_OUT_Init (uint16_t OutputDevice, uint8_t Volume, uint32_t AudioFreq)	Configures the audio peripherals.
uint8_t	BSP_AUDIO_OUT_Play (uint16_t *pBuffer, uint32_t Size)	Starts playing audio stream from a data buffer for a determined size.
void	BSP_AUDIO_OUT_ChangeBuffer (uint16_t *pData, uint16_t Size)	Sends n-Bytes on the SAI interface.
uint8_t	BSP_AUDIO_OUT_Pause (void)	This function Pauses the audio file stream.
uint8_t	BSP_AUDIO_OUT_Resume (void)	This function Resumes the audio file stream.
uint8_t	BSP_AUDIO_OUT_Stop (uint32_t Option)	Stops audio playing and Power down the Audio Codec.
uint8_t	BSP_AUDIO_OUT_SetVolume (uint8_t Volume)	Controls the current audio volume level.
uint8_t	BSP_AUDIO_OUT_SetMute (uint32_t Cmd)	Enables or disables the MUTE mode by software.

<code>uint8_t</code>	BSP_AUDIO_OUT_SetOutputMode (<code>uint8_t</code> Output)
	Switch dynamically (while audio file is played) the output target (speaker or headphone).
<code>void</code>	BSP_AUDIO_OUT_SetFrequency (<code>uint32_t</code> AudioFreq)
	Updates the audio frequency.
<code>void</code>	BSP_AUDIO_OUT_SetAudioFrameSlot (<code>uint32_t</code> AudioFrameSlot)
	Updates the Audio frame slot configuration.
<code>void</code>	BSP_AUDIO_OUT_DeInit (<code>void</code>)
	Deinit the audio peripherals.
<code>void</code>	HAL_SAI_TxCpltCallback (<code>SAI_HandleTypeDef</code> *hsai)
	Tx Transfer completed callbacks.
<code>void</code>	HAL_SAI_TxHalfCpltCallback (<code>SAI_HandleTypeDef</code> *hsai)
	Tx Half Transfer completed callbacks.
<code>void</code>	HAL_SAI_ErrorCallback (<code>SAI_HandleTypeDef</code> *hsai)
	SAI error callbacks.
<code>_weak void</code>	BSP_AUDIO_OUT_TransferComplete_CallBack (<code>void</code>)
	Manages the DMA full Transfer complete event.
<code>_weak void</code>	BSP_AUDIO_OUT_HalfTransfer_CallBack (<code>void</code>)
	Manages the DMA Half Transfer complete event.
<code>_weak void</code>	BSP_AUDIO_OUT_Error_CallBack (<code>void</code>)
	Manages the DMA FIFO error event.
<code>_weak void</code>	BSP_AUDIO_OUT_MspInit (<code>SAI_HandleTypeDef</code> *hsai, <code>void</code> *Params)
	Initializes BSP_AUDIO_OUT MSP.
<code>_weak void</code>	BSP_AUDIO_OUT_MspDeInit (<code>SAI_HandleTypeDef</code> *hsai, <code>void</code> *Params)
	Deinitializes SAI MSP.
<code>_weak void</code>	BSP_AUDIO_OUT_ClockConfig (<code>SAI_HandleTypeDef</code> *hsai, <code>uint32_t</code> AudioFreq, <code>void</code> *Params)
	Clock Config.

<code>uint8_t</code>	BSP_AUDIO_IN_Init (<code>uint32_t</code> AudioFreq, <code>uint32_t</code> BitRes, <code>uint32_t</code> ChnlNbr)
	Initializes wave recording.
<code>uint8_t</code>	BSP_AUDIO_IN_InitEx (<code>uint16_t</code> InputDevice, <code>uint32_t</code> AudioFreq, <code>uint32_t</code> BitRes, <code>uint32_t</code> ChnlNbr)
	Initializes wave recording.
<code>uint8_t</code>	BSP_AUDIO_IN_OUT_Init (<code>uint16_t</code> InputDevice, <code>uint16_t</code> OutputDevice, <code>uint32_t</code> AudioFreq, <code>uint32_t</code> BitRes, <code>uint32_t</code> ChnlNbr)
	Initializes wave recording and playback in parallel.
<code>uint8_t</code>	BSP_AUDIO_IN_Record (<code>uint16_t</code> *pbuf, <code>uint32_t</code> size)
	Starts audio recording.
<code>uint8_t</code>	BSP_AUDIO_IN_Stop (<code>uint32_t</code> Option)
	Stops audio recording.
<code>uint8_t</code>	BSP_AUDIO_IN_Pause (<code>void</code>)
	Pauses the audio file stream.
<code>uint8_t</code>	BSP_AUDIO_IN_Resume (<code>void</code>)
	Resumes the audio file stream.
<code>uint8_t</code>	BSP_AUDIO_IN_SetVolume (<code>uint8_t</code> Volume)
	Controls the audio in volume level.
<code>void</code>	BSP_AUDIO_IN_DeInit (<code>void</code>)
	Deinit the audio IN peripherals.
<code>void</code>	HAL_SAI_RxCpltCallback (<code>SAI_HandleTypeDef</code> *hsai)
	Rx Transfer completed callbacks.
<code>void</code>	HAL_SAI_RxHalfCpltCallback (<code>SAI_HandleTypeDef</code> *hsai)
	Rx Half Transfer completed callbacks.
<code>_weak void</code>	BSP_AUDIO_IN_TransferComplete_CallBack (<code>void</code>)
	User callback when record buffer is filled.
<code>_weak void</code>	BSP_AUDIO_IN_HalfTransfer_CallBack (<code>void</code>)
	Manages the DMA Half Transfer complete event.
<code>_weak void</code>	BSP_AUDIO_IN_Error_CallBack (<code>void</code>)
	Audio IN Error callback function.
	BSP_AUDIO_IN_MspInit (<code>SAI_HandleTypeDef</code> *hsai,

```
__weak void  void *Params)
    Initializes BSP_AUDIO_IN MSP.
```

```
__weak void  BSP_AUDIO_IN_MspDelInit (SAI_HandleTypeDefDef
*hsai, void *Params)
    DelInitializes BSP_AUDIO_IN MSP.
```

Variables

```
AUDIO_DrvTypeDef * audio_drv
SAI_HandleTypeDef haudio_out_sai = {0}
SAI_HandleTypeDef haudio_in_sai = {0}
TIM_HandleTypeDef haudio_tim
    uint16_t __IO AudioInVolume =
        DEFAULT_AUDIO_IN_VOLUME
```

Detailed Description

This file provides the Audio driver for the STM32746G-Discovery board.

Author:

MCD Application Team

Version:

v2.0.0

Date:

30-December-2016

How To use this driver:

+ This driver supports STM32F7xx devices on STM32746G-Discovery (MB1191) board.

+ Call the function `BSP_AUDIO_OUT_Init(`
 `Output`

Device: physical output mode (OUTPUT_DEVICE_SPEAKER,

OUTPUT_DEVICE_HEADPHONE or OUTPUT_DEVICE_BOTH)

Volume : Initial volume to be set (θ is min (mute), 100 is max (100%))

```
req    : Audio frequency in Hz (8000, 16000, 22  
500, 32000...)
```

this parameter is relative to the audio file/stream type.

)

This function configures all the hardware required for the audio application (code G, T2C, SAT).

GPIOs, DMA and interrupt if needed). This function returns AUDIO_OK if configuration is OK.

If the returned value is different from AUDIO_OK or the function is stuck then the communication with

the codec or the MFX has failed (try to un-plug the power or reset device in this case).

- OUTPUT_DEVICE_SPEAKER : only speaker will be set as output for the audio stream

- OUTPUT_DEVICE_HEADPHONE: only headphones will be set as output for the audio stream.

- OUTPUT_DEVICE_BOTH : both Speaker and Headphone are used as outputs for the audio stream

at the same time.

Note. On STM32746G-Discovery SAI_DMA is configured in CIRCULAR mode. Due to this the application

does NOT need to call BSP_AUDIO_OUT_T_ChangeBuffer() to assure streaming.

```
+ Call the function BSP_DISCOVERY_AUDIO  
_OUT_Play(
```

```
    pBuffer: pointer to the audio data file address  
    Size : size of the buffer to be sent in Bytes  
    )
```

to start playing (for the first time) from the audio file/stream.

+ Call the function BSP_AUDIO_OUT_Pause() to pause playing

+ Call the function BSP_AUDIO_OUT_Resume

e() to resume playing.

Note. After calling BSP_AUDIO_OUT_Pause() function for pause, only BSP_AUDIO_OUT_Resume() should be called

for resume (it is not allowed to call BSP_AUDIO_OUT_Play() in this case).

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

+ For each mode, you may need to implement the relative callback functions into your code.

The Callback functions are named AUDIO_OUT_XXX_CallBack() and only their prototypes are declared in

the stm32746g_discovery_audio.h file . (refer to the example for more details on the callbacks implementations)

+ To Stop playing, to modify the volume level, the frequency, the audio frame slot,

the device output mode the mute or the stop, use the functions: BSP_AUDIO_OUT_SetVolume(),

AUDIO_OUT_SetFrequency(), BSP_AUDIO_OUT_SetAudioFrameSlot(), BSP_AUDIO_OUT_SetOutputMode(),

BSP_AUDIO_OUT_SetMute() and BSP_AUDIO_OUT_Stop().

+ The driver API and the callback functions are at the end of the stm32746g_discovery_audio.h file.

Driver architecture:

+ This driver provides the High Audio Layer: consists of the function API exported in the stm32746g_discovery_audio.h file

```
(BSP_AUDIO_OUT_Init(), BSP_AUDIO_OUT_
Play() ...)

+ This driver provide also the Media Ac-
cess Layer (MAL): which consists of functions
allowing to access the media containing/
providing the audio file/stream. These
functions are also included as local functions
into
the stm32746g_discovery_audio_codec.c
file (SAIx_Out_Init() and SAIx_Out_DeInit(),
SAIx_In_Init() and SAIx_In_DeInit())
```

Known Limitations:

- 1- If the TDM Format used to play in parallel 2 audio Stream (the first Stream is configured in codec SLOT0 and second Stream in SLOT1) the Pause/Resume, volume and mute feature will control the both streams.
- 2- Parsing of audio file is not implemented (in order to determine audio file properties: Mono/Stereo, Data size, File size, Audio Frequency, Audio Data header size ...). The configuration is fixed for the given audio file.
- 3- Supports only Stereo audio streaming
- 4- Supports only 16-bits audio data size.

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_audio.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Defines Functions Variables

stm32746g_discovery_audio.h File Reference

This file contains the common defines and functions prototypes for the [stm32746g_discovery_audio.c](#) driver. [More...](#)

```
#include "../Components/wm8994/wm8994.h" #include  
"stm32746g_discovery.h"
```

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

Defines

```
#define CODEC_AUDIOFRAME_SLOT_0123 SAI_SLOTACTIVE_0 |  
SAI_SLOTACTIVE_2 | SAI_SLOTACTIVE_3  
#define CODEC_AUDIOFRAME_SLOT_02 SAI_SLOTACTIVE_0 | SAI_SLOTACTIVE_1  
#define CODEC_AUDIOFRAME_SLOT_13 SAI_SLOTACTIVE_1 | SAI_SLOTACTIVE_2  
#define AUDIO_OUT_SAIx SAI2_Block_A  
#define AUDIO_OUT_SAIx_CLK_ENABLE() __HAL_RCC_SAI2_CLK_ENABLE()  
#define AUDIO_OUT_SAIx_CLK_DISABLE() __HAL_RCC_SAI2_CLK_DISABLE()  
#define AUDIO_OUT_SAIx_SCK_AF GPIO_AF10_SAI2  
#define AUDIO_OUT_SAIx_FS_SD_MCLK_AF GPIO_AF10_SAI2  
#define AUDIO_OUT_SAIx_MCLK_ENABLE() __HAL_RCC_GPIOI_ENABLE()  
#define AUDIO_OUT_SAIx_MCLK_GPIO_PORT GPIOI  
#define AUDIO_OUT_SAIx_MCLK_PIN GPIO_PIN_4  
#define AUDIO_OUT_SAIx_SCK_SD_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()  
#define AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT GPIOI  
#define AUDIO_OUT_SAIx_SCK_PIN GPIO_PIN_5  
#define AUDIO_OUT_SAIx_SD_PIN GPIO_PIN_6  
#define AUDIO_OUT_SAIx_FS_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()  
#define AUDIO_OUT_SAIx_FS_GPIO_PORT GPIOI  
#define AUDIO_OUT_SAIx_FS_PIN GPIO_PIN_7  
#define AUDIO_OUT_SAIx_DMAx_CLK_ENABLE() __HAL_RCC_DMA2_CLK_ENABLE()  
#define AUDIO_OUT_SAIx_DMAx_STREAM DMA2_Stream4  
#define AUDIO_OUT_SAIx_DMAx_CHANNEL DMA_CHANNEL_3  
#define AUDIO_OUT_SAIx_DMAx_IRQ DMA2_Stream4_IRQn  
#define AUDIO_OUT_SAIx_DMAx_PERIPH_DATA_SIZE DMA_PDATA_SIZE  
#define AUDIO_OUT_SAIx_DMAx_MEM_DATA_SIZE DMA_MDATA_SIZE  
#define DMA_MAX_SZE ((uint16_t)0xFFFF)  
#define AUDIO_OUT_SAIx_DMAx_IRQHandler DMA2_Stream4_IRQHandler  
#define AUDIO_OUT_IRQ_PREPRIO ((uint32_t)0x0E) /* Select the priority level(0 is the highest) */  
#define AUDIO_IN_SAIx SAI2_Block_B  
#define AUDIO_IN_SAIx_CLK_ENABLE() __HAL_RCC_SAI2_CLK_ENABLE()
```

```
#define AUDIO_IN_SAIx_CLK_DISABLE() __HAL_RCC_SAI2_CLK_DISABLE()
#define AUDIO_IN_SAIx_SD_AF GPIO_AF10_SAI2
#define AUDIO_IN_SAIx_SD_ENABLE() __HAL_RCC_GPIOG_CLK_ENABLE()
#define AUDIO_IN_SAIx_SD_GPIO_PORT GPIOG
#define AUDIO_IN_SAIx_SD_PIN GPIO_PIN_10
#define AUDIO_IN_INT_GPIO_ENABLE() __HAL_RCC_GPIOH_CLK_ENABLE()
#define AUDIO_IN_INT_GPIO_PORT GPIOH
#define AUDIO_IN_INT_GPIO_PIN GPIO_PIN_15
#define AUDIO_IN_INT_IRQ EXTI15_10_IRQHandler
#define AUDIO_IN_SAIx_DMAx_CLK_ENABLE() __HAL_RCC_DMA2_CLK_ENABLE()
#define AUDIO_IN_SAIx_DMAx_STREAM DMA2_Stream7
#define AUDIO_IN_SAIx_DMAx_CHANNEL DMA_CHANNEL_0
#define AUDIO_IN_SAIx_DMAx_IRQ DMA2_Stream7_IRQHandler
#define AUDIO_IN_SAIx_DMAx_PERIPH_DATA_SIZE DMA_PDATAALIGN
#define AUDIO_IN_SAIx_DMAx_MEM_DATA_SIZE DMA_MDATAALIGN
#define AUDIO_IN_SAIx_DMAx_IRQHandler DMA2_Stream7_IRQHandler
#define AUDIO_IN_INT_IRQHandler EXTI15_10_IRQHandler
#define AUDIO_IN_IRQ_PREPRIO ((uint32_t)0x0F) /* Select the preemption priority (0 is the highest) */
#define AUDIODATA_SIZE ((uint16_t)2) /* 16-bits audio data size */
#define AUDIO_OK ((uint8_t)0)
#define AUDIO_ERROR ((uint8_t)1)
#define AUDIO_TIMEOUT ((uint8_t)2)
#define DEFAULT_AUDIO_IN_FREQ I2S_AUDIOFREQ_16K
#define DEFAULT_AUDIO_IN_BIT_RESOLUTION ((uint8_t)16)
#define DEFAULT_AUDIO_IN_CHANNEL_NBR ((uint8_t)2) /* Mono */
#define DEFAULT_AUDIO_IN_VOLUME ((uint16_t)64)
#define CODEC_RESET_DELAY ((uint8_t)5)
#define OUTPUT_DEVICE_HEADPHONE1 OUTPUT_DEVICE_HEADPHONE1
#define OUTPUT_DEVICE_HEADPHONE2 OUTPUT_DEVICE_SPEAKER
#define DMA_MAX(x) (((x) <= DMA_MAX_SZE)? (x):DMA_MAX_SZE
```

Functions

uint8_t	BSP_AUDIO_OUT_Init (uint16_t OutputDevice, uint8_t Volume, uint32_t AudioFreq)	Configures the audio peripherals.
uint8_t	BSP_AUDIO_OUT_Play (uint16_t *pBuffer, uint32_t Size)	Starts playing audio stream from a data buffer for a determined size.
void	BSP_AUDIO_OUT_ChangeBuffer (uint16_t *pData, uint16_t Size)	Sends n-Bytes on the SAI interface.
uint8_t	BSP_AUDIO_OUT_Pause (void)	This function Pauses the audio file stream.
uint8_t	BSP_AUDIO_OUT_Resume (void)	This function Resumes the audio file stream.
uint8_t	BSP_AUDIO_OUT_Stop (uint32_t Option)	Stops audio playing and Power down the Audio Codec.
uint8_t	BSP_AUDIO_OUT_SetVolume (uint8_t Volume)	Controls the current audio volume level.
void	BSP_AUDIO_OUT_SetFrequency (uint32_t AudioFreq)	Updates the audio frequency.
void	BSP_AUDIO_OUT_SetAudioFrameSlot (uint32_t AudioFrameSlot)	Updates the Audio frame slot configuration.
uint8_t	BSP_AUDIO_OUT_SetMute (uint32_t Cmd)	Enables or disables the MUTE mode by software.
uint8_t	BSP_AUDIO_OUT_SetOutputMode (uint8_t Output)	Switch dynamically (while audio file is played) the output target (speaker or headphone).
void	BSP_AUDIO_OUT_DeInit (void)	Deinit the audio peripherals.

__weak void	BSP_AUDIO_OUT_TransferComplete_CallBack (void)	Manages the DMA full Transfer complete event.
__weak void	BSP_AUDIO_OUT_HalfTransfer_CallBack (void)	Manages the DMA Half Transfer complete event.
__weak void	BSP_AUDIO_OUT_Error_CallBack (void)	Manages the DMA FIFO error event.
__weak void	BSP_AUDIO_OUT_ClockConfig (SAI_HandleTypeDefDef *hsai, uint32_t AudioFreq, void *Params)	Clock Config.
__weak void	BSP_AUDIO_OUT_MspInit (SAI_HandleTypeDefDef *hsai, void *Params)	Initializes BSP_AUDIO_OUT MSP.
__weak void	BSP_AUDIO_OUT_MspDeInit (SAI_HandleTypeDefDef *hsai, void *Params)	Deinitializes SAI MSP.
uint8_t	BSP_AUDIO_IN_Init (uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording.
uint8_t	BSP_AUDIO_IN_InitEx (uint16_t InputDevice, uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording.
uint8_t	BSP_AUDIO_IN_OUT_Init (uint16_t InputDevice, uint16_t OutputDevice, uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording and playback in parallel.
uint8_t	BSP_AUDIO_IN_Record (uint16_t *pData, uint32_t Size)	Starts audio recording.
uint8_t	BSP_AUDIO_IN_Stop (uint32_t Option)	Stops audio recording.
uint8_t	BSP_AUDIO_IN_Pause (void)	Pauses the audio file stream.
uint8_t	BSP_AUDIO_IN_Resume (void)	

Resumes the audio file stream.

uint8_t **BSP_AUDIO_IN_SetVolume** (uint8_t Volume)

Controls the audio in volume level.

void **BSP_AUDIO_IN_DeInit** (void)

Deinit the audio IN peripherals.

void **BSP_AUDIO_IN_TransferComplete_CallBack** (void)

User callback when record buffer is filled.

void **BSP_AUDIO_IN_HalfTransfer_CallBack** (void)

Manages the DMA Half Transfer complete event.

void **BSP_AUDIO_IN_Error_CallBack** (void)

Audio IN Error callback function.

void **BSP_AUDIO_IN_MspInit** (SAI_HandleTypeDef *hsai,

void *Params)

Initializes BSP_AUDIO_IN MSP.

void **BSP_AUDIO_IN_MspDeInit** (SAI_HandleTypeDef

*hsai, void *Params)

DeInitializes BSP_AUDIO_IN MSP.

Variables

`__IO uint16_t AudioInVolume`

Detailed Description

This file contains the common defines and functions prototypes for the [**stm32746g_discovery_audio.c**](#) driver.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_audio.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

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

stm32746g_discovery_camera.c File Reference

This file includes the driver for Camera modules mounted on STM32746G-Discovery board. [More...](#)

```
#include "stm32746g_discovery_camera.h" #include  
"stm32746g_discovery.h"
```

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

Defines

```
#define CAMERA_VGA_RES_X 640
#define CAMERA_VGA_RES_Y 480
#define CAMERA_480x272_RES_X 480
#define CAMERA_480x272_RES_Y 272
#define CAMERA_QVGA_RES_X 320
#define CAMERA_QVGA_RES_Y 240
#define CAMERA_QQVGA_RES_X 160
#define CAMERA_QQVGA_RES_Y 120
```

Functions

static uint32_t	GetSize (uint32_t resolution)	Get the capture size in pixels unit.
uint8_t	BSP_CAMERA_Init (uint32_t Resolution)	Initializes the camera.
uint8_t	BSP_CAMERA_DelInit (void)	DeInitializes the camera.
void	BSP_CAMERA_ContinuousStart (uint8_t *buff)	Starts the camera capture in continuous mode.
void	BSP_CAMERA_SnapshotStart (uint8_t *buff)	Starts the camera capture in snapshot mode.
void	BSP_CAMERA_Suspend (void)	Suspend the CAMERA capture.
void	BSP_CAMERA_Resume (void)	Resume the CAMERA capture.
uint8_t	BSP_CAMERA_Stop (void)	Stop the CAMERA capture.
void	BSP_CAMERA_PwrUp (void)	CAMERA power up.
void	BSP_CAMERA_PwrDown (void)	CAMERA power down.
void	BSP_CAMERA_ContrastBrightnessConfig (uint32_t contrast_level, uint32_t brightness_level)	Configures the camera contrast and brightness.
void	BSP_CAMERA_BlackWhiteConfig (uint32_t Mode)	Configures the camera white balance.
void	BSP_CAMERA_ColorEffectConfig (uint32_t Effect)	Configures the camera color effect.
__weak void	BSP_CAMERA_MspInit (DCMI_HandleTypeDefDef *hdcmi, void *Params)	Initializes the DCMI MSP.

__weak void	BSP_CAMERA_MspDeInit (DCMI_HandleTypeDefDef *hdcmi, void *Params) DeInitializes the DCMI MSP.
void	HAL_DCMI_LineEventCallback (DCMI_HandleTypeDefDef *hdcmi) Line event callback.
__weak void	BSP_CAMERA_LineEventCallback (void) Line Event callback.
void	HAL_DCMI_VsyncEventCallback (DCMI_HandleTypeDefDef *hdcmi) VSYNC event callback.
__weak void	BSP_CAMERA_VsyncEventCallback (void) VSYNC Event callback.
void	HAL_DCMI_FrameEventCallback (DCMI_HandleTypeDefDef *hdcmi) Frame event callback.
__weak void	BSP_CAMERA_FrameEventCallback (void) Frame Event callback.
void	HAL_DCMI_ErrorCallback (DCMI_HandleTypeDefDef *hdcmi) Error callback.
__weak void	BSP_CAMERA_ErrorCallback (void) Error callback.

Variables

DCMI_HandleTypeDef	hDcmiHandler
CAMERA_DrvTypeDef *	camera_drv
static uint32_t	CameraCurrentResolution
static uint32_t	CameraHwAddress

Detailed Description

This file includes the driver for Camera modules mounted on STM32746G-Discovery board.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

How to use this driver:

- This driver is used to drive the camera
- .
- The OV9655 component driver MUST be included with this driver.

Driver description:

- + Initialization steps:
 - o Initialize the camera using the BSP_CAMERA_Init() function.
 - o Start the camera capture/snapshot using the CAMERA_Start() function.
 - o Suspend, resume or stop the camera capture using the following functions:
 - BSP_CAMERA_Suspend()
 - BSP_CAMERA_Resume()
 - BSP_CAMERA_Stop()
- + Options
 - o Increase or decrease on the fly the brightness and/or contrast

using the following function:

- `BSP_CAMERA_ContrastBrightnessConfig`

o Add a special effect on the fly using the following functions:

- `BSP_CAMERA_BlackWhiteConfig()`
- `BSP_CAMERA_ColorEffectConfig()`

| **Attention:**

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_camera.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Defines Enumerations Functions

stm32746g_discovery_camera.h File Reference

This file contains the common defines and functions prototypes for the [stm32746g_discovery_camera.c](#) driver. [More...](#)

```
#include "../Components/ov9655/ov9655.h" #include  
"stm32746g_discovery.h"
```

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

Defines

```
#define RESOLUTION_R160x120 CAMERA_R160x120 /* QQVGA  
Resolution */  
#define RESOLUTION_R320x240 CAMERA_R320x240 /* QVGA  
Resolution */  
#define RESOLUTION_R480x272 CAMERA_R480x272 /* 480x272  
Resolution */  
#define RESOLUTION_R640x480 CAMERA_R640x480 /* VGA  
Resolution */  
#define BSP_CAMERA_IRQHandler DCMI_IRQHandler  
#define BSP_CAMERA_DMA_IRQHandler DMA2_Stream1_IRQHandler
```

Enumerations

```
enum Camera_StatusTypeDef {
    CAMERA_OK = 0x00, CAMERA_ERROR = 0x01,
    CAMERA_TIMEOUT = 0x02, CAMERA_NOT_DETECTED =
    0x03,
    CAMERA_NOT_SUPPORTED = 0x04
}
```

Camera State structures definition. [More...](#)

Functions

uint8_t **BSP_CAMERA_Init** (uint32_t Resolution)

Initializes the camera.

uint8_t **BSP_CAMERA_DelInit** (void)

Deinitializes the camera.

void **BSP_CAMERA_ContinuousStart** (uint8_t *buff)

Starts the camera capture in continuous mode.

void **BSP_CAMERA_SnapshotStart** (uint8_t *buff)

Starts the camera capture in snapshot mode.

void **BSP_CAMERA_Suspend** (void)

Suspend the CAMERA capture.

void **BSP_CAMERA_Resume** (void)

Resume the CAMERA capture.

uint8_t **BSP_CAMERA_Stop** (void)

Stop the CAMERA capture.

void **BSP_CAMERA_PwrUp** (void)

CAMERA power up.

void **BSP_CAMERA_PwrDown** (void)

CAMERA power down.

weak void **BSP_CAMERA_LineEventCallback** (void)

Line Event callback.

weak void **BSP_CAMERA_VsyncEventCallback** (void)

VSYNC Event callback.

weak void **BSP_CAMERA_FrameEventCallback** (void)

Frame Event callback.

weak void **BSP_CAMERA_ErrorCallback** (void)

Error callback.

void **BSP_CAMERA_ContrastBrightnessConfig** (uint32_t

contrast_level, uint32_t brightness_level)

Configures the camera contrast and brightness.

void **BSP_CAMERA_BlackWhiteConfig** (uint32_t Mode)

Configures the camera white balance.

void **BSP_CAMERA_ColorEffectConfig** (uint32_t Effect)
Configures the camera color effect.

weak void **BSP_CAMERA_MspInit** (DCMI_HandleTypeDefDef
*hdcmi, void *Params)
Initializes the DCMI MSP.

weak void **BSP_CAMERA_MspDeInit** (DCMI_HandleTypeDefDef
*hdcmi, void *Params)
DeInitializes the DCMI MSP.

Detailed Description

This file contains the common defines and functions prototypes for the [**stm32746g_discovery_camera.c**](#) driver.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_camera.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Functions Variables

stm32746g_discovery_eeprom.c File Reference

This file provides a set of functions needed to manage an I2C M24LR64 EEPROM memory. [More...](#)

```
#include "stm32746g_discovery_eeprom.h"
```

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

Functions

uint32_t	BSP_EEPROM_Init (void)	Initializes peripherals used by the I2C EEPROM driver.
uint8_t	BSP_EEPROM_DelInit (void)	DeInitializes the EEPROM.
uint32_t	BSP_EEPROM_ReadBuffer (uint8_t *pBuffer, uint16_t ReadAddr, uint16_t *NumByteToRead)	Reads a block of data from the EEPROM.
uint32_t	BSP_EEPROM_WritePage (uint8_t *pBuffer, uint16_t WriteAddr, uint8_t *NumByteToWrite)	Writes more than one byte to the EEPROM with a single WRITE cycle.
uint32_t	BSP_EEPROM_WriteBuffer (uint8_t *pBuffer, uint16_t WriteAddr, uint16_t NumByteToWrite)	Writes buffer of data to the I2C EEPROM.
uint32_t	BSP_EEPROM_WaitEepromStandbyState (void)	Wait for EEPROM Standby state.
__weak void	BSP_EEPROM_TIMEOUT_UserCallback (void)	Basic management of the timeout situation.

Variables

```
__IO uint16_t EEPROMAddress = 0  
__IO uint16_t EEPROMDataRead  
__IO uint8_t EEPROMDataWrite
```

Detailed Description

This file provides a set of functions needed to manage an I2C M24LR64 EEPROM memory.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

To be able to use this driver, the switch EE_M24LR64 must be defined in your toolchain compiler preprocessor

=====

=====

Notes:

- The I2C EEPROM memory (M24LR64) is available on separate daughter board ANT7-M24LR-A, which is not provided with the STM32746G_DISCOVERY board.

To use this driver you have to connect the ANT7-M24LR-A to CN3 connector of STM32746G_DISCOVERY board.

=====

=====

It implements a high level communication layer for read and write from/to this memory. The needed S

TM32F7xx hardware resources (I2C and GPIO) are defined in `stm32746g_discovery.h` file, and the initialization is performed in `EEPROM_IO_Init()` function declared in `stm32746g_discovery.c` file.

You can easily tailor this driver to any other development board, by just adapting the defines for hardware resources and `EEPROM_IO_Init()` function.

@note In this driver, basic read and write functions (`BSP_EEPROM_ReadBuffer()` and `BSP_EEPROM_WritePage()`) use DMA mode to perform the data transfer to/from EEPROM memory.

@note Regarding `BSP_EEPROM_WritePage()`, it is an optimized function to perform small write (less than 1 page) BUT the number of bytes (combined to write start address) must not

cross the EEPROM page boundary. This function can only writes into the boundaries of an EEPROM page.

This function doesn't check on boundaries condition (in this driver the function `BSP_EEPROM_WriteBuffer()` which calls `BSP_EEPROM_WritePage()` is responsible of checking on Page boundaries).

Pin assignment for M24		
LR64 EEPROM	STM32F7xx I2C Pins	
EEPROM	Pin	
E0(GND)	1 (0V)	
AC0	2	
AC1	3	
VSS	4 (0V)	
SDA	5	
SCL	6	
E1(GND)	7 (0V)	
VDD	8 (3.3V)	

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_eeprom.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Defines Functions

stm32746g_discovery_eeprom.h File Reference

This file contains all the functions prototypes for the [stm32746g_discovery_eeprom.c](#) firmware driver. [More...](#)

```
#include "stm32746g_discovery.h"
```

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

Defines

```
#define EEPROM_PAGESIZE ((uint8_t)4)
#define EEPROM_MAX_SIZE ((uint16_t)0x2000) /* 64Kbit */
#define EEPROM_MAX_TRIALS ((uint32_t)3000)
#define EEPROM_OK ((uint32_t)0)
#define EEPROM_FAIL ((uint32_t)1)
#define EEPROM_TIMEOUT ((uint32_t)2)
```

Functions

uint32_t	BSP_EEPROM_Init (void) Initializes peripherals used by the I2C EEPROM driver.
uint8_t	BSP_EEPROM_DelInit (void) DeInitializes the EEPROM.
uint32_t	BSP_EEPROM_ReadBuffer (uint8_t *pBuffer, uint16_t ReadAddr, uint16_t *NumByteToRead) Reads a block of data from the EEPROM.
uint32_t	BSP_EEPROM_WritePage (uint8_t *pBuffer, uint16_t WriteAddr, uint8_t *NumByteToWrite) Writes more than one byte to the EEPROM with a single WRITE cycle.
uint32_t	BSP_EEPROM_WriteBuffer (uint8_t *pBuffer, uint16_t WriteAddr, uint16_t NumByteToWrite) Writes buffer of data to the I2C EEPROM.
uint32_t	BSP_EEPROM_WaitEepromStandbyState (void) Wait for EEPROM Standby state.
__weak void	BSP_EEPROM_TIMEOUT_UserCallback (void) Basic management of the timeout situation.
void	EEPROM_IO_Init (void) Initializes peripherals used by the I2C EEPROM driver.
HAL_StatusTypeDef	EEPROM_IO_WriteData (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize) Write data to I2C EEPROM driver in using DMA channel.
HAL_StatusTypeDef	EEPROM_IO_ReadData (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize)

	Read data from I2C EEPROM driver in using DMA channel.
HAL_StatusTypeDef	EEPROM_IO_IsDeviceReady (uint16_t DevAddress, uint32_t Trials) Checks if target device is ready for communication.

Detailed Description

This file contains all the functions prototypes for the [**stm32746g_discovery_eeprom.c**](#) firmware driver.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_eeprom.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	
Defines Functions Variables			

stm32746g_discovery_lcd.c File Reference

This file includes the driver for Liquid Crystal Display (LCD) module mounted on STM32746G-Discovery board. [More...](#)

```
#include "stm32746g_discovery_lcd.h" #include  
"../../../../Utilities/Fonts/fonts.h"  
#include "../../../../Utilities/Fonts/font24.c"  
#include "../../../../Utilities/Fonts/font20.c"  
#include "../../../../Utilities/Fonts/font16.c"  
#include "../../../../Utilities/Fonts/font12.c"  
#include "../../../../Utilities/Fonts/font8.c"
```

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

Defines

```
#define POLY_X(Z) ((int32_t)((Points + Z)->X))
#define POLY_Y(Z) ((int32_t)((Points + Z)->Y))
#define ABS(X) ((X) > 0 ? (X) : -(X))
```

Functions

static void	DrawChar (uint16_t Xpos, uint16_t Ypos, const uint8_t *c)	Draws a character on LCD.
static void	FillTriangle (uint16_t x1, uint16_t x2, uint16_t x3, uint16_t y1, uint16_t y2, uint16_t y3)	Fills a triangle (between 3 points).
static void	LL_FillBuffer (uint32_t LayerIndex, void *pDst, uint32_t xSize, uint32_t ySize, uint32_t OffLine, uint32_t ColorIndex)	Fills a buffer.
static void	LL_ConvertLineToARGB8888 (void *pSrc, void *pDst, uint32_t xSize, uint32_t ColorMode)	Converts a line to an ARGB8888 pixel format.
uint8_t	BSP_LCD_Init (void)	Initializes the LCD.
uint8_t	BSP_LCD_DelInit (void)	DeInitializes the LCD.
uint32_t	BSP_LCD_GetXSize (void)	Gets the LCD X size.
uint32_t	BSP_LCD_GetYSize (void)	Gets the LCD Y size.
void	BSP_LCD_SetXSize (uint32_t imageWidthPixels)	Set the LCD X size.
void	BSP_LCD_SetYSize (uint32_t imageHeightPixels)	Set the LCD Y size.
void	BSP_LCD_LayerDefaultInit (uint16_t LayerIndex, uint32_t FB_Address)	Initializes the LCD layer in ARGB8888 format (32 bits per pixel).
void	BSP_LCD_LayerRgb565Init (uint16_t LayerIndex, uint32_t FB_Address)	

	Initializes the LCD layer in RGB565 format (16 bits per pixel).
void	BSP_LCD_SelectLayer (uint32_t LayerIndex) Selects the LCD Layer.
void	BSP_LCD_SetLayerVisible (uint32_t LayerIndex, FunctionalState State) Sets an LCD Layer visible.
void	BSP_LCD_SetLayerVisible_NoReload (uint32_t LayerIndex, FunctionalState State) Sets an LCD Layer visible without reloading.
void	BSP_LCD_SetTransparency (uint32_t LayerIndex, uint8_t Transparency) Configures the transparency.
void	BSP_LCD_SetTransparency_NoReload (uint32_t LayerIndex, uint8_t Transparency) Configures the transparency without reloading.
void	BSP_LCD_SetLayerAddress (uint32_t LayerIndex, uint32_t Address) Sets an LCD layer frame buffer address.
void	BSP_LCD_SetLayerAddress_NoReload (uint32_t LayerIndex, uint32_t Address) Sets an LCD layer frame buffer address without reloading.
void	BSP_LCD_SetLayerWindow (uint16_t LayerIndex, uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height) Sets display window.
void	BSP_LCD_SetLayerWindow_NoReload (uint16_t LayerIndex, uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height) Sets display window without reloading.
void	BSP_LCD_SetColorKeying (uint32_t LayerIndex, uint32_t RGBValue) Configures and sets the color keying.

void	BSP_LCD_SetColorKeying_NoReload (uint32_t LayerIndex, uint32_t RGBValue)	Configures and sets the color keying without reloading.
void	BSP_LCD_ResetColorKeying (uint32_t LayerIndex)	Disables the color keying.
void	BSP_LCD_ResetColorKeying_NoReload (uint32_t LayerIndex)	Disables the color keying without reloading.
void	BSP_LCD_Reload (uint32_t ReloadType)	Disables the color keying without reloading.
void	BSP_LCD_SetTextColor (uint32_t Color)	Sets the LCD text color.
uint32_t	BSP_LCD_GetTextColor (void)	Gets the LCD text color.
void	BSP_LCD_SetBackColor (uint32_t Color)	Sets the LCD background color.
uint32_t	BSP_LCD_GetBackColor (void)	Gets the LCD background color.
void	BSP_LCD_SetFont (sFONT *fonts)	Sets the LCD text font.
sFONT *	BSP_LCD_GetFont (void)	Gets the LCD text font.
uint32_t	BSP_LCD_ReadPixel (uint16_t Xpos, uint16_t Ypos)	Reads an LCD pixel.
void	BSP_LCD_Clear (uint32_t Color)	Clears the hole LCD.
void	BSP_LCD_ClearStringLine (uint32_t Line)	Clears the selected line.
void	BSP_LCD_DisplayChar (uint16_t Xpos, uint16_t Ypos, uint8_t Ascii)	Displays one character.
void	BSP_LCD_DisplayStringAt (uint16_t Xpos, uint16_t Ypos, uint8_t *Text, Text_AlignModeTypdef Mode)	Displays characters on the LCD.

void	BSP_LCD_DisplayStringAtLine (uint16_t Line, uint8_t *ptr)	Displays a maximum of 60 characters on the LCD.
void	BSP_LCD_DrawHLine (uint16_t Xpos, uint16_t Ypos, uint16_t Length)	Draws an horizontal line.
void	BSP_LCD_DrawVLine (uint16_t Xpos, uint16_t Ypos, uint16_t Length)	Draws a vertical line.
void	BSP_LCD_DrawLine (uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2)	Draws an uni-line (between two points).
void	BSP_LCD_DrawRect (uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height)	Draws a rectangle.
void	BSP_LCD_DrawCircle (uint16_t Xpos, uint16_t Ypos, uint16_t Radius)	Draws a circle.
void	BSP_LCD_DrawPolygon (pPoint Points, uint16_t PointCount)	Draws an poly-line (between many points).
void	BSP_LCD_DrawEllipse (int Xpos, int Ypos, int XRadius, int YRadius)	Draws an ellipse on LCD.
void	BSP_LCD_DrawPixel (uint16_t Xpos, uint16_t Ypos, uint32_t RGB_Code)	Draws a pixel on LCD.
void	BSP_LCD_DrawBitmap (uint32_t Xpos, uint32_t Ypos, uint8_t *pbmp)	Draws a bitmap picture loaded in the internal Flash in ARGB888 format (32 bits per pixel).
void	BSP_LCD_FillRect (uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height)	Draws a full rectangle.

void	BSP_LCD_FillCircle (uint16_t Xpos, uint16_t Ypos, uint16_t Radius)	Draws a full circle.
void	BSP_LCD_FillPolygon (pPoint Points, uint16_t PointCount)	Draws a full poly-line (between many points).
void	BSP_LCD_FillEllipse (int Xpos, int Ypos, int XRadius, int YRadius)	Draws a full ellipse.
void	BSP_LCD_DisplayOn (void)	Enables the display.
void	BSP_LCD_DisplayOff (void)	Disables the display.
<u>weak</u> void	BSP_LCD_MspInit (LTDC_HandleTypeDef *hltcd, void *Params)	Initializes the LTDC MSP.
<u>weak</u> void	BSP_LCD_MspDeInit (LTDC_HandleTypeDef *hltcd, void *Params)	DeInitializes BSP_LCD MSP.
<u>weak</u> void	BSP_LCD_ClockConfig (LTDC_HandleTypeDef *hltcd, void *Params)	Clock Config.

Variables

```
LTDC_HandleTypeDef hLtdcHandler  
static DMA2D_HandleTypeDef hDma2dHandler  
    static uint32_t ActiveLayer = 0  
static LCD_DrawPropTypeDef DrawProp [MAX_LAYER_NUMBER]
```

Detailed Description

This file includes the driver for Liquid Crystal Display (LCD) module mounted on STM32746G-Discovery board.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

1. How To use this driver:

-
- This driver is used to drive directly a n LCD TFT using the LTDC controller.
 - This driver uses timing and setting for RK043FN48H LCD.

2. Driver description:

-
- + Initialization steps:
 - o Initialize the LCD using the BSP_LCD_Init() function.

o Apply the Layer configuration using t he BSP_LCD_LayerDefaultInit() function.

o Select the LCD layer to be used using the BSP_LCD_SelectLayer() function.

o Enable the LCD display using the BSP_ LCD_DisplayOn() function.

+ Options

o Configure and enable the color keying functionality using the
BSP_LCD_SetColorKeying() function.

- o Modify in the fly the transparency and/or the frame buffer address using the following functions:
 - BSP_LCD_SetTransparency()
 - BSP_LCD_SetLayerAddress()
- + Display on LCD
 - o Clear the hole LCD using BSP_LCD_Clear() function or only one specified string line using the BSP_LCD_ClearStringLine() function.
 - o Display a character on the specified line and column using the BSP_LCD_DisplayChar()
 - function or a complete string line using the BSP_LCD_DisplayStringAtLine() function.
 - .
 - o Display a string line on the specified position (x,y in pixel) and align mode using the BSP_LCD_DisplayStringAtLine() function.
 - o Draw and fill a basic shapes (dot, line, rectangle, circle, ellipse, .. bitmap) on LCD using the available set of functions.

| **Attention:**

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_lcd.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

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

stm32746g_discovery_lcd.h File Reference

This file contains the common defines and functions prototypes for the [stm32746g_discovery_lcd.c](#) driver. [More...](#)

```
#include "../Components/rk043fn48h/rk043fn48h.h" #include  
"stm32746g_discovery_sdram.h"  
#include "stm32746g_discovery.h"  
#include "../../../../../Utilities/Fonts/fonts.h"
```

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

Data Structures

```
struct LCD_DrawPropTypeDef
```

```
struct Point
```

Defines

```
#define MAX_LAYER_NUMBER ((uint32_t)2)
#define LCD_LayerCfgTypeDef LTDC_LayerCfgTypeDef
#define LTDC_ACTIVE_LAYER ((uint32_t)1) /* Layer 1 */
#define LCD_OK ((uint8_t)0x00)
LCD status structure definition.

#define LCD_ERROR ((uint8_t)0x01)
#define LCD_TIMEOUT ((uint8_t)0x02)
#define LCD_FB_START_ADDRESS ((uint32_t)0xC0000000)
LCD FB_StartAddress.

#define LCD_COLOR_BLUE ((uint32_t)0xFF0000FF)
LCD color.

#define LCD_COLOR_GREEN ((uint32_t)0xFF00FF00)
#define LCD_COLOR_RED ((uint32_t)0xFFFF0000)
#define LCD_COLOR_CYAN ((uint32_t)0xFF00FFFF)
#define LCD_COLOR_MAGENTA ((uint32_t)0xFFFF00FF)
#define LCD_COLOR_YELLOW ((uint32_t)0xFFFFFFFF00)
#define LCD_COLOR_LIGHTBLUE ((uint32_t)0xFF8080FF)
#define LCD_COLOR_LIGHTGREEN ((uint32_t)0xFF80FF80)
#define LCD_COLOR_LIGHTRED ((uint32_t)0xFFFF8080)
#define LCD_COLOR_LIGHTCYAN ((uint32_t)0xFF80FFFF)
#define LCD_COLOR_LIGHTMAGENTA ((uint32_t)0xFFFF80FF)
#define LCD_COLOR_LIGHTYELLOW ((uint32_t)0xFFFFFFFF80)
#define LCD_COLOR_DARKBLUE ((uint32_t)0xFF000080)
#define LCD_COLOR_DARKGREEN ((uint32_t)0xFF008000)
#define LCD_COLOR_DARKRED ((uint32_t)0xFF800000)
#define LCD_COLOR_DARKCYAN ((uint32_t)0xFF008080)
#define LCD_COLOR_DARKMAGENTA ((uint32_t)0xFF800080)
#define LCD_COLOR_DARKYELLOW ((uint32_t)0xFF808000)
#define LCD_COLOR_WHITE ((uint32_t)0xFFFFFFFFFF)
#define LCD_COLOR_LIGHTGRAY ((uint32_t)0xFFD3D3D3)
#define LCD_COLOR_GRAY ((uint32_t)0xFF808080)
```

```
#define LCD_COLOR_DARKGRAY ((uint32_t)0xFF404040)
#define LCD_COLOR_BLACK ((uint32_t)0xFF000000)
#define LCD_COLOR_BROWN ((uint32_t)0xFFA52A2A)
#define LCD_COLOR_ORANGE ((uint32_t)0xFFFFA500)
#define LCD_COLOR_TRANSPARENT ((uint32_t)0xFF000000)
#define LCD_DEFAULT_FONT Font24
LCD default font.

#define LCD_RELOAD_IMMEDIATE ((uint32_t)LTDC_SRCCR_IMR)
LCD Reload Types.

#define LCD_RELOAD_VERTICAL_BLANKING ((uint32_t)LTDC_SF
#define LCD_DISP_PIN GPIO_PIN_12
LCD special pins.

#define LCD_DISP_GPIO_PORT GPIOI
#define LCD_DISP_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()
#define LCD_DISP_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DISABLE()
#define LCD_BL_CTRL_PIN GPIO_PIN_3
#define LCD_BL_CTRL_GPIO_PORT GPIOK
#define LCD_BL_CTRL_GPIO_CLK_ENABLE() __HAL_RCC_GPIOK_CLK_ENABLE()
#define LCD_BL_CTRL_GPIO_CLK_DISABLE() __HAL_RCC_GPIOK_CLK_DISABLE()
```

TypeDefs

```
typedef struct Point * pPoint
```

Enumerations

enum **Text_AlignModeTypdef** { **CENTER_MODE** = 0x01,
RIGHT_MODE = 0x02, **LEFT_MODE** = 0x03 }

Line mode structures definition. More...

Functions

uint8_t **BSP_LCD_Init** (void)

Initializes the LCD.

uint8_t **BSP_LCD_DelInit** (void)

Deinitializes the LCD.

uint32_t **BSP_LCD_GetXSize** (void)

Gets the LCD X size.

uint32_t **BSP_LCD_GetYSize** (void)

Gets the LCD Y size.

void **BSP_LCD_SetXSize** (uint32_t imageWidthPixels)

Set the LCD X size.

void **BSP_LCD_SetYSize** (uint32_t imageHeightPixels)

Set the LCD Y size.

void **BSP_LCD_LayerDefaultInit** (uint16_t LayerIndex,

uint32_t FB_Address)

Initializes the LCD layer in ARGB8888 format (32 bits per pixel).

void **BSP_LCD_LayerRgb565Init** (uint16_t LayerIndex,

uint32_t FB_Address)

Initializes the LCD layer in RGB565 format (16 bits per pixel).

void **BSP_LCD_SetTransparency** (uint32_t LayerIndex,

uint8_t Transparency)

Configures the transparency.

void **BSP_LCD_SetTransparency_NoReload** (uint32_t

LayerIndex, uint8_t Transparency)

Configures the transparency without reloading.

void **BSP_LCD_SetLayerAddress** (uint32_t LayerIndex,

uint32_t Address)

Sets an LCD layer frame buffer address.

void **BSP_LCD_SetLayerAddress_NoReload** (uint32_t

LayerIndex, uint32_t Address)

	Sets an LCD layer frame buffer address without reloading.
void	BSP_LCD_SetColorKeying (uint32_t LayerIndex, uint32_t RGBValue) Configures and sets the color keying.
void	BSP_LCD_SetColorKeying_NoReload (uint32_t LayerIndex, uint32_t RGBValue) Configures and sets the color keying without reloading.
void	BSP_LCD_ResetColorKeying (uint32_t LayerIndex) Disables the color keying.
void	BSP_LCD_ResetColorKeying_NoReload (uint32_t LayerIndex) Disables the color keying without reloading.
void	BSP_LCD_SetLayerWindow (uint16_t LayerIndex, uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height) Sets display window.
void	BSP_LCD_SetLayerWindow_NoReload (uint16_t LayerIndex, uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height) Sets display window without reloading.
void	BSP_LCD_SelectLayer (uint32_t LayerIndex) Selects the LCD Layer.
void	BSP_LCD_SetLayerVisible (uint32_t LayerIndex, FunctionalState State) Sets an LCD Layer visible.
void	BSP_LCD_SetLayerVisible_NoReload (uint32_t LayerIndex, FunctionalState State) Sets an LCD Layer visible without reloading.
void	BSP_LCD_Reload (uint32_t ReloadType) Disables the color keying without reloading.
void	BSP_LCD_SetTextColor (uint32_t Color) Sets the LCD text color.
uint32_t	BSP_LCD_GetTextColor (void)

	Gets the LCD text color.
void	BSP_LCD_SetBackColor (uint32_t Color) Sets the LCD background color.
uint32_t	BSP_LCD_GetBackColor (void) Gets the LCD background color.
void	BSP_LCD_SetFont (sFONT *fonts) Sets the LCD text font.
sFONT *	BSP_LCD_GetFont (void) Gets the LCD text font.
uint32_t	BSP_LCD_ReadPixel (uint16_t Xpos, uint16_t Ypos) Reads an LCD pixel.
void	BSP_LCD_DrawPixel (uint16_t Xpos, uint16_t Ypos, uint32_t RGB_Code) Draws a pixel on LCD.
void	BSP_LCD_Clear (uint32_t Color) Clears the hole LCD.
void	BSP_LCD_ClearStringLine (uint32_t Line) Clears the selected line.
void	BSP_LCD_DisplayStringAtLine (uint16_t Line, uint8_t *ptr) Displays a maximum of 60 characters on the LCD.
void	BSP_LCD_DisplayStringAt (uint16_t Xpos, uint16_t Ypos, uint8_t *Text, Text_AlignModeTypdef Mode) Displays characters on the LCD.
void	BSP_LCD_DisplayChar (uint16_t Xpos, uint16_t Ypos, uint8_t Ascii) Displays one character.
void	BSP_LCD_DrawHLine (uint16_t Xpos, uint16_t Ypos, uint16_t Length) Draws an horizontal line.
void	BSP_LCD_DrawVLine (uint16_t Xpos, uint16_t Ypos, uint16_t Length) Draws a vertical line.
	BSP_LCD_DrawLine (uint16_t x1, uint16_t y1, uint16_t

void	x2, uint16_t y2)	Draws an uni-line (between two points).
void	BSP_LCD_DrawRect (uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height)	Draws a rectangle.
void	BSP_LCD_DrawCircle (uint16_t Xpos, uint16_t Ypos, uint16_t Radius)	Draws a circle.
void	BSP_LCD_DrawPolygon (pPoint Points, uint16_t PointCount)	Draws an poly-line (between many points).
void	BSP_LCD_DrawEllipse (int Xpos, int Ypos, int XRadius, int YRadius)	Draws an ellipse on LCD.
void	BSP_LCD_DrawBitmap (uint32_t Xpos, uint32_t Ypos, uint8_t *pbmp)	Draws a bitmap picture loaded in the internal Flash in ARGB888 format (32 bits per pixel).
void	BSP_LCD_FillRect (uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height)	Draws a full rectangle.
void	BSP_LCD_FillCircle (uint16_t Xpos, uint16_t Ypos, uint16_t Radius)	Draws a full circle.
void	BSP_LCD_FillPolygon (pPoint Points, uint16_t PointCount)	Draws a full poly-line (between many points).
void	BSP_LCD_FillEllipse (int Xpos, int Ypos, int XRadius, int YRadius)	Draws a full ellipse.
void	BSP_LCD_DisplayOff (void)	Disables the display.
void	BSP_LCD_DisplayOn (void)	Enables the display.

- `__weak void BSP_LCD_MspInit (LTDC_HandleTypeDef *hltcd, void *Params)`
Initializes the LTDC MSP.
-
- `__weak void BSP_LCD_MspDeInit (LTDC_HandleTypeDef *hltcd, void *Params)`
DeInitializes BSP_LCD MSP.
-
- `__weak void BSP_LCD_ClockConfig (LTDC_HandleTypeDef *hltcd, void *Params)`
Clock Config.

Detailed Description

This file contains the common defines and functions prototypes for the [**stm32746g_discovery_lcd.c**](#) driver.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_lcd.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	
Functions Variables			

stm32746g_discovery_qspi.c File Reference

This file includes a standard driver for the N25Q128A QSPI memory mounted on STM32746G-Discovery board. [More...](#)

```
#include "stm32746g_discovery_qspi.h"
```

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

Functions

static uint8_t **QSPI_ResetMemory** (QSPI_HandleTypeDef *hqspi)

This function reset the QSPI memory.

static uint8_t **QSPI_DummyCyclesCfg** (QSPI_HandleTypeDef *hqspi)

This function configure the dummy cycles on memory side.

static uint8_t **QSPI_WriteEnable** (QSPI_HandleTypeDef *hqspi)

This function send a Write Enable and wait it is effective.

static uint8_t **QSPI_AutoPollingMemReady** (QSPI_HandleTypeDef *hqspi, uint32_t Timeout)

This function read the SR of the memory and wait the EOP.

uint8_t **BSP_QSPI_Init** (void)

Initializes the QSPI interface.

uint8_t **BSP_QSPI_DeInit** (void)

De-Initializes the QSPI interface.

uint8_t **BSP_QSPI_Read** (uint8_t *pData, uint32_t ReadAddr, uint32_t Size)

Reads an amount of data from the QSPI memory.

uint8_t **BSP_QSPI_Write** (uint8_t *pData, uint32_t WriteAddr, uint32_t Size)

Writes an amount of data to the QSPI memory.

uint8_t **BSP_QSPI_Erase_Block** (uint32_t BlockAddress)

Erases the specified block of the QSPI memory.

uint8_t **BSP_QSPI_Erase_Chip** (void)

Erases the entire QSPI memory.

uint8_t **BSP_QSPI_GetStatus** (void)

Reads current status of the QSPI memory.

uint8_t **BSP_QSPI_GetInfo** (QSPI_Info * pInfo)

Return the configuration of the QSPI memory.

`uint8_t BSP_QSPI_EnableMemoryMappedMode (void)`

Configure the QSPI in memory-mapped mode.

`__weak void BSP_QSPI_MspInit (QSPI_HandleTypeDef *hqspi, void *Params)`

QSPI MSP Initialization This function configures the hardware resources used in this example:

`__weak void BSP_QSPI_MspDeInit (QSPI_HandleTypeDef *hqspi, void *Params)`

QSPI MSP De-Initialization This function frees the hardware resources used in this example:

Variables

QSPI_HandleTypeDef **QSPIHandle**

Detailed Description

This file includes a standard driver for the N25Q128A QSPI memory mounted on STM32746G-Discovery board.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

```
=====
=====
##### How to use this driver #####
=====
[.]
( #) This driver is used to drive the N25Q128A QSPI external
     memory mounted on STM32746G-Discovery board.

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

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

(#) QSPI memory operations

(++) QSPI memory can be accessed with read/write operations once it is initialized.

Read/write operation can be performed with AHB access using the functions `BSP_QSPI_Read()`/`BSP_QSPI_Write()`.

(++) The function `BSP_QSPI_GetInfo()` returns the configuration of the QSPI memory.
(see the QSPI memory data sheet)

(++) Perform erase block operation using the function `BSP_QSPI_Erase_Block()` and by specifying the block address. You can perform an erase operation of the whole chip by calling the function `BSP_QSPI_Erase_Chip()`.

(++) The function `BSP_QSPI_GetStatus()` returns the current status of the QSPI memory.
(see the QSPI memory data sheet)

| **Attention:**

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_qspi.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	
Data Structures Defines Functions			
<h2>stm32746g_discovery_qspi.h File Reference</h2>			

This file contains the common defines and functions prototypes for the [stm32746g_discovery_qspi.c](#) driver. [More...](#)

```
#include "stm32f7xx_hal.h" #include  
"../../Components/n25q128a/n25q128a.h"
```

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

Data Structures

```
struct QSPI_Info
```

Defines

```
#define QSPI_OK ((uint8_t)0x00)
#define QSPI_ERROR ((uint8_t)0x01)
#define QSPI_BUSY ((uint8_t)0x02)
#define QSPI_NOT_SUPPORTED ((uint8_t)0x04)
#define QSPI_SUSPENDED ((uint8_t)0x08)
#define QSPI_CLK_ENABLE() __HAL_RCC_QSPI_CLK_ENABLE()
#define QSPI_CLK_DISABLE() __HAL_RCC_QSPI_CLK_DISABLE()
#define QSPI_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define QSPI_CLK_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define QSPI_D0_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define QSPI_D1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define QSPI_D2_GPIO_CLK_ENABLE() __HAL_RCC_GPIOE_CLK_ENABLE()
#define QSPI_D3_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define QSPI_FORCE_RESET() __HAL_RCC_QSPI_FORCE_RESET()
#define QSPI_RELEASE_RESET() __HAL_RCC_QSPI_RELEASE_RESET()
#define QSPI_CS_PIN GPIO_PIN_6
#define QSPI_CS_GPIO_PORT GPIOB
#define QSPI_CLK_PIN GPIO_PIN_2
#define QSPI_CLK_GPIO_PORT GPIOB
#define QSPI_D0_PIN GPIO_PIN_11
#define QSPI_D0_GPIO_PORT GPIOD
#define QSPI_D1_PIN GPIO_PIN_12
#define QSPI_D1_GPIO_PORT GPIOD
#define QSPI_D2_PIN GPIO_PIN_2
#define QSPI_D2_GPIO_PORT GPIOE
#define QSPI_D3_PIN GPIO_PIN_13
#define QSPI_D3_GPIO_PORT GPIOD
#define QSPI_FLASH_SIZE 23 /* Address bus width to access whole
#define QSPI_PAGE_SIZE 256
#define BSP_QSPI_MemoryMappedMode BSP_QSPI_EnableMem
```

Functions

uint8_t **BSP_QSPI_Init** (void)

Initializes the QSPI interface.

uint8_t **BSP_QSPI_DelInit** (void)

De-Initializes the QSPI interface.

uint8_t **BSP_QSPI_Read** (uint8_t *pData, uint32_t ReadAddr,
uint32_t Size)

Reads an amount of data from the QSPI memory.

uint8_t **BSP_QSPI_Write** (uint8_t *pData, uint32_t WriteAddr,
uint32_t Size)

Writes an amount of data to the QSPI memory.

uint8_t **BSP_QSPI_Erase_Block** (uint32_t BlockAddress)

Erases the specified block of the QSPI memory.

uint8_t **BSP_QSPI_Erase_Chip** (void)

Erases the entire QSPI memory.

uint8_t **BSP_QSPI_GetStatus** (void)

Reads current status of the QSPI memory.

uint8_t **BSP_QSPI_GetInfo** (QSPI_Info *pInfo)

Return the configuration of the QSPI memory.

uint8_t **BSP_QSPI_EnableMemoryMappedMode** (void)

Configure the QSPI in memory-mapped mode.

void **BSP_QSPI_MspInit** (QSPI_HandleTypeDef *hqspi, void
*Params)

QSPI MSP Initialization This function configures the hardware resources used in this example:

void **BSP_QSPI_MspDelInit** (QSPI_HandleTypeDef *hqspi, void
*Params)

QSPI MSP De-Initialization This function frees the hardware resources used in this example:

Detailed Description

This file contains the common defines and functions prototypes for the [**stm32746g_discovery_qspi.c**](#) driver.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_qspi.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Functions Variables

stm32746g_discovery_sd.c File Reference

This file includes the uSD card driver mounted on STM32746G-Discovery board. [More...](#)

```
#include "stm32746g_discovery_sd.h"
```

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

Functions

uint8_t **BSP_SD_Init** (void)

Initializes the SD card device.

uint8_t **BSP_SD_DelInit** (void)

Deinitializes the SD card device.

uint8_t **BSP_SD_ITConfig** (void)

Configures Interrupt mode for SD detection pin.

uint8_t **BSP_SD_IsDetected** (void)

Detects if SD card is correctly plugged in the memory slot or not.

uint8_t **BSP_SD_ReadBlocks** (uint32_t *pData, uint32_t

ReadAddr, uint32_t NumOfBlocks, uint32_t Timeout)

Reads block(s) from a specified address in an SD card, in polling mode.

uint8_t **BSP_SD_WriteBlocks** (uint32_t *pData, uint32_t

WriteAddr, uint32_t NumOfBlocks, uint32_t Timeout)

Writes block(s) to a specified address in an SD card, in polling mode.

uint8_t **BSP_SD_ReadBlocks_DMA** (uint32_t *pData, uint32_t

ReadAddr, uint32_t NumOfBlocks)

Reads block(s) from a specified address in an SD card, in DMA mode.

uint8_t **BSP_SD_WriteBlocks_DMA** (uint32_t *pData, uint32_t

WriteAddr, uint32_t NumOfBlocks)

Writes block(s) to a specified address in an SD card, in DMA mode.

uint8_t **BSP_SD_Erase** (uint32_t StartAddr, uint32_t EndAddr)

Erases the specified memory area of the given SD card.

__weak void **BSP_SD_MspInit** (SD_HandleTypeDef *hsd, void *Params)

Initializes the SD MSP.

<code>__weak void</code>	BSP_SD_Detect_MspInit (SD_HandleTypeDef *hsd, void *Params)	Initializes the SD Detect pin MSP.
<code>__weak void</code>	BSP_SD_MspDeInit (SD_HandleTypeDef *hsd, void *Params)	DeInitializes the SD MSP.
<code>uint8_t</code>	BSP_SD_GetCardState (void)	Gets the current SD card data status.
<code>void</code>	BSP_SD_GetCardInfo (HAL_SD_CardInfoTypeDef *CardInfo)	Get SD information about specific SD card.
<code>void</code>	HAL_SD_AbortCallback (SD_HandleTypeDef *hsd)	SD Abort callbacks.
<code>void</code>	HAL_SD_TxCpltCallback (SD_HandleTypeDef *hsd)	Tx Transfer completed callbacks.
<code>void</code>	HAL_SD_RxCpltCallback (SD_HandleTypeDef *hsd)	Rx Transfer completed callbacks.
<code>__weak void</code>	BSP_SD_AbortCallback (void)	BSP SD Abort callbacks.
<code>__weak void</code>	BSP_SD_WriteCpltCallback (void)	BSP Tx Transfer completed callbacks.
<code>__weak void</code>	BSP_SD_ReadCpltCallback (void)	BSP Rx Transfer completed callbacks.

Variables

SD_HandleTypeDef **uSdHandle**

Detailed Description

This file includes the uSD card driver mounted on STM32746G-Discovery board.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

1. How To use this driver:

-
- This driver is used to drive the micro SD external card mounted on STM32746G-Discovery board.
 - This driver does not need a specific component driver for the micro SD device to be included with.

2. Driver description:

-
- + Initialization steps:
 - o Initialize the micro SD card using the BSP_SD_Init() function. This function includes the MSP layer hardware resources initialization and the SDIO interface configuration to interface with the external micro SD. It also includes the micro SD initialization sequence.
 - o To check the SD card presence you can use the function BSP_SD_IsDetected() which

returns the detection status

- o If SD presence detection interrupt mode is desired, you must configure the SD detection interrupt mode by calling the function BSP_SD_ITConfig(). The interrupt is generated as an external interrupt whenever the micro SD card is plugged/unplugged in/from the board.
- o The function BSP_SD_GetCardInfo() is used to get the micro SD card information which is stored in the structure "HAL_SD_CardInfoTypedef".

+ Micro SD card operations

- o The micro SD card can be accessed with read/write block(s) operations once it is ready for access. The access can be performed whether using the polling mode by calling the functions BSP_SD_ReadBlocks()/BSP_SD_WriteBlocks(), or by DMA transfer using the functions BSP_SD_ReadBlocks_DMA()/BSP_SD_WriteBlocks_DMA()
- o The DMA transfer complete is used with interrupt mode. Once the SD transfer is complete, the SD interrupt is handled using the function BSP_SD_IRQHandler(), the DMA Tx/Rx transfer complete are handled using the functions BSP_SD_DMA_Tx_IRQHandler()/BSP_SD_DMA_Rx_IRQHandler(). The corresponding user callbacks are implemented by the user at application level.
- o The SD erase block(s) is performed using the function BSP_SD_Erase() with specifying

the number of blocks to erase.

- o The SD runtime status is returned when calling the function `BSP_SD_GetCardState()`.

| **Attention:**

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_sd.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Defines Functions

stm32746g_discovery_sd.h File Reference

This file contains the common defines and functions prototypes for the [stm32746g_discovery_sd.c](#) driver. [More...](#)

```
#include "stm32746g_discovery.h"
```

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

Defines

```
#define BSP_SD_CardInfo HAL_SD_CardInfoTypeDef
SD Card information structure.

#define MSD_OK ((uint8_t)0x00)
SD status structure definition.

#define MSD_ERROR ((uint8_t)0x01)
#define MSD_ERROR_SD_NOT_PRESENT ((uint8_t)0x02)
#define SD_TRANSFER_OK ((uint8_t)0x00)
SD transfer state definition.

#define SD_TRANSFER_BUSY ((uint8_t)0x01)
#define SD_PRESENT ((uint8_t)0x01)
#define SD_NOT_PRESENT ((uint8_t)0x00)
#define SD_DATATIMEOUT ((uint32_t)1000000000)
#define _DMAx_TxRx_CLK_ENABLE __HAL_RCC_DMA2_CLK_ENABLE
#define SD_DMAx_Tx_CHANNEL DMA_CHANNEL_4
#define SD_DMAx_Rx_CHANNEL DMA_CHANNEL_4
#define SD_DMAx_Tx_STREAM DMA2_Stream6
#define SD_DMAx_Rx_STREAM DMA2_Stream3
#define SD_DMAx_Tx_IRQn DMA2_Stream6_IRQn
#define SD_DMAx_Rx_IRQn DMA2_Stream3_IRQn
#define BSP_SDMMC_IRQHandler SDMMC1_IRQHandler
#define BSP_SDMMC_DMA_Tx_IRQHandler DMA2_Stream6_IRQHandler
#define BSP_SDMMC_DMA_Rx_IRQHandler DMA2_Stream3_IRQHandler
#define SD_DetectIRQHandler() HAL_GPIO_EXTI_IRQHandler(SD_
```

Functions

uint8_t **BSP_SD_Init** (void)

Initializes the SD card device.

uint8_t **BSP_SD_DelInit** (void)

Deinitializes the SD card device.

uint8_t **BSP_SD_ITConfig** (void)

Configures Interrupt mode for SD detection pin.

uint8_t **BSP_SD_ReadBlocks** (uint32_t *pData, uint32_t

ReadAddr, uint32_t NumOfBlocks, uint32_t Timeout)

Reads block(s) from a specified address in an SD card,
in polling mode.

uint8_t **BSP_SD_WriteBlocks** (uint32_t *pData, uint32_t

WriteAddr, uint32_t NumOfBlocks, uint32_t Timeout)

Writes block(s) to a specified address in an SD card, in
polling mode.

uint8_t **BSP_SD_ReadBlocks_DMA** (uint32_t *pData, uint32_t

ReadAddr, uint32_t NumOfBlocks)

Reads block(s) from a specified address in an SD card,
in DMA mode.

uint8_t **BSP_SD_WriteBlocks_DMA** (uint32_t *pData, uint32_t

WriteAddr, uint32_t NumOfBlocks)

Writes block(s) to a specified address in an SD card, in
DMA mode.

uint8_t **BSP_SD_Erase** (uint32_t StartAddr, uint32_t EndAddr)

Erases the specified memory area of the given SD
card.

uint8_t **BSP_SD_GetCardState** (void)

Gets the current SD card data status.

void **BSP_SD_GetCardInfo** (HAL_SD_CardInfoTypeDef
*CardInfo)

Get SD information about specific SD card.

uint8_t **BSP_SD_IsDetected** (void)

	Detects if SD card is correctly plugged in the memory slot or not.
<u>__weak void</u>	BSP_SD_MspInit (SD_HandleTypeDef *hsd, void *Params) Initializes the SD MSP.
<u>__weak void</u>	BSP_SD_Detect_MspInit (SD_HandleTypeDef *hsd, void *Params) Initializes the SD Detect pin MSP.
<u>__weak void</u>	BSP_SD_MspDeInit (SD_HandleTypeDef *hsd, void *Params) DeInitializes the SD MSP.
<u>__weak void</u>	BSP_SD_AbortCallback (void) BSP SD Abort callbacks.
<u>__weak void</u>	BSP_SD_WriteCpltCallback (void) BSP Tx Transfer completed callbacks.
<u>__weak void</u>	BSP_SD_ReadCpltCallback (void) BSP Rx Transfer completed callbacks.

Detailed Description

This file contains the common defines and functions prototypes for the [**stm32746g_discovery_sd.c**](#) driver.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_sd.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Functions Variables

stm32746g_discovery_sdram.c File Reference

This file includes the SDRAM driver for the MT48LC4M32B2B5-7 memory device mounted on STM32746G-Discovery board. [More...](#)

```
#include "stm32746g_discovery_sdram.h"
```

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

Functions

uint8_t	BSP_SDRAM_Init (void)	Initializes the SDRAM device.
uint8_t	BSP_SDRAM_DeInit (void)	DeInitializes the SDRAM device.
void	BSP_SDRAM_Initialization_sequence (uint32_t RefreshCount)	Programs the SDRAM device.
uint8_t	BSP_SDRAM_ReadData (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Reads an amount of data from the SDRAM memory in polling mode.
uint8_t	BSP_SDRAM_ReadData_DMA (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Reads an amount of data from the SDRAM memory in DMA mode.
uint8_t	BSP_SDRAM_WriteData (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Writes an amount of data to the SDRAM memory in polling mode.
uint8_t	BSP_SDRAM_WriteData_DMA (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Writes an amount of data to the SDRAM memory in DMA mode.
uint8_t	BSP_SDRAM_Sendcmd (FMC_SDRAM_CommandTypeDef *SdramCmd)	Sends command to the SDRAM bank.
__weak void	BSP_SDRAM_MspInit (SDRAM_HandleTypeDef *hsdram, void *Params)	Initializes SDRAM MSP.
__weak void	BSP_SDRAM_MspDeInit (SDRAM_HandleTypeDef *hsdram, void *Params)	DeInitializes SDRAM MSP.

Variables

SDRAM_HandleTypeDef	sdramHandle
static FMC_SDRAM_TimingTypeDef	Timing
static FMC_SDRAM_CommandTypeDef	Command

Detailed Description

This file includes the SDRAM driver for the MT48LC4M32B2B5-7 memory device mounted on STM32746G-Discovery board.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

1. How To use this driver:

-
- This driver is used to drive the MT48LC4M32B2B5-7 SDRAM external memory mounted on STM32746G-Discovery board.
 - This driver does not need a specific component driver for the SDRAM device to be included with.

2. Driver description:

-
- + Initialization steps:
 - o Initialize the SDRAM external memory using the BSP_SDRAM_Init() function. This function includes the MSP layer hardware resources initialization and the FMC controller configuration to interface with the external SDRAM memory.
 - o It contains the SDRAM initialization sequence to program the SDRAM external device using the function BSP_SDRAM_Initialization_sequence(). Note that this sequence is standard for all SDRAM d

evices, but can include some differences from a device to another. If it is the case, the right sequence should be implemented separately.

- + SDRAM read/write operations

- o SDRAM external memory can be accessed with read/write operations once it is initialized.

- Read/write operation can be performed with AHB access using the functions

- BSP_SDRAM_ReadData()/BSP_SDRAM_WriteData(), or by DMA transfer using the functions

- BSP_SDRAM_ReadData_DMA()/BSP_SDRAM_WriteData_DMA().

- o The AHB access is performed with 32-bit width transaction, the DMA transfer configuration is fixed at single (no burst) word transfer (see the SDRAM_MspInit() static function).

- o User can implement his own functions for read/write access with his desired configurations.

- o If interrupt mode is used for DMA transfer, the function BSP_SDRAM_DMA_IRQHandler()

- is called in IRQ handler file, to serve the generated interrupt once the DMA transfer is complete.

- o You can send a command to the SDRAM device in runtime using the function

- BSP_SDRAM_Sendcmd(), and giving the desired command as parameter chosen between the predefined commands of the "FMC_SDRAM_CommandTypeDef" structure.

| **Attention:**

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_sdram.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Defines Functions

stm32746g_discovery_sdram.h File Reference

This file contains the common defines and functions prototypes for the [stm32746g_discovery_sdram.c](#) driver. More...

```
#include "stm32f7xx_hal.h"
```

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

Defines

```
#define SDRAM_OK ((uint8_t)0x00)
    SDRAM status structure definition.

#define SDRAM_ERROR ((uint8_t)0x01)

#define SDRAM_DEVICE_ADDR ((uint32_t)0xC0000000)

#define SDRAM_DEVICE_SIZE ((uint32_t)0x800000) /* SDRAM device size */

#define SDRAM_MEMORY_WIDTH FMC_SDRAM_MEM_BUS_WIDTH

#define SDCLOCK_PERIOD FMC_SDRAM_CLOCK_PERIOD_2

#define REFRESH_COUNT ((uint32_t)0x0603) /* SDRAM refresh count */

#define SDRAM_TIMEOUT ((uint32_t)0xFFFF)

#define __DMAx_CLK_ENABLE __HAL_RCC_DMA2_CLK_ENABLE
#define __DMAx_CLK_DISABLE __HAL_RCC_DMA2_CLK_DISABLE

#define SDRAM_DMAx_CHANNEL DMA_CHANNEL_0

#define SDRAM_DMAx_STREAM DMA2_Stream0

#define SDRAM_DMAx_IRQn DMA2_Stream0_IRQn

#define BSP_SDRAM_DMA_IRQHandler DMA2_Stream0_IRQHandler

#define SDRAM_MODEREG_BURST_LENGTH_1 ((uint16_t)0x0000)
    FMC SDRAM Mode definition register defines.

#define SDRAM_MODEREG_BURST_LENGTH_2 ((uint16_t)0x0001)
#define SDRAM_MODEREG_BURST_LENGTH_4 ((uint16_t)0x0002)
#define SDRAM_MODEREG_BURST_LENGTH_8 ((uint16_t)0x0004)
#define SDRAM_MODEREG_BURST_TYPE_SEQUENTIAL ((uint16_t)0x0008)
#define SDRAM_MODEREG_BURST_TYPE_INTERLEAVED ((uint16_t)0x0010)
#define SDRAM_MODEREG_CAS_LATENCY_2 ((uint16_t)0x0020)
#define SDRAM_MODEREG_CAS_LATENCY_3 ((uint16_t)0x0030)
#define SDRAM_MODEREG_OPERATING_MODE_STANDARD ((uint16_t)0x0040)
#define SDRAM_MODEREG_WRITEBURST_MODE_PROGRAMMED ((uint16_t)0x0080)
#define SDRAM_MODEREG_WRITEBURST_MODE_SINGLE ((uint16_t)0x0100)
```

Functions

uint8_t	BSP_SDRAM_Init (void)	Initializes the SDRAM device.
uint8_t	BSP_SDRAM_DeInit (void)	DeInitializes the SDRAM device.
void	BSP_SDRAM_Initialization_sequence (uint32_t RefreshCount)	Programs the SDRAM device.
uint8_t	BSP_SDRAM_ReadData (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Reads an amount of data from the SDRAM memory in polling mode.
uint8_t	BSP_SDRAM_ReadData_DMA (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Reads an amount of data from the SDRAM memory in DMA mode.
uint8_t	BSP_SDRAM_WriteData (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Writes an amount of data to the SDRAM memory in polling mode.
uint8_t	BSP_SDRAM_WriteData_DMA (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Writes an amount of data to the SDRAM memory in DMA mode.
uint8_t	BSP_SDRAM_Sendcmd (FMC_SDRAM_CommandTypeDef *SdramCmd)	Sends command to the SDRAM bank.
__weak void	BSP_SDRAM_MspInit (SDRAM_HandleTypeDef *hsdram, void *Params)	Initializes SDRAM MSP.
__weak void	BSP_SDRAM_MspDeInit (SDRAM_HandleTypeDef *hsdram, void *Params)	DeInitializes SDRAM MSP.

Detailed Description

This file contains the common defines and functions prototypes for the [**stm32746g_discovery_sram.c**](#) driver.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_sdram.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	Functions Variables

stm32746g_discovery_ts.c File Reference

This file provides a set of functions needed to manage the Touch Screen on STM32746G-Discovery board. [More...](#)

```
#include "stm32746g_discovery_ts.h"
```

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

Functions

uint8_t **BSP_TS_Init** (uint16_t ts_SizeX, uint16_t ts_SizeY)

Initializes and configures the touch screen functionalities and configures all necessary hardware resources (GPIOs, I2C, clocks..).

uint8_t **BSP_TS_DelInit** (void)

DeInitializes the TouchScreen.

uint8_t **BSP_TS_ITConfig** (void)

Configures and enables the touch screen interrupts.

uint8_t **BSP_TS_ITGetStatus** (void)

Gets the touch screen interrupt status.

uint8_t **BSP_TS_GetState** (**TS_StateTypeDef** *TS_State)

Returns status and positions of the touch screen.

uint8_t **BSP_TS_Get_GestureId** (**TS_StateTypeDef** *TS_State)

Update gesture Id following a touch detected.

void **BSP_TS_ITClear** (void)

Clears all touch screen interrupts.

uint8_t **BSP_TS_ResetTouchData** (**TS_StateTypeDef** *TS_State)

Function used to reset all touch data before a new acquisition of touch information.

Variables

```
static TS_DrvTypeDef * tsDriver
    static uint16_t tsXBoundary
    static uint16_t tsYBoundary
    static uint8_t tsOrientation
    static uint8_t I2cAddress
```

Detailed Description

This file provides a set of functions needed to manage the Touch Screen on STM32746G-Discovery board.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

1. How To use this driver:

- This driver is used to drive the touch screen module of the STM32746G-Discovery board on the RK043FN48H-CT672B 480x272 LCD screen with capacitive touch screen.
- The FT5336 component driver must be included in project files according to the touch screen driver present on this board.

2. Driver description:

+ Initialization steps:
o Initialize the TS module using the BSP_TS_Init() function. This function includes the MSP layer hardware resources initialization and the communication layer configuration to start the TS use. The LCD size properties (x and y) are passed as parameters.
o If TS interrupt mode is desired, you must configure the TS interrupt mode

by calling the function `BSP_TS_ITConfig()`. The TS interrupt mode is generated as an external interrupt whenever a touch is detected.

The interrupt mode internally uses the IO functionalities driver driven by the IO expander, to configure the IT line.

- + Touch screen use
 - o The touch screen state is captured whenever the function `BSP_TS_GetState()` is used. This function returns information about the last LCD touch occurred in the `TS_StateTypeDef` structure.
 - o If TS interrupt mode is used, the function `BSP_TS_ITGetStatus()` is needed to get the interrupt status. To clear the IT pending bits, you should call the function `BSP_TS_ITClear()`.
 - o The IT is handled using the corresponding external interrupt IRQ handler, the user IT callback treatment is implemented on the same external interrupt callback.

| **Attention:**

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_ts.c](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

Data Structures | Defines | Enumerations |
Functions | Variables

stm32746g_discovery_ts.h File Reference

This file contains the common defines and functions prototypes for the [stm32746g_discovery_ts.c](#) driver. [More...](#)

```
#include "stm32746g_discovery.h" #include  
"../Components/ft5336/ft5336.h"
```

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

Data Structures

struct **TS_StateTypeDef**

TS_StateTypeDef Define TS State structure. More...

Defines

```
#define TS_MAX_NB_TOUCH ((uint32_t)  
FT5336_MAX_DETECTABLE_TOUCH)  
With FT5336 : maximum 5 touches detected simultaneously.  
  
#define TS_NO_IRQ_PENDING ((uint8_t) 0)  
#define TS_IRQ_PENDING ((uint8_t) 1)  
#define TS_SWAP_NONE ((uint8_t) 0x01)  
#define TS_SWAP_X ((uint8_t) 0x02)  
#define TS_SWAP_Y ((uint8_t) 0x04)  
#define TS_SWAP_XY ((uint8_t) 0x08)
```

Enumerations

enum	TS_StatusTypeDef { TS_OK = 0x00, TS_ERROR = 0x01, TS_TIMEOUT = 0x02, TS_DEVICE_NOT_FOUND = 0x03 }
	TS_GestureIdTypeDef { GEST_ID_NO_GESTURE = 0x00, GEST_ID_MOVE_UP = 0x01, GEST_ID_MOVE_RIGHT = 0x02, GEST_ID_MOVE_DOWN = 0x03, GEST_ID_MOVE_LEFT = 0x04, GEST_ID_ZOOM_IN = 0x05, GEST_ID_ZOOM_OUT = 0x06, GEST_ID_NB_MAX = 0x07 }
	TS_GestureIdTypeDef Define Possible managed gesture identification values returned by touch screen driver. More...
enum	TS_TouchEventTypeDef { TOUCH_EVENT_NO_EVT = 0x00, TOUCH_EVENT_PRESS_DOWN = 0x01, TOUCH_EVENT_LIFT_UP = 0x02, TOUCH_EVENT_CONTACT = 0x03, TOUCH_EVENT_NB_MAX = 0x04 }
	TS_TouchEventTypeDef Define Possible touch events kind as returned values by touch screen IC Driver. More...

Functions

uint8_t **BSP_TS_Init** (uint16_t ts_SizeX, uint16_t ts_SizeY)

Initializes and configures the touch screen functionalities and configures all necessary hardware resources (GPIOs, I2C, clocks..).

uint8_t **BSP_TS_DelInit** (void)

DeInitializes the TouchScreen.

uint8_t **BSP_TS_GetState** (**TS_StateTypeDef** *TS_State)

Returns status and positions of the touch screen.

uint8_t **BSP_TS_Get_GestureId** (**TS_StateTypeDef** *TS_State)

Update gesture Id following a touch detected.

uint8_t **BSP_TS_ITConfig** (void)

Configures and enables the touch screen interrupts.

uint8_t **BSP_TS_ITGetStatus** (void)

Gets the touch screen interrupt status.

void **BSP_TS_ITClear** (void)

Clears all touch screen interrupts.

uint8_t **BSP_TS_ResetTouchData** (**TS_StateTypeDef** *TS_State)

Function used to reset all touch data before a new acquisition of touch information.

Variables

char * **ts_event_string_tab [TOUCH_EVENT_NB_MAX]**

Table for touchscreen event information display on LCD : table indexed on enum **TS_TouchEventTypeDef** information.

char * **ts_gesture_id_string_tab [GEST_ID_NB_MAX]**

Table for touchscreen gesture Id information display on LCD : table indexed on enum **TS_GestureIdTypeDef** information.

Detailed Description

This file contains the common defines and functions prototypes for the [**stm32746g_discovery_ts.c**](#) driver.

Author:

MCD Application Team

Version:

V2.0.0

Date:

30-December-2016

| Attention:

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Definition in file [stm32746g_discovery_ts.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			

Modules

Here is a list of all modules:

- **BSP**
 - **STM32746G_DISCOVERY**
 - **STM32746G_DISCOVERY_LOW_LEVEL**
 - **STM32746G_DISCOVERY_LOW_LEVEL Private Types Definitions**
 - **STM32746G_DISCOVERY_LOW_LEVEL Private Defines**
 - **STM32746G_DISCOVERY_LOW_LEVEL Private Macros**
 - **STM32746G_DISCOVERY_LOW_LEVEL Private Variables**
 - **STM32746G_DISCOVERY_LOW_LEVEL Private Function Prototypes**
 - **STM32746G_DISCOVERY_LOW_LEVEL STM32746G Exported Functions**
 - **STM32746G_DISCOVERY_LOW_LEVEL Exported Types**
 - **STM32746G_DISCOVERY_LOW_LEVEL Exported Constants**
 - **STM32746G_DISCOVERY_LOW_LEVEL_LED**
 - **STM32746G_DISCOVERY_LOW_LEVEL_BUTTON**
 - **STM32746G_DISCOVERY_LOW_LEVEL_SIGNAL**
 - **STM32746G_DISCOVERY_LOW_LEVEL_COM**
 - **STM32746G_DISCOVERY_LOW_LEVEL Exported Macros**

- **STM32746G_DISCOVERY_AUDIO**
 - STM32746G_DISCOVERY_AUDIO Private Types
 - STM32746G_DISCOVERY_AUDIO Private Defines
 - STM32746G_DISCOVERY_AUDIO Private Macros
 - STM32746G_DISCOVERY_AUDIO Private Variables
 - STM32746G_DISCOVERY_AUDIO Private Function Prototypes
 - STM32746G_DISCOVERY_AUDIO Out Exported Functions
 - STM32746G_DISCOVERY_AUDIO Out Private Functions
 - STM32746G_DISCOVERY_AUDIO Exported Types
 - STM32746G_DISCOVERY_AUDIO Exported Constants
 - STM32746G_DISCOVERY_AUDIO Exported Variables
 - STM32746G_DISCOVERY_AUDIO Exported Macros
 - STM32746G_DISCOVERY_AUDIO_IN Exported Functions
- **STM32746G-Discovery_QSPI**
 - STM32746G_DISCOVERY_QSPI Private Variables
 - STM32746G_DISCOVERY_QSPI Private Functions
 - STM32746G_DISCOVERY_QSPI Exported Functions
 - STM32746G_DISCOVERY_QSPI Exported Constants
 - STM32746G_DISCOVERY_QSPI Exported Types
- **STM32746G_DISCOVERY_SD**
 - STM32746G_DISCOVERY_SD Private Types Definitions
 - STM32746G_DISCOVERY_SD Private Defines
 - STM32746G_DISCOVERY_SD Private Macros
 - STM32746G_DISCOVERY_SD Private Variables
 - STM32746G_DISCOVERY_SD Private Function Prototypes

- [STM32746G_DISCOVERY_SD Exported Functions](#)
- [STM32746G_DISCOVERY_SD Exported Types](#)
- [STM32746G_DISCOVERY_SD Exported Constants](#)
- [STM32746G_DISCOVERY_SD Exported Macro](#)
- [STM32746G_DISCOVERY_SDRAM](#)
 - [STM32746G_DISCOVERY_SDRAM Private Types Definitions](#)
 - [STM32746G_DISCOVERY_SDRAM Private Defines](#)
 - [STM32746G_DISCOVERY_SDRAM Private Macros](#)
 - [STM32746G_DISCOVERY_SDRAM Private Variables](#)
 - [STM32746G_DISCOVERY_SDRAM Private Function Prototypes](#)
 - [STM32746G_DISCOVERY_SDRAM Exported Functions](#)
 - [STM32746G_DISCOVERY_SDRAM Exported Types](#)
 - [STM32746G_DISCOVERY_SDRAM Exported Constants](#)
 - [STM32746G_DISCOVERY_SDRAM Exported Macro](#)
- [STM32746G_DISCOVERY_TS](#)
 - [STM32746G_DISCOVERY_TS Types Definitions](#)
 - [STM32746G_DISCOVERY_TS Types Defines](#)
 - [STM32746G_DISCOVERY_TS Private Macros](#)
 - [STM32746G_DISCOVERY_TS Imported Variables](#)
 - [STM32746G_DISCOVERY_TS Private Variables](#)
 - [STM32746G_DISCOVERY_TS Private Function Prototypes](#)
 - [STM32746G_DISCOVERY_TS Exported Functions](#)
 - [TS Private Functions](#)
 - [STM32746G_DISCOVERY_TS Exported Constants](#)
 - [STM32746G_DISCOVERY_TS Exported Types](#)
- [STM32746G_DISCOVERY_CAMERA](#)
 - [STM32746G_DISCOVERY_CAMERA Private Types](#)

Definitions

- **STM32746G_DISCOVERY_CAMERA Private Defines**
- **STM32746G_DISCOVERY_CAMERA Private Macros**
- **STM32746G_DISCOVERY_CAMERA Private Variables**
- **STM32746G_DISCOVERY_CAMERA Private Function Prototypes**
- **STM32746G_DISCOVERY_CAMERA Exported Functions**
- **STM32746G_DISCOVERY_CAMERA Exported Types**
- **STM32746G_DISCOVERY_CAMERA Exported Constants**
- **STM32746G_DISCOVERY_EEPROM**
 - **STM32746G_DISCOVERY_EEPROM Private Types**
 - **STM32746G_DISCOVERY_EEPROM Private Defines**
 - **STM32746G_DISCOVERY_EEPROM Private Macros**
 - **STM32746G_DISCOVERY_EEPROM Private Variables**
 - **STM32746G_DISCOVERY_EEPROM Private Function Prototypes**
 - **STM32746G_DISCOVERY_EEPROM Exported Functions**
 - **STM32746G_DISCOVERY_EEPROM Exported Types**
 - **STM32746G_DISCOVERY_EEPROM Exported Constants**
 - **STM32746G_DISCOVERY_EEPROM Exported Macros**
- **STM32746G_DISCOVERY_LCD**
 - **STM32746G_DISCOVERY_LCD Private Types Definitions**
 - **STM32746G_DISCOVERY_LCD Private Defines**

- [**STM32746G_DISCOVERY_LCD Private Macros**](#)
 - [**STM32746G_DISCOVERY_LCD Private Variables**](#)
 - [**STM32746G_DISCOVERY_LCD Private Function Prototypes**](#)
 - [**STM32746G_DISCOVERY_LCD Exported Functions**](#)
 - [**STM32746G_DISCOVERY_LCD Exported Types**](#)
 - [**STM32746G_DISCOVERY_LCD Exported Constants**](#)
-

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Data Structures	Data Structure Index	Data Fields	

Data Structures

Here are the data structures with brief descriptions:

LCD_DrawPropTypeDef	
Point	
QSPI_Info	
TS_StateTypeDef	TS_StateTypeDef Define TS State structure

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
<h2>File List</h2>			

Here is a list of all files with brief descriptions:

stm32746g_discovery.c [code]	This file provides a set of functions to manage LEDs, buttons and COM ports available on STM32746G-Discovery board(MB1191) from STMicroelectronics
stm32746g_discovery.h [code]	This file contains definitions for STM32746G_DISCOVERY LEDs, push-buttons and COM ports hardware resources
stm32746g_discovery_audio.c [code]	This file provides the Audio driver for the STM32746G-Discovery board
stm32746g_discovery_audio.h [code]	This file contains the common defines and functions prototype for the stm32746g_discovery_audio driver
stm32746g_discovery_camera.c [code]	This file includes the driver for Camera modules mounted on the STM32746G-Discovery board
stm32746g_discovery_camera.h [code]	This file contains the common

	defines and functions protocol for the stm32746g_discovery_eeprom driver
stm32746g_discovery_eeprom.c [code]	This file provides a set of functions needed to manage an I2C M24LR64 EEPROM memory
stm32746g_discovery_eeprom.h [code]	This file contains all the function prototypes for the stm32746g_discovery_eeprom firmware driver
stm32746g_discovery_lcd.c [code]	This file includes the driver for the Liquid Crystal Display (LCD) module mounted on STM32 Discovery board
stm32746g_discovery_lcd.h [code]	This file contains the command defines and functions protocol for the stm32746g_discovery_lcd driver
stm32746g_discovery_qspi.c [code]	This file includes a standard driver for the N25Q128A QSPI memory mounted on STM32746G-Discovery board
stm32746g_discovery_qspi.h [code]	This file contains the command defines and functions protocol for the stm32746g_discovery_qspi driver
stm32746g_discovery_sd.c [code]	This file includes the uSD card driver mounted on STM32 Discovery board
stm32746g_discovery_sd.h [code]	This file contains the command defines and functions protocol

	for the stm32746g_discovery_sdram driver
stm32746g_discovery_sdram.c [code]	This file includes the SDRAM driver for the MT48LC4M32 7 memory device mounted on the STM32746G-Discovery board.
stm32746g_discovery_sdram.h [code]	This file contains the common defines and functions prototype for the stm32746g_discovery_sdram driver.
stm32746g_discovery_ts.c [code]	This file provides a set of functions needed to manage the Touch Screen on STM32746G-Discovery board.
stm32746g_discovery_ts.h [code]	This file contains the common defines and functions prototype for the stm32746g_discovery_ts driver.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Directories			

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

- [Drivers](#)
 - [BSP](#)
 - [STM32746G-Discovery](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules		

STM32746G_DISCOVERY_LOW_LEVEL

[STM32746G_DISCOVERY](#)

Modules

[**STM32746G_DISCOVERY_LOW_LEVEL Private Types Definitions**](#)
[**STM32746G_DISCOVERY_LOW_LEVEL Private Defines**](#)
[**STM32746G_DISCOVERY_LOW_LEVEL Private Macros**](#)
[**STM32746G_DISCOVERY_LOW_LEVEL Private Variables**](#)
[**STM32746G_DISCOVERY_LOW_LEVEL Private Function Prototypes**](#)
[**STM32746G_DISCOVERY_LOW_LEVEL Exported Functions**](#)
[**STM32746G_DISCOVERY_LOW_LEVEL Exported Types**](#)
[**STM32746G_DISCOVERY_LOW_LEVEL Exported Constants**](#)
[**STM32746G_DISCOVERY_LOW_LEVEL Exported Macros**](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules		

STM32746G_DISCOVERY AUDIO

[STM32746G_DISCOVERY](#)

This file includes the low layer driver for wm8994 Audio Codec available on STM32746G-Discovery board(MB1191). [More...](#)

Modules

[**STM32746G_DISCOVERY_AUDIO Private Types**](#)
[**STM32746G_DISCOVERY_AUDIO Private Defines**](#)
[**STM32746G_DISCOVERY_AUDIO Private Macros**](#)
[**STM32746G_DISCOVERY_AUDIO Private Variables**](#)
[**STM32746G_DISCOVERY_AUDIO Private Function Prototypes**](#)
[**STM32746G_DISCOVERY_AUDIO Out Exported Functions**](#)
[**STM32746G_DISCOVERY_AUDIO Out Private Functions**](#)
[**STM32746G_DISCOVERY_AUDIO Exported Types**](#)
[**STM32746G_DISCOVERY_AUDIO Exported Constants**](#)
[**STM32746G_DISCOVERY_AUDIO Exported Variables**](#)
[**STM32746G_DISCOVERY_AUDIO Exported Macros**](#)
[**STM32746G_DISCOVERY_AUDIO_IN Exported Functions**](#)

Detailed Description

This file includes the low layer driver for wm8994 Audio Codec available on STM32746G-Discovery board(MB1191).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Data Structures	Data Structure Index	Data Fields	

Data Structure Index

[L](#) | [P](#) | [Q](#) | [T](#)

L

P

Q

T

[LCD_DrawPropTypeDef](#) [Point](#) [QSPI_Info](#) [TS_StateTypeDef](#)

[L](#) | [P](#) | [Q](#) | [T](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Data Structures				
STM32746G_DISCOVERY_QSPI Exported Types				
STM32746G-Discovery QSPI				

Data Structures

struct **QSPI_Info**

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_qspi.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_qspi.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains the common d
efines and functions prototypes for
00008 *           the stm32746g_discovery_qspi.c
driver.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and bin
ary forms, with or without modification,
00015 * are permitted provided that the followin
g conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038
00039  /** @addtogroup BSP
00040      * @{
00041      */
00042
00043  /** @addtogroup STM32746G_DISCOVERY
00044      * @{
00045      */
00046
00047 /* Define to prevent recursive inclusion ---
00048 -----
00049 #ifndef __STM32746G_DISCOVERY_QSPI_H
00050 #define __STM32746G_DISCOVERY_QSPI_H
00051
00052 #ifdef __cplusplus
00053     extern "C" {
00054 #endif
00055
00056 /* Includes ---
00057 -----
00058 #include "stm32f7xx_hal.h"
00059 #include "../Components/n25q128a/n25q128a.h"
00060
00061  /** @addtogroup STM32746G_DISCOVERY_QSPI
00062      * @{
00063      */
00064
00065 /* Exported constants ---
00066 -----
00067 /** @defgroup STM32746G_DISCOVERY_QSPI_Expor
00068     _Constants STM32746G_DISCOVERY_QSPI_Exported_Co
00069     nstants
```

```

00066      * @{
00067      */
00068 /* QSPI Error codes */
00069 #define QSPI_OK          ((uint8_t)0x00)
00070 #define QSPI_ERROR        ((uint8_t)0x01)
00071 #define QSPI_BUSY         ((uint8_t)0x02)
00072 #define QSPI_NOT_SUPPORTED ((uint8_t)0x04)
00073 #define QSPI_SUSPENDED    ((uint8_t)0x08)
00074
00075
00076 /* Definition for QSPI clock resources */
00077 #define QSPI_CLK_ENABLE()           __HAL_RCC
00078 _QSPI_CLK_ENABLE()
00079 #define QSPI_CLK_DISABLE()         __HAL_RCC
00080 _QSPI_CLK_DISABLE()
00081 #define QSPI_CS_GPIO_CLK_ENABLE()   __HAL_RCC
00082 _GPIOB_CLK_ENABLE()
00083 #define QSPI_CLK_GPIO_CLK_ENABLE() __HAL_RCC
00084 _GPIOB_CLK_ENABLE()
00085 #define QSPI_D0_GPIO_CLK_ENABLE()   __HAL_RCC
00086 _GPIOD_CLK_ENABLE()
00087 #define QSPI_D1_GPIO_CLK_ENABLE()   __HAL_RCC
00088 _GPIOD_CLK_ENABLE()
00089 #define QSPI_D2_GPIO_CLK_ENABLE()   __HAL_RCC
00090 _GPIOE_CLK_ENABLE()
00091 #define QSPI_D3_GPIO_CLK_ENABLE()   __HAL_RCC
00092 _GPIOE_CLK_ENABLE()
00093 #define QSPI_FORCE_RESET()         __HAL_RCC
00094 _QSPI_FORCE_RESET()
00095 #define QSPI_RELEASE_RESET()       __HAL_RCC
00096 _QSPI_RELEASE_RESET()
00097
00098 /* Definition for QSPI Pins */
00099 #define QSPI_CS_PIN             GPIO_PIN_6
00100 #define QSPI_CS_GPIO_PORT       GPIOB

```

```

00092 #define QSPI_CLK_PIN           GPIO_PIN_2
00093 #define QSPI_CLK_GPIO_PORT     GPIOB
00094 #define QSPI_D0_PIN           GPIO_PIN_
11
00095 #define QSPI_D0_GPIO_PORT     GPIOD
00096 #define QSPI_D1_PIN           GPIO_PIN_
12
00097 #define QSPI_D1_GPIO_PORT     GPIOD
00098 #define QSPI_D2_PIN           GPIO_PIN_2
00099 #define QSPI_D2_GPIO_PORT     GPIOE
00100 #define QSPI_D3_PIN           GPIO_PIN_
13
00101 #define QSPI_D3_GPIO_PORT     GPIOD
00102
00103 /* N25Q128A13EF840E Micron memory */
00104 /* Size of the flash */
00105 #define QSPI_FLASH_SIZE        23      /*
Address bus width to access whole memory space */
00106 #define QSPI_PAGE_SIZE         256
00107
00108 /* This alias is added as the name of Memory
mapped fucntion changed */
00109 #define BSP_QSPI_MemoryMappedMode  BSP_QSPI_
EnableMemoryMappedMode
00110 /**
00111   * @}
00112   */
00113
00114 /* Exported types -----
-----*/
00115 /** @defgroup STM32746G_DISCOVERY_QSPI_Expor
ted_Types STM32746G_DISCOVERY_QSPI Exported Types
00116   * @{
00117   */
00118 /* QSPI Info */

```

```
00119 typedef struct {
00120     uint32_t FlashSize;           /*!< Size of
the flash */
00121     uint32_t EraseSectorSize;    /*!< Size of
sectors for the erase operation */
00122     uint32_t EraseSectorsNumber; /*!< Number o
f sectors for the erase operation */
00123     uint32_t ProgPageSize;       /*!< Size of
pages for the program operation */
00124     uint32_t ProgPagesNumber;    /*!< Number o
f pages for the program operation */
00125 } QSPI_Info;
00126
00127 /**
00128 * @}
00129 */
00130
00131
00132 /* Exported functions -----
-----*/
00133 /** @addtogroup STM32746G_DISCOVERY_QSPI_Exp
orted_Functions
00134 */
00135
00136 uint8_t BSP_QSPI_Init      (void);
00137 uint8_t BSP_QSPI_DeInit   (void);
00138 uint8_t BSP_QSPI_Read     (uint8_t* pData,
    uint32_t ReadAddr, uint32_t Size);
00139 uint8_t BSP_QSPI_Write    (uint8_t* pData,
    uint32_t WriteAddr, uint32_t Size);
00140 uint8_t BSP_QSPI_Erase_Block(uint32_t BlockA
ddress);
00141 uint8_t BSP_QSPI_Erase_Chip (void);
00142 uint8_t BSP_QSPI_GetStatus (void);
00143 uint8_t BSP_QSPI_GetInfo   (QSPI_Info* pInfo);
00144 uint8_t BSP_QSPI_EnableMemoryMappedMode(void
```

```
 );
00145
00146 /* These functions can be modified in case t
he current settings
00147     need to be changed for specific applicati
on needs */
00148 void BSP_QSPI_MspInit(QSPI_HandleTypeDefDef *hq
spi, void *Params);
00149 void BSP_QSPI_MspDeInit(QSPI_HandleTypeDefDef * hqspi, void *Params);
00150
00151 /**
00152     * @}
00153     */
00154
00155 /**
00156     * @}
00157     */
00158
00159 #ifdef __cplusplus
00160 }
00161 #endif
00162
00163 #endif /* __STM32746G_DISCOVERY_QSPI_H */
00164 /**
00165     * @}
00166     */
00167
00168 /**
00169     * @}
00170     */
00171
00172 /***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/
```

BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_qspi.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_qspi.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file includes a standard driver for the N25Q128A QSPI
00008 *           memory mounted on STM32746G-Discovery board.
00009 * @verbatim
00010 =====
=====
00011                                     ##### How to use this driver #####
00012 =====
=====
00013 [..]
00014 (#) This driver is used to drive the N25Q128A QSPI external
00015           memory mounted on STM32746G-Discovery
```

board.

00016

00017 (#) This driver need a specific component
driver (N25Q128A) to be included with.

00018

00019 (#) Initialization steps:

00020 (++) Initialize the QPSI external mem
ory using the BSP_QSPI_Init() function. This
00021 function includes the MSP layer
hardware resources initialization and the
00022 QSPI interface with the external
memory.

00023

00024 (#) QSPI memory operations

00025 (++) QSPI memory can be accessed with
read/write operations once it is
00026 initialized.

00027 Read/write operation can be perf
ormed with AHB access using the functions
00028 BSP_QSPI_Read()/BSP_QSPI_Write()

.

00029 (++) The function BSP_QSPI_GetInfo()
returns the configuration of the QSPI memory.

00030 (see the QSPI memory data sheet)

00031 (++) Perform erase block operation us
ing the function BSP_QSPI_Erase_Block() and by
00032 specifying the block address. Yo
u can perform an erase operation of the whole
00033 chip by calling the function BSP
_QSPI_Erase_Chip().

00034 (++) The function BSP_QSPI_GetStatus()
) returns the current status of the QSPI memory.

00035 (see the QSPI memory data sheet)

00036 @endverbatim

00037 ****

00038 * @attention

00039 *
00040 * <h2><center>© COPYRIGHT(c) 2016 STMicroelectronics</center></h2>
00041 *
00042 * Redistribution and use in source and binary forms, with or without modification,
00043 * are permitted provided that the following conditions are met:
00044 * 1. Redistributions of source code must retain the above copyright notice,
00045 * this list of conditions and the following disclaimer.
00046 * 2. Redistributions in binary form must reproduce the above copyright notice,
00047 * this list of conditions and the following disclaimer in the documentation
00048 * and/or other materials provided with the distribution.
00049 * 3. Neither the name of STMicroelectronics nor the names of its contributors
00050 * may be used to endorse or promote products derived from this software
00051 * without specific prior written permission.
00052 *
00053 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
00054 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
00055 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
00056 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
00057 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
00058 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

```
00059      * SERVICES; LOSS OF USE, DATA, OR PROFITS;  
00060      * OR BUSINESS INTERRUPTION) HOWEVER  
00060      * CAUSED AND ON ANY THEORY OF LIABILITY, W  
00061      * HETHER IN CONTRACT, STRICT LIABILITY,  
00061      * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
00062      * SE) ARISING IN ANY WAY OUT OF THE USE  
00062      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
00062      * POSSIBILITY OF SUCH DAMAGE.  
00063      *  
00064      ****  
00065      */  
00066  
00067 /* Includes -----  
-----*/  
00068 #include "stm32746g_discovery_qspi.h"  
00069  
00070 /** @addtogroup BSP  
00071     * @{  
00072     */  
00073  
00074 /** @addtogroup STM32746G_DISCOVERY  
00075     * @{  
00076     */  
00077  
00078 /** @defgroup STM32746G_DISCOVERY_QSPI STM32  
00079     * @{  
00080     */  
00081  
00082  
00083 /* Private variables -----  
-----*/  
00084  
00085 /** @defgroup STM32746G_DISCOVERY_QSPI_Priva  
00085     * Variables STM32746G_DISCOVERY_QSPI_Private Vari  
00085     * ables
```

```
00086      * @@
00087      */
00088 QSPI_HandleTypeDef QSPIHandle;
00089
00090 /**
00091      * @@
00092      */
00093
00094
00095
00096 /* Private functions -----
00097 ----- */
00098 /** @defgroup STM32746G_DISCOVERY_QSPI_Private_Functions STM32746G_DISCOVERY QSPI Private Functions
00099      * @@
00100      */
00101 static uint8_t QSPI_ResetMemory          (QS
00102 PI_HandleTypeDef *hqspi);
00103 static uint8_t QSPI_DummyCyclesCfg     (QS
00104 PI_HandleTypeDef *hqspi);
00105 static uint8_t QSPI_WriteEnable         (QS
00106 PI_HandleTypeDef *hqspi);
00107 static uint8_t QSPI_AutoPollingMemReady (QS
00108 PI_HandleTypeDef *hqspi, uint32_t Timeout);
00109
00110 /**
00111      * @@
00112      */
00113
```

```
00114 /**
00115  * @brief  Initializes the QSPI interface.
00116  * @retval QSPI memory status
00117 */
00118 uint8_t BSP_QSPI_Init(void)
00119 {
00120     QSPIHandle.Instance = QUADSPI;
00121
00122     /* Call the DeInit function to reset the d
river */
00123     if (HAL_QSPI_DeInit(&QSPIHandle) != HAL_OK)
00124     {
00125         return QSPI_ERROR;
00126     }
00127
00128     /* System level initialization */
00129     BSP_QSPI_MspInit(&QSPIHandle, NULL);
00130
00131     /* QSPI initialization */
00132     QSPIHandle.Init.ClockPrescaler      = 1; /* 
QSPI freq = 216 MHz/(1+1) = 108 Mhz */
00133     QSPIHandle.Init.FifoThreshold      = 4;
00134     QSPIHandle.Init.SampleShifting    = QSPI_
SAMPLE_SHIFTING_HALFCYCLE;
00135     QSPIHandle.Init.FlashSize          = POSIT
ION_VAL(N25Q128A_FLASH_SIZE) - 1;
00136     QSPIHandle.Init.ChipSelectHighTime = QSPI_
CS_HIGH_TIME_6_CYCLE; /* Min 50ns for nonRead */
00137     QSPIHandle.Init.ClockMode         = QSPI_
CLOCK_MODE_0;
00138     QSPIHandle.Init.FlashID          = QSPI_
FLASH_ID_1;
00139     QSPIHandle.Init.DualFlash        = QSPI_
DUALFLASH_DISABLE;
00140
00141     if (HAL_QSPI_Init(&QSPIHandle) != HAL_OK)
```

```
00142     {
00143         return QSPI_ERROR;
00144     }
00145
00146     /* QSPI memory reset */
00147     if (QSPI_ResetMemory(&QSPIHandle) != QSPI_OK)
00148     {
00149         return QSPI_NOT_SUPPORTED;
00150     }
00151
00152     /* Configuration of the dummy cycles on QS
00153     PI memory side */
00154     if (QSPI_DummyCyclesCfg(&QSPIHandle) != QS
00155     PI_OK)
00156     {
00157         return QSPI_NOT_SUPPORTED;
00158     }
00159 }
00160
00161 /**
00162 * @brief De-Initializes the QSPI interface.
00163 * @retval QSPI memory status
00164 */
00165 uint8_t BSP_QSPI_DeInit(void)
00166 {
00167     QSPIHandle.Instance = QUADSPI;
00168
00169     /* Call the DeInit function to reset the d
00170     river */
00171     if (HAL_QSPI_DeInit(&QSPIHandle) != HAL_OK
00172     )
00173     {
00174         return QSPI_ERROR;
```

```
00173     }
00174
00175     /* System level De-initialization */
00176     BSP_QSPI_MspDeInit(&QSPIHandle, NULL);
00177
00178     return QSPI_OK;
00179 }
00180
00181 /**
00182     * @brief Reads an amount of data from the
00183     * QSPI memory.
00184     * @param pData: Pointer to data to be read
00185     * @param ReadAddr: Read start address
00186     * @param Size: Size of data to read
00187     * @retval QSPI memory status
00188 */
00189 uint8_t BSP_QSPI_Read(uint8_t* pData, uint32
_t ReadAddr, uint32_t Size)
00190 {
00191     QSPI_CommandTypeDef s_command;
00192
00193     /* Initialize the read command */
00194     s_command.InstructionMode      = QSPI_INSTRU
CTION_1_LINE;
00195     s_command.Instruction          = QUAD_INOUT_F
AST_READ_CMD;
00196     s_command.AddressMode         = QSPI_ADDRESS
_4_LINES;
00197     s_command.AddressSize        = QSPI_ADDRESS
_24_BITS;
00198     s_command.Address            = ReadAddr;
00199     s_command.AlternateByteMode = QSPI_ALTERNA
TE_BYTES_NONE;
00200     s_command.DataMode          = QSPI_DATA_4_
LINES;
00201     s_command.DummyCycles       = N25Q128A_DUM
```

```

MY_CYCLES_READ_QUAD;
00201    s_command.NbData           = Size;
00202    s_command.DdrMode          = QSPI_DDR_MOD
E_DISABLE;
00203    s_command.DdrHoldHalfCycle = QSPI_DDR_HHC
_ANALOG_DELAY;
00204    s_command.SIO0Mode         = QSPI_SIO0_IN
ST_EVERY_CMD;
00205
00206    /* Configure the command */
00207    if (HAL_QSPI_Command(&QSPIHandle, &s_comma
nd, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00208    {
00209        return QSPI_ERROR;
00210    }
00211
00212    /* Set S# timing for Read command */
00213    MODIFY_REG(QSPIHandle.Instance->DCR, QUADS
PI_DCR_CSHT, QSPI_CS_HIGH_TIME_3_CYCLE);
00214
00215    /* Reception of the data */
00216    if (HAL_QSPI_Receive(&QSPIHandle, pData, H
AL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00217    {
00218        return QSPI_ERROR;
00219    }
00220
00221    /* Restore S# timing for nonRead commands
*/
00222    MODIFY_REG(QSPIHandle.Instance->DCR, QUADS
PI_DCR_CSHT, QSPI_CS_HIGH_TIME_6_CYCLE);
00223
00224    return QSPI_OK;
00225 }
00226
00227 /**
00228 * @brief Writes an amount of data to the

```

```
QSPI memory.

00229     * @param  pData: Pointer to data to be written
00230     * @param  WriteAddr: Write start address
00231     * @param  Size: Size of data to write
00232     * @retval QSPI memory status
00233 */
00234 uint8_t BSP_QSPI_Write(uint8_t* pData, uint32_t WriteAddr, uint32_t Size)
00235 {
00236     QSPI_CommandTypeDef s_command;
00237     uint32_t end_addr, current_size, current_addr;
00238
00239     /* Calculation of the size between the write address and the end of the page */
00240     current_size = N25Q128A_PAGE_SIZE - (WriteAddr % N25Q128A_PAGE_SIZE);
00241
00242     /* Check if the size of the data is less than the remaining place in the page */
00243     if (current_size > Size)
00244     {
00245         current_size = Size;
00246     }
00247
00248     /* Initialize the address variables */
00249     current_addr = WriteAddr;
00250     end_addr = WriteAddr + Size;
00251
00252     /* Initialize the program command */
00253     s_command.InstructionMode      = QSPI_INSTRUCTION_1_LINE;
00254     s_command.Instruction          = EXT_QUAD_IN_FAST_PROG_CMD;
00255     s_command.AddressMode         = QSPI_ADDRESS_4_LINES;
```

```
00256     s_command.AddressSize      = QSPI_ADDRESS  
_24_BITS;  
00257     s_command.AlternateByteMode = QSPI_ALTERNA  
TE_BYTES_NONE;  
00258     s_command.DataMode        = QSPI_DATA_4_  
LINES;  
00259     s_command.DummyCycles    = 0;  
00260     s_command.DdrMode         = QSPI_DDR_MOD  
E_DISABLE;  
00261     s_command.DdrHoldHalfCycle = QSPI_DDR_HHC  
_ANALOG_DELAY;  
00262     s_command.SI00Mode        = QSPI_SI00_IN  
ST_EVERY_CMD;  
00263  
00264     /* Perform the write page by page */  
00265     do  
00266     {  
00267         s_command.Address = current_addr;  
00268         s_command.NbData  = current_size;  
00269  
00270         /* Enable write operations */  
00271         if (QSPI_WriteEnable(&QSPIHandle) != QSP  
I_OK)  
00272             {  
00273                 return QSPI_ERROR;  
00274             }  
00275  
00276         /* Configure the command */  
00277         if (HAL_QSPI_Command(&QSPIHandle, &s_com  
mand, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)  
00278             {  
00279                 return QSPI_ERROR;  
00280             }  
00281  
00282         /* Transmission of the data */  
00283         if (HAL_QSPI_Transmit(&QSPIHandle, pData  
, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
```

```
00284     {
00285         return QSPI_ERROR;
00286     }
00287
00288     /* Configure automatic polling mode to w
ait for end of program */
00289     if (QSPI_AutoPollingMemReady(&QSPIHandle
, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != QSPI_OK)
00290     {
00291         return QSPI_ERROR;
00292     }
00293
00294     /* Update the address and size variables
for next page programming */
00295     current_addr += current_size;
00296     pData += current_size;
00297     current_size = ((current_addr + N25Q128A
_PAGE_SIZE) > end_addr) ? (end_addr - current_addr
) : N25Q128A_PAGE_SIZE;
00298 } while (current_addr < end_addr);
00299
00300 return QSPI_OK;
00301 }
00302
00303 /**
00304 * @brief Erases the specified block of th
e QSPI memory.
00305 * @param BlockAddress: Block address to e
rase
00306 * @retval QSPI memory status
00307 */
00308 uint8_t BSP_QSPI_Erase_Block(uint32_t BlockA
ddress)
00309 {
00310     QSPI_CommandTypeDef s_command;
00311
00312     /* Initialize the erase command */
```

```
00313     s_command.InstructionMode    = QSPI_INSTRUCTION_1_LINE;
00314     s_command.Instruction        = SUBSECTOR_ERASE_CMD;
00315     s_command.AddressMode       = QSPI_ADDRESS_1_LINE;
00316     s_command.AddressSize      = QSPI_ADDRESS_24_BITS;
00317     s_command.Address          = BlockAddress;
00318     s_command.AlternateByteMode = QSPI_ALTERNA_TE_BYTES_NONE;
00319     s_command.DataMode         = QSPI_DATA_NO_NE;
00320     s_command.DummyCycles      = 0;
00321     s_command.DdrMode          = QSPI_DDR_MODE_DISABLE;
00322     s_command.DdrHoldHalfCycle = QSPI_DDR_HHC_ANALOG_DELAY;
00323     s_command.SIO0Mode         = QSPI_SIO0_IN_ST_EVERY_CMD;
00324
00325     /* Enable write operations */
00326     if (QSPI_WriteEnable(&QSPIHandle) != QSPI_OK)
00327     {
00328         return QSPI_ERROR;
00329     }
00330
00331     /* Send the command */
00332     if (HAL_QSPI_Command(&QSPIHandle, &s_command, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00333     {
00334         return QSPI_ERROR;
00335     }
00336
00337     /* Configure automatic polling mode to wait
```

```

t for end of erase */
00338     if (QSPI_AutoPollingMemReady(&QSPIHandle,
N25Q128A_SUBSECTOR_ERASE_MAX_TIME) != QSPI_OK)
00339     {
00340         return QSPI_ERROR;
00341     }
00342
00343     return QSPI_OK;
00344 }
00345
00346 /**
00347     * @brief Erases the entire QSPI memory.
00348     * @retval QSPI memory status
00349 */
00350 uint8_t BSP_QSPI_Erase_Chip(void)
00351 {
00352     QSPI_CommandTypeDef s_command;
00353
00354     /* Initialize the erase command */
00355     s_command.InstructionMode      = QSPI_INSTRUCTION_1_LINE;
00356     s_command/Instruction          = BULK_ERASE_CMD;
00357     s_command.AddressMode         = QSPI_ADDRESS_NONE;
00358     s_command/AlternateByteMode   = QSPI_ALTERNA_BYTES_NONE;
00359     s_command.DataMode            = QSPI_DATA_NO_NE;
00360     s_command.DummyCycles        = 0;
00361     s_command.DdrMode             = QSPI_DDR_MODE_DISABLE;
00362     s_command.DdrHoldHalfCycle   = QSPI_DDR_HHC_ANALOG_DELAY;
00363     s_command.SIO0Mode            = QSPI_SIO0_IN_ST_EVERY_CMD;
00364

```

```
00365     /* Enable write operations */
00366     if (QSPI_WriteEnable(&QSPIHandle) != QSPI_
00367         OK)
00368     {
00369         return QSPI_ERROR;
00370     }
00371     /* Send the command */
00372     if (HAL_QSPI_Command(&QSPIHandle, &s_comma
00373         nd, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00374     {
00375         return QSPI_ERROR;
00376     }
00377     /* Configure automatic polling mode to wai
00378         t for end of erase */
00379     if (QSPI_AutoPollingMemReady(&QSPIHandle,
00380         N25Q128A_BULK_ERASE_MAX_TIME) != QSPI_OK)
00381     {
00382         return QSPI_ERROR;
00383     }
00384 }
00385
00386 /**
00387     * @brief Reads current status of the QSPI
00388     * memory.
00389     * @retval QSPI memory status
00390     */
00391 uint8_t BSP_QSPI_GetStatus(void)
00392 {
00393     QSPI_CommandTypeDef s_command;
00394     uint8_t reg;
00395     /* Initialize the read flag status register
00396         command */
```

```

00396     s_command.InstructionMode    = QSPI_INSTRUCTION_1_LINE;
00397     s_command.Instruction        = READ_FLAG_STATUS_REG_CMD;
00398     s_command.AddressMode       = QSPI_ADDRESS_NONE;
00399     s_command.AlternateByteMode = QSPI_ALTERNATE_BYTES_NONE;
00400     s_command.DataMode         = QSPI_DATA_1_LINE;
00401     s_command.DummyCycles      = 0;
00402     s_command.NbData           = 1;
00403     s_command.DdrMode          = QSPI_DDR_MODE_DISABLE;
00404     s_command.DdrHoldHalfCycle = QSPI_DDR_HHC_ANALOG_DELAY;
00405     s_command.SIO0Mode         = QSPI_SIO0_IN_ST_EVERY_CMD;
00406
00407     /* Configure the command */
00408     if (HAL_QSPI_Command(&QSPIHandle, &s_command, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00409     {
00410         return QSPI_ERROR;
00411     }
00412
00413     /* Reception of the data */
00414     if (HAL_QSPI_Receive(&QSPIHandle, &reg, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00415     {
00416         return QSPI_ERROR;
00417     }
00418
00419     /* Check the value of the register */
00420     if ((reg & (N25Q128A_FSR_PRERR | N25Q128A_FSR_VPPERR | N25Q128A_FSR_PGERR | N25Q128A_FSR_ERR)) != 0)

```

```
00421     {
00422         return QSPI_ERROR;
00423     }
00424     else if ((reg & (N25Q128A_FSR_PGSUS | N25Q
00425 128A_FSR_ERSUS)) != 0)
00426     {
00427         return QSPI_SUSPENDED;
00428     }
00429     else if ((reg & N25Q128A_FSR_READY) != 0)
00430     {
00431         return QSPI_OK;
00432     }
00433     else
00434     {
00435         return QSPI_BUSY;
00436     }
00437
00438 /**
00439 * @brief Return the configuration of the
00440 * QSPI memory.
00441 * @param pInfo: pointer on the configuration
00442 structure
00443 * @retval QSPI memory status
00444 */
00445 uint8_t BSP_QSPI_GetInfo(QSPI_Info* pInfo)
00446 {
00447     /* Configure the structure with the memory
00448 configuration */
00449     pInfo->FlashSize          = N25Q128A_FLASH
00450 _SIZE;
00451     pInfo->EraseSectorSize    = N25Q128A_SUBSE
00452 CTOR_SIZE;
00453     pInfo->EraseSectorsNumber = (N25Q128A_FLAS
00454 H_SIZE/N25Q128A_SUBSECTOR_SIZE);
00455     pInfo->ProgPageSize      = N25Q128A_PAGE_
00456 SIZE;
```

```
00450     pInfo->ProgPagesNumber      = (N25Q128A_FLASH_SIZE/N25Q128A_PAGE_SIZE);
00451
00452     return QSPI_OK;
00453 }
00454
00455 /**
00456  * @brief Configure the QSPI in memory-map
00457  * ped mode
00458  * @retval QSPI memory status
00459 */
00459 uint8_t BSP_QSPI_EnableMemoryMappedMode(void)
00460 {
00461     QSPI_CommandTypeDef     s_command;
00462     QSPI_MemoryMappedTypeDef s_mem_mapped_cfg;
00463
00464     /* Configure the command for the read instruction */
00465     s_command.InstructionMode    = QSPI_INSTRUCTION_1_LINE;
00466     s_command.Instruction        = QUAD_INOUT_FAST_READ_CMD;
00467     s_command.AddressMode       = QSPI_ADDRESS_4_LINES;
00468     s_command.AddressSize       = QSPI_ADDRESS_24_BITS;
00469     s_command.AlternateByteMode = QSPI_ALTERNATE_BYTES_NONE;
00470     s_command.DataMode          = QSPI_DATA_4_LINES;
00471     s_command.DummyCycles      = N25Q128A_DUMMY_CYCLES_READ_QUAD;
00472     s_command.DdrMode          = QSPI_DDR_MODE_DISABLE;
00473     s_command.DdrHoldHalfCycle = QSPI_DDR_HHC_ANALOG_DELAY;
```

```
00474     s_command.SIO0Mode          = QSPI_SIO0_IN
ST_EVERY_CMD;
00475
00476     /* Configure the memory mapped mode */
00477     s_mem_mapped_cfg.TimeOutActivation = QSPI_
TIMEOUT_COUNTER_DISABLE;
00478     s_mem_mapped_cfg.TimeOutPeriod      = 0;
00479
00480     if (HAL_QSPI_MemoryMapped(&QSPIHandle, &s_
command, &s_mem_mapped_cfg) != HAL_OK)
00481     {
00482         return QSPI_ERROR;
00483     }
00484
00485     return QSPI_OK;
00486 }
00487
00488 /**
00489 * @}
00490 */
00491
00492 /** @addtogroup STM32746G_DISCOVERY_QSPI_Pri
vate_Functions
00493 * @{
00494 */
00495
00496 /**
00497 * @brief QSPI MSP Initialization
00498 *       This function configures the hard
ware resources used in this example:
00499 *           - Peripheral's clock enable
00500 *           - Peripheral's GPIO Configurat
ion
00501 *           - NVIC configuration for QSPI
interrupt
00502 * @retval None
00503 */
```

```
00504 __weak void BSP_QSPI_MspInit(QSPI_HandleTypeDef
Def *hqspi, void *Params)
00505 {
00506     GPIO_InitTypeDef gpio_init_structure;
00507
00508     /*##-1- Enable peripherals and GPIO Clocks
#####
00509     /* Enable the QuadSPI memory interface clock */
00510     QSPI_CLK_ENABLE();
00511     /* Reset the QuadSPI memory interface */
00512     QSPI_FORCE_RESET();
00513     QSPI_RELEASE_RESET();
00514     /* Enable GPIO clocks */
00515     QSPI_CS_GPIO_CLK_ENABLE();
00516     QSPI_CLK_GPIO_CLK_ENABLE();
00517     QSPI_D0_GPIO_CLK_ENABLE();
00518     QSPI_D1_GPIO_CLK_ENABLE();
00519     QSPI_D2_GPIO_CLK_ENABLE();
00520     QSPI_D3_GPIO_CLK_ENABLE();
00521
00522     /*##-2- Configure peripheral GPIO #####
#####
00523     /* QSPI CS GPIO pin configuration */
00524     gpio_init_structure.Pin      = QSPI_CS_PIN
;
00525     gpio_init_structure.Mode    = GPIO_MODE_
AF_PP;
00526     gpio_init_structure.Pull    = GPIO_PULLU
P;
00527     gpio_init_structure.Speed   = GPIO_SPEED
_HIGH;
00528     gpio_init_structure.Alternate = GPIO_AF10_
QUADSPI;
00529     HAL_GPIO_Init(QSPI_CS_GPIO_PORT, &gpio_ini
t_structure);
00530
```

```
00531 /* QSPI CLK GPIO pin configuration */
00532 gpio_init_structure.Pin      = QSPI_CLK_P
IN;
00533 gpio_init_structure.Pull    = GPIO_NOPUL
L;
00534 gpio_init_structure.Alternate = GPIO_AF9_Q
UADSPI;
00535 HAL_GPIO_Init(QSPI_CLK_GPIO_PORT, &gpio_in
it_structure);
00536
00537 /* QSPI D0 GPIO pin configuration */
00538 gpio_init_structure.Pin      = QSPI_D0_PIN
;
00539 gpio_init_structure.Alternate = GPIO_AF9_Q
UADSPI;
00540 HAL_GPIO_Init(QSPI_D0_GPIO_PORT, &gpio_ini
t_structure);
00541
00542 /* QSPI D1 GPIO pin configuration */
00543 gpio_init_structure.Pin      = QSPI_D1_PIN
;
00544 gpio_init_structure.Alternate = GPIO_AF9_Q
UADSPI;
00545 HAL_GPIO_Init(QSPI_D1_GPIO_PORT, &gpio_in
it_structure);
00546
00547 /* QSPI D2 GPIO pin configuration */
00548 gpio_init_structure.Pin      = QSPI_D2_PIN
;
00549 gpio_init_structure.Alternate = GPIO_AF9_Q
UADSPI;
00550 HAL_GPIO_Init(QSPI_D2_GPIO_PORT, &gpio_in
it_structure);
00551
00552 /* QSPI D3 GPIO pin configuration */
00553 gpio_init_structure.Pin      = QSPI_D3_PIN
;
```

```
00554     gpio_init_structure.Alternate = GPIO_AF9_Q  
UADSPI;  
00555     HAL_GPIO_Init(QSPI_D3_GPIO_PORT, &gpio_ini  
t_structure);  
00556  
00557     /*##-3- Configure the NVIC for QSPI #####  
#####*/  
00558     /* NVIC configuration for QSPI interrupt */  
  
00559     HAL_NVIC_SetPriority(QUADSPI IRQn, 0x0F, 0  
);  
00560     HAL_NVIC_EnableIRQ(QUADSPI IRQn);  
00561 }  
00562  
00563 /**  
00564     * @brief QSPI MSP De-Initialization  
00565     *         This function frees the hardware  
resources used in this example:  
00566     *             - Disable the Peripheral's clock  
  
00567     *             - Revert GPIO and NVIC configura  
tion to their default state  
00568     * @retval None  
00569     */  
00570 __weak void BSP_QSPI_MspDeInit(QSPI_HandleTypeDef  
*hqspi, void *Params)  
00571 {  
00572     /*##-1- Disable the NVIC for QSPI #####  
#####*/  
00573     HAL_NVIC_DisableIRQ(QUADSPI IRQn);  
00574  
00575     /*##-2- Disable peripherals and GPIO Clock  
s #####*/  
00576     /* De-Configure QSPI pins */  
00577     HAL_GPIO_DeInit(QSPI_CS_GPIO_PORT, QSPI_CS  
_PIN);  
00578     HAL_GPIO_DeInit(QSPI_CLK_GPIO_PORT, QSPI_C
```

```
    LK_PIN);
00579     HAL_GPIO_DeInit(QSPI_D0_GPIO_PORT, QSPI_D0
    _PIN);
00580     HAL_GPIO_DeInit(QSPI_D1_GPIO_PORT, QSPI_D1
    _PIN);
00581     HAL_GPIO_DeInit(QSPI_D2_GPIO_PORT, QSPI_D2
    _PIN);
00582     HAL_GPIO_DeInit(QSPI_D3_GPIO_PORT, QSPI_D3
    _PIN);
00583
00584     /*##-3- Reset peripherals #####
#####*/
00585     /* Reset the QuadSPI memory interface */
00586     QSPI_FORCE_RESET();
00587     QSPI_RELEASE_RESET();
00588
00589     /* Disable the QuadSPI memory interface cl
ock */
00590     QSPI_CLK_DISABLE();
00591 }
00592
00593 /**
00594     * @brief This function reset the QSPI mem
ory.
00595     * @param hqspi: QSPI handle
00596     * @retval None
00597 */
00598 static uint8_t QSPI_ResetMemory(QSPI_HandleTypeDef
    *hqspi)
00599 {
00600     QSPI_CommandTypeDef s_command;
00601
00602     /* Initialize the reset enable command */
00603     s_command.InstructionMode      = QSPI_INSTRU
    CTION_1_LINE;
00604     s_command.Instruction        = RESET_ENABLE
    _CMD;
```

```
00605     s_command.AddressMode      = QSPI_ADDRESS
00606     _NONE;
00607     s_command.AlternateByteMode = QSPI_ALTERNA
00608     TE_BYTES_NONE;
00609     s_command.DataMode        = QSPI_DATA_NO
00610     NE;
00611     s_command.DummyCycles    = 0;
00612     s_command.DdrMode         = QSPI_DDR_MOD
00613     E_DISABLE;
00614     s_command.DdrHoldHalfCycle = QSPI_DDR_HHC
00615     _ANALOG_DELAY;
00616     s_command.SI00Mode        = QSPI_SI00_IN
00617     ST_EVERY_CMD;
00618
00619     /* Send the command */
00620     if (HAL_QSPI_Command(hqspi, &s_command, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00621     {
00622         return QSPI_ERROR;
00623     }
00624
00625
00626     /* Send the reset memory command */
00627     s_command.Instruction = RESET_MEMORY_CMD;
00628     if (HAL_QSPI_Command(hqspi, &s_command, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00629     {
00630         return QSPI_ERROR;
00631     }
```

```
00632     return QSPI_OK;
00633 }
00634
00635 /**
00636  * @brief This function configure the dummy cycles on memory side.
00637  * @param hqspi: QSPI handle
00638  * @retval None
00639 */
00640 static uint8_t QSPI_DummyCyclesCfg(QSPI_HandleTypeDefDef *hqspi)
00641 {
00642     QSPI_CommandTypeDef s_command;
00643     uint8_t reg;
00644
00645     /* Initialize the read volatile configuration register command */
00646     s_command.InstructionMode      = QSPI_INSTRUCTION_1_LINE;
00647     s_command.Instruction          = READ_VOL_CFG_REG_CMD;
00648     s_command.AddressMode         = QSPI_ADDRESS_NONE;
00649     s_command.AlternateByteMode   = QSPI_ALTERNA_TE_BYTES_NONE;
00650     s_command.DataMode            = QSPI_DATA_1_LINE;
00651     s_command.DummyCycles        = 0;
00652     s_command.NbData              = 1;
00653     s_command.DdrMode             = QSPI_DDR_MODE_DISABLE;
00654     s_command.DdrHoldHalfCycle   = QSPI_DDR_HHC_ANALOG_DELAY;
00655     s_command.SI00Mode            = QSPI_SI00_IN_ST_EVERY_CMD;
00656
00657     /* Configure the command */
```

```
00658     if (HAL_QSPI_Command(hqspi, &s_command, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00659     {
00660         return QSPI_ERROR;
00661     }
00662
00663     /* Reception of the data */
00664     if (HAL_QSPI_Receive(hqspi, &reg, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00665     {
00666         return QSPI_ERROR;
00667     }
00668
00669     /* Enable write operations */
00670     if (QSPI_WriteEnable(hqspi) != QSPI_OK)
00671     {
00672         return QSPI_ERROR;
00673     }
00674
00675     /* Update volatile configuration register
00676      (with new dummy cycles) */
00677     s_command.Instruction = WRITE_VOL_CFG_REG_CMD;
00678     MODIFY_REG(reg, N25Q128A_VCR_NB_DUMMY, (N25Q128A_DUMMY_CYCLES_READ_QUAD << POSITION_VAL(N25Q128A_VCR_NB_DUMMY)));
00679
00680     /* Configure the write volatile configuration register command */
00681     if (HAL_QSPI_Command(hqspi, &s_command, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00682     {
00683         return QSPI_ERROR;
00684     }
00685
00686     /* Transmission of the data */
00687     if (HAL_QSPI_Transmit(hqspi, &reg, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
```

```
I_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00687 {
00688     return QSPI_ERROR;
00689 }
00690
00691     return QSPI_OK;
00692 }
00693
00694 /**
00695     * @brief This function send a Write Enable and wait it is effective.
00696     * @param hqspi: QSPI handle
00697     * @retval None
00698 */
00699 static uint8_t QSPI_WriteEnable(QSPI_HandleTypeDefDef *hqspi)
00700 {
00701     QSPI_CommandTypeDef     s_command;
00702     QSPI_AutoPollingTypeDef s_config;
00703
00704     /* Enable write operations */
00705     s_command.InstructionMode    = QSPI_INSTRUCTION_1_LINE;
00706     s_command/Instruction        = WRITE_ENABLE_CMD;
00707     s_command.AddressMode       = QSPI_ADDRESS_NONE;
00708     s_command.AlternateByteMode = QSPI_ALTERNA_TE_BYTES_NONE;
00709     s_command.DataMode          = QSPI_DATA_NO_NE;
00710     s_command.DummyCycles      = 0;
00711     s_command.DdrMode          = QSPI_DDR_MODE_DISABLE;
00712     s_command.DdrHoldHalfCycle = QSPI_DDR_HHC_ANALOG_DELAY;
00713     s_command.SIO0Mode         = QSPI_SIO0_IN
```

```
ST_EVERY_CMD;
00714
00715     if (HAL_QSPI_Command(hqspi, &s_command, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00716     {
00717         return QSPI_ERROR;
00718     }
00719
00720     /* Configure automatic polling mode to wait for write enabling */
00721     s_config.Match          = N25Q128A_SR_WREN;
00722     s_config.Mask           = N25Q128A_SR_WREN;
00723     s_config.MatchMode      = QSPI_MATCH_MODE_AND;
00724     s_config.StatusBytesSize = 1;
00725     s_config.Interval        = 0x10;
00726     s_config.AutomaticStop   = QSPI_AUTOMATIC_STOP_ENABLE;
00727
00728     s_command.Instruction    = READ_STATUS_REG_CMD;
00729     s_command.DataMode       = QSPI_DATA_1_LIN_E;
00730
00731     if (HAL_QSPI_AutoPolling(hqspi, &s_command, &s_config, HAL_QPSI_TIMEOUT_DEFAULT_VALUE) != HAL_OK)
00732     {
00733         return QSPI_ERROR;
00734     }
00735
00736     return QSPI_OK;
00737 }
00738
00739 /**
```

```
00740     * @brief This function read the SR of the
00741     * memory and wait the EOP.
00742     * @param hqspi: QSPI handle
00743     * @param Timeout
00744     * @retval None
00745     */
00746 static uint8_t QSPI_AutoPollingMemReady(QSPI
00747 _HandleTypeDef *hqspi, uint32_t Timeout)
00748 {
00749     QSPI_CommandTypeDef      s_command;
00750     QSPI_AutoPollingTypeDef s_config;
00751
00752     /* Configure automatic polling mode to wait for memory ready */
00753     s_command.InstructionMode    = QSPI_INSTRU
00754     TION_1_LINE;
00755     s_command.Instruction        = READ_STATUS_
00756     REG_CMD;
00757     s_command.AddressMode       = QSPI_ADDRESS
00758     _NONE;
00759     s_command.AlternateByteMode = QSPI_ALTERNA
00760     TE_BYTES_NONE;
00761     s_command.DataMode          = QSPI_DATA_1_
00762     LINE;
00763     s_command.DummyCycles      = 0;
00764     s_command.DdrMode          = QSPI_DDR_MOD
00765     E_DISABLE;
00766     s_command.DdrHoldHalfCycle = QSPI_DDR_HHC
00767     _ANALOG_DELAY;
00768     s_command.SI00Mode         = QSPI_SI00_IN
00769     ST_EVERY_CMD;
00770
00771     s_config.Match              = 0;
00772     s_config.Mask               = N25Q128A_SR_WIP
00773 ;
00774     s_config.MatchMode          = QSPI_MATCH_MODE
00775     _AND;
```

```
00764     s_config.StatusBytesSize = 1;
00765     s_config.Interval        = 0x10;
00766     s_config.AutomaticStop   = QSPI_AUTOMATIC_
STOP_ENABLE;
00767
00768     if (HAL_QSPI_AutoPolling(hspi, &s_command
, &s_config, Timeout) != HAL_OK)
00769     {
00770         return QSPI_ERROR;
00771     }
00772
00773     return QSPI_OK;
00774 }
00775 /**
00776 * @}
00777 */
00778
00779 /**
00780 * @}
00781 */
00782
00783 /**
00784 * @}
00785 */
00786
00787 /**
00788 * @}
00789 */
00790
00791 ***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/
00792
```

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules Defines		

STM32746G_DISCOVERY_SD

[STM32746G_DISCOVERY](#)

Modules

[**STM32746G_DISCOVERY_SD Private Types Definitions**](#)
[**STM32746G_DISCOVERY_SD Private Defines**](#)
[**STM32746G_DISCOVERY_SD Private Macros**](#)
[**STM32746G_DISCOVERY_SD Private Variables**](#)
[**STM32746G_DISCOVERY_SD Private Function Prototypes**](#)
[**STM32746G_DISCOVERY_SD Exported Functions**](#)
[**STM32746G_DISCOVERY_SD Exported Types**](#)
[**STM32746G_DISCOVERY_SD Exported Constants**](#)
[**STM32746G_DISCOVERY_SD Exported Macro**](#)

Defines

```
#define MSD_OK ((uint8_t)0x00)
    SD status structure definition.

#define MSD_ERROR ((uint8_t)0x01)

#define MSD_ERROR_SD_NOT_PRESENT ((uint8_t)0x02)

#define SD_TRANSFER_OK ((uint8_t)0x00)
    SD transfer state definition.

#define SD_TRANSFER_BUSY ((uint8_t)0x01)
```

Define Documentation

#define MSD_ERROR ((uint8_t)0x01)

Definition at line [78](#) of file [stm32746g_discovery_sd.h](#).

Referenced by [BSP_SD_DelInit\(\)](#), [BSP_SD_Erase\(\)](#), [BSP_SD_Init\(\)](#), [BSP_SD_ReadBlocks\(\)](#), [BSP_SD_ReadBlocks_DMA\(\)](#), [BSP_SD_WriteBlocks\(\)](#), and [BSP_SD_WriteBlocks_DMA\(\)](#).

#define MSD_ERROR_SD_NOT_PRESENT ((uint8_t)0x02)

Definition at line [79](#) of file [stm32746g_discovery_sd.h](#).

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

#define MSD_OK ((uint8_t)0x00)

SD status structure definition.

Definition at line [77](#) of file [stm32746g_discovery_sd.h](#).

Referenced by [BSP_SD_DelInit\(\)](#), [BSP_SD_Erase\(\)](#), [BSP_SD_Init\(\)](#), [BSP_SD_ITConfig\(\)](#), [BSP_SD_ReadBlocks\(\)](#), [BSP_SD_ReadBlocks_DMA\(\)](#), [BSP_SD_WriteBlocks\(\)](#), and [BSP_SD_WriteBlocks_DMA\(\)](#).

#define SD_TRANSFER_BUSY ((uint8_t)0x01)

Definition at line [85](#) of file [stm32746g_discovery_sd.h](#).

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

#define SD_TRANSFER_OK ((uint8_t)0x00)

SD transfer state definition.

Definition at line **84** of file [stm32746g_discovery_sd.h](#).

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

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules		

STM32746G_DISCOVERY_SDRAM

[STM32746G_DISCOVERY](#)

Modules

[**STM32746G_DISCOVERY_SDRAM Private Types Definitions**](#)

[**STM32746G_DISCOVERY_SDRAM Private Defines**](#)

[**STM32746G_DISCOVERY_SDRAM Private Macros**](#)

[**STM32746G_DISCOVERY_SDRAM Private Variables**](#)

[**STM32746G_DISCOVERY_SDRAM Private Function Prototypes**](#)

[**STM32746G_DISCOVERY_SDRAM Exported Functions**](#)

[**STM32746G_DISCOVERY_SDRAM Exported Types**](#)

[**STM32746G_DISCOVERY_SDRAM Exported Macro**](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [**doxygen**](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules		

STM32746G_DISCOVERY_TS

[STM32746G_DISCOVERY](#)

Modules

[STM32746G_DISCOVERY_TS Types Definitions](#)
[STM32746G_DISCOVERY_TS Types Defines](#)
[STM32746G_DISCOVERY_TS Private Macros](#)
[STM32746G_DISCOVERY_TS Imported Variables](#)
[STM32746G_DISCOVERY_TS Private Variables](#)
[STM32746G_DISCOVERY_TS Private Function Prototypes](#)
[STM32746G_DISCOVERY_TS Exported Functions](#)
[STM32746G_DISCOVERY_TS Exported Constants](#)
[STM32746G_DISCOVERY_TS Exported Types](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by  1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Data Structures

STM32746G_DISCOVERY_TS Exported Types

[STM32746G_DISCOVERY_TS](#)

Data Structures

struct **TS_StateTypeDef**

TS_StateTypeDef Define TS State structure. More...

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_ts.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_ts.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains the common d
efines and functions prototypes for
00008 *           the stm32746g_discovery_ts.c dr
iver.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and bin
ary forms, with or without modification,
00015 * are permitted provided that the followin
g conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038      */
00039 /* Define to prevent recursive inclusion ---
00040 -----
00041 #ifndef __STM32746G_DISCOVERY_TS_H
00042 #define __STM32746G_DISCOVERY_TS_H
00043
00044 #ifdef __cplusplus
00045     extern "C" {
00046 #endif
00047
00048 /* Includes -----
00049 -----
00050 #include "stm32746g_discovery.h"
00051 /* Include touch screen FT5336 component Driver */
00052 #include "../Components/ft5336/ft5336.h"
00053
00054 /** @addtogroup BSP
00055     @{
00056
00057 /** @addtogroup STM32746G_DISCOVERY
00058     @{
00059
00060 /** @addtogroup STM32746G_DISCOVERY_TS
00061     @{
00062
00063
00064 /** @defgroup STM32746G_DISCOVERY_TS_Exported_Constants STM32746G_DISCOVERY_TS Exported Constants
00065     @{
00066
```

```
00066      */
00067
00068 /** @brief With FT5336 : maximum 5 touches detected simultaneously
00069 */
00070 #define TS_MAX_NB_TOUCH ((uint32_t) FT5336_MAX_DETECTABLE_TOUCH)
00071
00072 #define TS_NO_IRQ_PENDING ((uint8_t) 0)
00073 #define TS_IRQ_PENDING ((uint8_t) 1)
00074
00075 #define TS_SWAP_NONE ((uint8_t) 0x01)
00076 #define TS_SWAP_X ((uint8_t) 0x02)
00077 #define TS_SWAP_Y ((uint8_t) 0x04)
00078 #define TS_SWAP_XY ((uint8_t) 0x08)
00079
00080 /**
00081 * @}
00082 */
00083
00084 /** @defgroup STM32746G_DISCOVERY_TS_Exported_Types STM32746G_DISCOVERY_TS Exported Types
00085 */
00086 */
00087 /**
00088 * @brief TS_StateTypeDef
00089 * Define TS State structure
00090 */
00091 typedef struct
00092 {
00093     uint8_t touchDetected; /*!
```

```

< Total number of active touches detected at last
scan */
00094     uint16_t touchX[TS_MAX_NB_TOUCH];      /* !
< Touch X[0], X[1] coordinates on 12 bits */
00095     uint16_t touchY[TS_MAX_NB_TOUCH];      /* !
< Touch Y[0], Y[1] coordinates on 12 bits */
00096
00097 #if (TS_MULTI_TOUCH_SUPPORTED == 1)
00098     uint8_t  touchWeight[TS_MAX_NB_TOUCH]; /* !
< Touch_Weight[0], Touch_Weight[1] : weight proper
ty of touches */
00099     uint8_t  touchEventId[TS_MAX_NB_TOUCH];
    /*!< Touch_EventId[0], Touch_EventId[1] : take v
alue of type @ref TS_TouchEventTypeDef */
00100    uint8_t  touchArea[TS_MAX_NB_TOUCH];      /* !
< Touch_Area[0], Touch_Area[1] : touch area of eac
h touch */
00101    uint32_t gestureId; /*!< type of gesture d
etected : take value of type @ref TS_GestureIdType
Def */
00102 #endif /* TS_MULTI_TOUCH_SUPPORTED == 1 */
00103
00104 } TS_StateTypeDef;
00105
00106 /**
00107 * @}
00108 */
00109
00110 /** @defgroup STM32746G_DISCOVERY_TS_Expor
ted_Constants STM32746G_DISCOVERY_TS Exported Consta
nts
00111 * @{
00112 */
00113
00114 typedef enum
00115 {
00116     TS_OK
00117     = 0x00, /*!< Touch Ok

```

```

/*
00117   TS_ERROR          = 0x01, /*!< Touch Error */
00118   TS_TIMEOUT         = 0x02, /*!< Touch Timeout */
00119   TS_DEVICE_NOT_FOUND = 0x03 /*!< Touchscreen device not found */
00120 }TS_StatusTypeDef;
00121
00122 /**
00123 * @brief TS_GestureIdTypeDef
00124 * Define Possible managed gesture identification values returned by touch screen
00125 * driver.
00126 */
00127 typedef enum
00128 {
00129   GEST_ID_NO_GESTURE = 0x00, /*!< Gesture not defined / recognized */
00130   GEST_ID_MOVE_UP     = 0x01, /*!< Gesture Move Up */
00131   GEST_ID_MOVE_RIGHT  = 0x02, /*!< Gesture Move Right */
00132   GEST_ID_MOVE_DOWN   = 0x03, /*!< Gesture Move Down */
00133   GEST_ID_MOVE_LEFT   = 0x04, /*!< Gesture Move Left */
00134   GEST_ID_ZOOM_IN     = 0x05, /*!< Gesture Zoom In */
00135   GEST_ID_ZOOM_OUT    = 0x06, /*!< Gesture Zoom Out */
00136   GEST_ID_NB_MAX      = 0x07 /*!< max number of gesture id */
00137
00138 } TS_GestureIdTypeDef;
00139
00140 /**

```

```
00141 * @brief TS_TouchEventTypeDef
00142 * Define Possible touch events kind as returned values
00143 * by touch screen IC Driver.
00144 */
00145 typedef enum
00146 {
00147     TOUCH_EVENT_NO_EVT          = 0x00, /* !< Touch Event : undetermined */
00148     TOUCH_EVENT_PRESS_DOWN      = 0x01, /* !< Touch Event Press Down */
00149     TOUCH_EVENT_LIFT_UP          = 0x02, /* !< Touch Event Lift Up */
00150     TOUCH_EVENT_CONTACT         = 0x03, /* !< Touch Event Contact */
00151     TOUCH_EVENT_NB_MAX          = 0x04 /* !< max number of touch events kind */
00152
00153 } TS_TouchEventTypeDef;
00154 /**
00155 * @}
00156 */
00157
00158 /** @defgroup STM32746G_DISCOVERY_TS_Imported_Variables STM32746G_DISCOVERY_TS Imported Variables
00159 * @{
00160 */
00161 /**
00162 * @brief Table for touchscreen event information display on LCD :
00163 * table indexed on enum @ref TS_TouchEventTypeDef information
00164 */
00165 extern char * ts_event_string_tab[TOUCH_EVENT_NB_MAX];
00166
```

```
00167 /**
00168  * @brief Table for touchscreen gesture Id
00169  * information display on LCD : table indexed
00170  * on enum @ref TS_GestureIdTypeDef informa
00171  * tion
00172 */
00173  * @}
00174 */
00175
00176 /** @addtogroup STM32746G_DISCOVERY_TS_Expor
00177  * ted_Functions
00178 */
00179 uint8_t BSP_TS_Init(uint16_t ts_SizeX, uint1
00180 6_t ts_SizeY);
00180 uint8_t BSP_TS_DeInit(void);
00181 uint8_t BSP_TS_GetState(TS_StateTypeDef *TS_
00182 State);
00183 #if (TS_MULTI_TOUCH_SUPPORTED == 1)
00184 uint8_t BSP_TS_Get_GestureId(TS_StateTypeDef
00185 *TS_State);
00185 #endif /* TS_MULTI_TOUCH_SUPPORTED == 1 */
00186
00187 uint8_t BSP_TS_ITConfig(void);
00188 uint8_t BSP_TS_ITGetStatus(void);
00189 void BSP_TS_ITClear(void);
00190 uint8_t BSP_TS_ResetTouchData(TS_StateTypeDef
00191 *TS_State);
00191 /**
00192  * @}
00193 */
00194
00195 /**
```

```
00196     * @}
00197     */
00198
00199 /**
00200     * @}
00201     */
00202
00203 /**
00204     * @}
00205     */
00206
00207
00208 #ifdef __cplusplus
00209 }
00210 #endif
00211
00212 #endif /* __STM32746G_DISCOVERY_TS_H */
00213
00214 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Defines Enumerations		
STM32746G_DISCOVERY_TS Exported Constants	STM32746G_DISCOVERY_TS		

Defines

```
#define TS_MAX_NB_TOUCH ((uint32_t)  
FT5336_MAX_DETECTABLE_TOUCH)  
With FT5336 : maximum 5 touches detected simultaneously.  
  
#define TS_NO_IRQ_PENDING ((uint8_t) 0)  
#define TS_IRQ_PENDING ((uint8_t) 1)  
#define TS_SWAP_NONE ((uint8_t) 0x01)  
#define TS_SWAP_X ((uint8_t) 0x02)  
#define TS_SWAP_Y ((uint8_t) 0x04)  
#define TS_SWAP_XY ((uint8_t) 0x08)
```

Enumerations

enum	<pre>TS_StatusTypeDef { TS_OK = 0x00, TS_ERROR = 0x01, TS_TIMEOUT = 0x02, TS_DEVICE_NOT_FOUND = 0x03 }</pre>
enum	<pre>TS_GestureIdTypeDef { GEST_ID_NO_GESTURE = 0x00, GEST_ID_MOVE_UP = 0x01, GEST_ID_MOVE_RIGHT = 0x02, GEST_ID_MOVE_DOWN = 0x03, GEST_ID_MOVE_LEFT = 0x04, GEST_ID_ZOOM_IN = 0x05, GEST_ID_ZOOM_OUT = 0x06, GEST_ID_NB_MAX = 0x07 }</pre> <p>TS_GestureIdTypeDef Define Possible managed gesture identification values returned by touch screen driver. More...</p>
enum	<pre>TS_TouchEventTypeDef { TOUCH_EVENT_NO_EVT = 0x00, TOUCH_EVENT_PRESS_DOWN = 0x01, TOUCH_EVENT_LIFT_UP = 0x02, TOUCH_EVENT_CONTACT = 0x03, TOUCH_EVENT_NB_MAX = 0x04 }</pre> <p>TS_TouchEventTypeDef Define Possible touch events kind as returned values by touch screen IC Driver. More...</p>

Define Documentation

```
#define TS_IRQ_PENDING ((uint8_t) 1)
```

Definition at line [73](#) of file [stm32746g_discovery_ts.h](#).

```
#define TS_MAX_NB_TOUCH ((uint32_t) FT5336_MAX_DETECTAE)
```

With FT5336 : maximum 5 touches detected simultaneously.

Definition at line [70](#) of file [stm32746g_discovery_ts.h](#).

Referenced by [BSP_TS_GetState\(\)](#), and
[BSP_TS_ResetTouchData\(\)](#).

```
#define TS_NO_IRQ_PENDING ((uint8_t) 0)
```

Definition at line [72](#) of file [stm32746g_discovery_ts.h](#).

```
#define TS_SWAP_NONE ((uint8_t) 0x01)
```

Definition at line [75](#) of file [stm32746g_discovery_ts.h](#).

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

```
#define TS_SWAP_X ((uint8_t) 0x02)
```

Definition at line [76](#) of file [stm32746g_discovery_ts.h](#).

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

```
#define TS_SWAP_XY ((uint8_t) 0x08)
```

Definition at line [78](#) of file [stm32746g_discovery_ts.h](#).

Referenced by [BSP_TS_GetState\(\)](#), and [BSP_TS_Init\(\)](#).

```
#define TS_SWAP_Y ((uint8_t) 0x04)
```

Definition at line [77](#) of file [stm32746g_discovery_ts.h](#).

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

Enumeration Type Documentation

enum TS_GestureIdTypeDef

TS_GestureIdTypeDef Define Possible managed gesture identification values returned by touch screen driver.

Enumerator:

GEST_ID_NO_GESTURE Gesture not defined / recognized

GEST_ID_MOVE_UP Gesture Move Up

GEST_ID_MOVE_RIGHT Gesture Move Right

GEST_ID_MOVE_DOWN Gesture Move Down

GEST_ID_MOVE_LEFT Gesture Move Left

GEST_ID_ZOOM_IN Gesture Zoom In

GEST_ID_ZOOM_OUT Gesture Zoom Out

GEST_ID_NB_MAX max number of gesture id

Definition at line [127](#) of file [stm32746g_discovery_ts.h](#).

enum TS_StatusTypeDef

Enumerator:

<i>TS_OK</i>	Touch Ok
<i>TS_ERROR</i>	Touch Error
<i>TS_TIMEOUT</i>	Touch Timeout
<i>TS_DEVICE_NOT_FOUND</i>	Touchscreen device not found

Definition at line [114](#) of file [stm32746g_discovery_ts.h](#).

enum TS_TouchEventTypeDef

`TS_TouchEventTypeDef` Define Possible touch events kind as returned values by touch screen IC Driver.

Enumerator:

<i>TOUCH_EVENT_NO_EVT</i>	Touch Event : undetermined
<i>TOUCH_EVENT_PRESS_DOWN</i>	Touch Event Press Down
<i>TOUCH_EVENT_LIFT_UP</i>	Touch Event Lift Up
<i>TOUCH_EVENT_CONTACT</i>	Touch Event Contact
<i>TOUCH_EVENT_NB_MAX</i>	max number of touch events kind

Definition at line **145** of file **stm32746g_discovery_ts.h**.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_ts.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_ts.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file provides a set of functions needed to manage the Touch
00008 *           Screen on STM32746G-Discovery board.
00009 @verbatim
00010 1. How To use this driver:
00011 -----
00012 - This driver is used to drive the touch screen module of the STM32746G-Discovery
00013 board on the RK043FN48H-CT672B 480x272 LCD screen with capacitive touch screen.
00014 - The FT5336 component driver must be included in project files according to
00015 the touch screen driver present on this board.
```

```
00016
00017      2. Driver description:
00018      -----
00019          + Initialization steps:
00020              o Initialize the TS module using the
00021                  BSP_TS_Init() function. This
00022                      function includes the MSP layer ha
00023      rdware resources initialization and the
00024      communication layer configuration
00025      to start the TS use. The LCD size properties
00026          (x and y) are passed as parameters.

00027          o If TS interrupt mode is desired, you
00028      must configure the TS interrupt mode
00029          by calling the function BSP_TS_ITC
00030      onfig(). The TS interrupt mode is generated
00031          as an external interrupt whenever
00032      a touch is detected.

00033          The interrupt mode internally uses
00034      the IO functionalities driver driven by
00035          the IO expander, to configure the
IT line.

00036
00037          + Touch screen use
00038              o The touch screen state is captured
00039      whenever the function BSP_TS_GetState() is
00040          used. This function returns inform
00041      ation about the last LCD touch occurred
00042          in the TS_StateTypeDef structure.
00043              o If TS interrupt mode is used, the
00044      function BSP_TS_ITGetStatus() is needed to get
00045          the interrupt status. To clear the
00046      IT pending bits, you should call the
00047          function BSP_TS_ITClear().
00048              o The IT is handled using the corres
00049      ponding external interrupt IRQ handler,
00050          the user IT callback treatment is
```

```
implemented on the same external interrupt
00039          callback.
00040      @endverbatim
00041  ****
*****  

00042  * @attention
00043  *
00044  * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00045  *
00046  * Redistribution and use in source and bin
ary forms, with or without modification,
00047  * are permitted provided that the followin
g conditions are met:
00048  *   1. Redistributions of source code must
    retain the above copyright notice,
00049  *       this list of conditions and the fol
lowing disclaimer.
00050  *   2. Redistributions in binary form must
    reproduce the above copyright notice,
00051  *       this list of conditions and the fol
lowing disclaimer in the documentation
00052  *       and/or other materials provided wit
h the distribution.
00053  *   3. Neither the name of STMicroelectron
ics nor the names of its contributors
00054  *       may be used to endorse or promote p
roducts derived from this software
00055  *       without specific prior written perm
ission.
00056  *
00057  * THIS SOFTWARE IS PROVIDED BY THE COPYRIG
HT HOLDERS AND CONTRIBUTORS "AS IS"
00058  * AND ANY EXPRESS OR IMPLIED WARRANTIES, I
NCLUDING, BUT NOT LIMITED TO, THE
00059  * IMPLIED WARRANTIES OF MERCHANTABILITY AN
D FITNESS FOR A PARTICULAR PURPOSE ARE
```

```
00060      * DISCLAIMED. IN NO EVENT SHALL THE COPYRI
GHT HOLDER OR CONTRIBUTORS BE LIABLE
00061      * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP
ECIAL, EXEMPLARY, OR CONSEQUENTIAL
00062      * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR
00063      * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER
00064      * CAUSED AND ON ANY THEORY OF LIABILITY, W
HETHER IN CONTRACT, STRICT LIABILITY,
00065      * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI
SE) ARISING IN ANY WAY OUT OF THE USE
00066      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
00067      *
00068      ****
00069      */
00070
00071 /* Includes -----
----- */
00072 #include "stm32746g_discovery_ts.h"
00073
00074 /** @addtogroup BSP
00075     * @{
00076     */
00077
00078 /** @addtogroup STM32746G_DISCOVERY
00079     * @{
00080     */
00081
00082 /** @defgroup STM32746G_DISCOVERY_TS STM3274
6G_DISCOVERY_TS
00083     * @{
00084     */
00085
00086 /** @defgroup STM32746G_DISCOVERY_TS_Private
```

```
_Types_Definitions STM32746G_DISCOVERY_TS Types De
finitions
00087      * @@
00088      */
00089 /**
00090      * @@
00091      */
00092
00093 /** @defgroup STM32746G_DISCOVERY_TS_Private
_Defines STM32746G_DISCOVERY_TS Types Defines
00094      * @@
00095      */
00096 /**
00097      * @@
00098      */
00099
00100 /** @defgroup STM32746G_DISCOVERY_TS_Private
_Macros STM32746G_DISCOVERY_TS Private Macros
00101      * @@
00102      */
00103 /**
00104      * @@
00105      */
00106
00107 /** @defgroup STM32746G_DISCOVERY_TS_Importe
d_Variables STM32746G_DISCOVERY_TS Imported Variab
les
00108      * @@
00109      */
00110 /**
00111      * @@
00112      */
00113
00114 /** @defgroup STM32746G_DISCOVERY_TS_Private
_Variables STM32746G_DISCOVERY_TS Private Variables
00115      * @@
```

```
00116     */
00117 static TS_DrvTypeDef *tsDriver;
00118 static uint16_t tsXBoundary, tsYBoundary;
00119 static uint8_t tsOrientation;
00120 static uint8_t I2cAddress;
00121 /**
00122  * @}
00123 */
00124
00125 /** @defgroup STM32746G_DISCOVERY_TS_Private
00126 _Function_Protoypes STM32746G_DISCOVERY_TS Private Function Prototypes
00127 */
00128 /**
00129  * @}
00130 */
00131
00132 /** @defgroup STM32746G_DISCOVERY_TS_Exports
00133 _Functions STM32746G_DISCOVERY_TS Exported Functions
00134 */
00135
00136 /**
00137  * @brief Initializes and configures the touch screen functionalities and
00138  *        configures all necessary hardware resources (GPIOs, I2C, clocks..).
00139  * @param ts_SizeX: Maximum X size of the TS area on LCD
00140  * @param ts_SizeY: Maximum Y size of the TS area on LCD
00141  * @retval TS_OK if all initializations are OK. Other value if error.
00142 */
00143 uint8_t BSP_TS_Init(uint16_t ts_SizeX, uint1
```

```
00143     6_t ts_SizeY)
00144 {
00145     uint8_t status = TS_OK;
00146     tsXBoundary = ts_SizeX;
00147     tsYBoundary = ts_SizeY;
00148
00149     /* Read ID and verify if the touch screen
00150      driver is ready */
00150     ft5336_ts_drv.Init(TS_I2C_ADDRESS);
00151     if(ft5336_ts_drv.ReadID(TS_I2C_ADDRESS) ==
00152         FT5336_ID_VALUE)
00152     {
00153         /* Initialize the TS driver structure */
00154         tsDriver = &ft5336_ts_drv;
00155         I2cAddress = TS_I2C_ADDRESS;
00156         tsOrientation = TS_SWAP_XY;
00157
00158         /* Initialize the TS driver */
00159         tsDriver->Start(I2cAddress);
00160     }
00161     else
00162     {
00163         status = TS_DEVICE_NOT_FOUND;
00164     }
00165
00166     return status;
00167 }
00168
00169 /**
00170  * @brief DeInitializes the TouchScreen.
00171  * @retval TS state
00172 */
00173 uint8_t BSP_TS_DeInit(void)
00174 {
00175     /* Actually ts_driver does not provide a D
00176      eInit function */
00176     return TS_OK;
```

```
00177 }
00178 /**
00179  * @brief Configures and enables the touch
00180  * screen interrupts.
00181  * @retval TS_OK if all initializations are
00182  * OK. Other value if error.
00183 uint8_t BSP_TS_ITConfig(void)
00184 {
00185     GPIO_InitTypeDef gpio_init_structure;
00186
00187     /* Configure Interrupt mode for SD detection pin */
00188     gpio_init_structure.Pin = TS_INT_PIN;
00189     gpio_init_structure.Pull = GPIO_NOPULL;
00190     gpio_init_structure.Speed = GPIO_SPEED_FAST;
00191     gpio_init_structure.Mode = GPIO_MODE_IT_RISING;
00192     HAL_GPIO_Init(TS_INT_GPIO_PORT, &gpio_init_structure);
00193
00194     /* Enable and set Touch screen EXTI Interrupt to the lowest priority */
00195     HAL_NVIC_SetPriority((IRQn_Type)(TS_INT_EXTI_IRQn), 0x0F, 0x00);
00196     HAL_NVIC_EnableIRQ((IRQn_Type)(TS_INT_EXTI_IRQn));
00197
00198     /* Enable the TS ITs */
00199     tsDriver->EnableIT(I2cAddress);
00200
00201     return TS_OK;
00202 }
00203
00204 /**
```

```
00205     * @brief Gets the touch screen interrupt
00206     * @retval TS_OK if all initializations are
00207     * OK. Other value if error.
00208     */
00209 uint8_t BSP_TS_ITGetStatus(void)
00210 {
00211     /* Return the TS IT status */
00212     return (tsDriver->GetITStatus(I2cAddress))
00213 ;
00214 }
00215 /**
00216     * @brief Returns status and positions of
00217     * the touch screen.
00218     * @param TS_State: Pointer to touch screen
00219     * current state structure
00220     * @retval TS_OK if all initializations are
00221     * OK. Other value if error.
00222     */
00223 uint8_t BSP_TS_GetState(TS_StateTypeDef *TS_
00224 State)
00225 {
00226     static uint32_t _x[TS_MAX_NB_TOUCH] = {0,
00227     0};
00228     static uint32_t _y[TS_MAX_NB_TOUCH] = {0,
00229     0};
00230     uint8_t ts_status = TS_OK;
00231     uint16_t x[TS_MAX_NB_TOUCH];
00232     uint16_t y[TS_MAX_NB_TOUCH];
00233     uint16_t brute_x[TS_MAX_NB_TOUCH];
00234     uint16_t brute_y[TS_MAX_NB_TOUCH];
00235     uint16_t x_diff;
00236     uint16_t y_diff;
00237     uint32_t index;
00238 #if (TS_MULTI_TOUCH_SUPPORTED == 1)
00239     uint32_t weight = 0;
```

```
00233     uint32_t area = 0;
00234     uint32_t event = 0;
00235 #endif /* TS_MULTI_TOUCH_SUPPORTED == 1 */
00236
00237     /* Check and update the number of touches
active detected */
00238     TS_State->touchDetected = tsDriver->Detect
Touch(I2cAddress);
00239
00240     if(TS_State->touchDetected)
00241     {
00242         for(index=0; index < TS_State->touchDete
cted; index++)
00243         {
00244             /* Get each touch coordinates */
00245             tsDriver->GetXY(I2cAddress, &(brute_x[
index]), &(brute_y[index]));
00246
00247             if(tsOrientation == TS_SWAP_NONE)
00248             {
00249                 x[index] = brute_x[index];
00250                 y[index] = brute_y[index];
00251             }
00252
00253             if(tsOrientation & TS_SWAP_X)
00254             {
00255                 x[index] = 4096 - brute_x[index];
00256             }
00257
00258             if(tsOrientation & TS_SWAP_Y)
00259             {
00260                 y[index] = 4096 - brute_y[index];
00261             }
00262
00263             if(tsOrientation & TS_SWAP_XY)
00264             {
00265                 y[index] = brute_x[index];
```

```
00266         x[index] = brute_y[index];
00267     }
00268
00269     x_diff = x[index] > _x[index]? (x[inde
x] - _x[index]): (_x[index] - x[index]);
00270     y_diff = y[index] > _y[index]? (y[inde
x] - _y[index]): (_y[index] - y[index]);
00271
00272     if ((x_diff + y_diff) > 5)
00273     {
00274         _x[index] = x[index];
00275         _y[index] = y[index];
00276     }
00277
00278     if(I2cAddress == FT5336_I2C_SLAVE_ADDR
ESS)
00279     {
00280         TS_State->touchX[index] = x[index];
00281         TS_State->touchY[index] = y[index];
00282     }
00283     else
00284     {
00285         /* 2^12 = 4096 : indexes are express
ed on a dynamic of 4096 */
00286         TS_State->touchX[index] = (tsXBounda
ry * _x[index]) >> 12;
00287         TS_State->touchY[index] = (tsYBounda
ry * _y[index]) >> 12;
00288     }
00289
00290 #if (TS_MULTI_TOUCH_SUPPORTED == 1)
00291
00292     /* Get touch info related to the curre
nt touch */
00293     ft5336_TS_GetTouchInfo(I2cAddress, ind
ex, &weight, &area, &event);
00294
```

```
00295     /* Update TS_State structure */
00296     TS_State->touchWeight[index] = weight;
00297     TS_State->touchArea[index]   = area;
00298
00299     /* Remap touch event */
00300     switch(event)
00301     {
00302         case FT5336_TOUCH_EVT_FLAG_PRESS_DOWN :
00303             TS_State->touchEventId[index] = TOUCH_EVENT_PRESS_DOWN;
00304             break;
00305         case FT5336_TOUCH_EVT_FLAG_LIFT_UP :
00306             TS_State->touchEventId[index] = TOUCH_EVENT_LIFT_UP;
00307             break;
00308         case FT5336_TOUCH_EVT_FLAG_CONTACT :
00309             TS_State->touchEventId[index] = TOUCH_EVENT_CONTACT;
00310             break;
00311         case FT5336_TOUCH_EVT_FLAG_NO_EVENT :
00312             TS_State->touchEventId[index] = TOUCH_EVENT_NO_EVT;
00313             break;
00314         default :
00315             ts_status = TS_ERROR;
00316             break;
00317     } /* of switch(event) */
00318
00319 #endif /* TS_MULTI_TOUCH_SUPPORTED == 1 */
00320
00321     } /* of for(index=0; index < TS_State->touchDetected; index++) */
00322
00323 #if (TS_MULTI_TOUCH_SUPPORTED == 1)
00324     /* Get gesture Id */
```

```
00325     ts_status = BSP_TS_Get_GestureId(TS_Status);
00326 #endif /* TS_MULTI_TOUCH_SUPPORTED == 1 */
00327
00328 } /* end of if(TS_State->touchDetected != 0) */
00329
00330 return (ts_status);
00331 }
00332
00333 #if (TS_MULTI_TOUCH_SUPPORTED == 1)
00334 /**
00335 * @brief Update gesture Id following a touch detected.
00336 * @param TS_State: Pointer to touch screen current state structure
00337 * @retval TS_OK if all initializations are OK. Other value if error.
00338 */
00339 uint8_t BSP_TS_Get_GestureId(TS_StateTypeDef
 *TS_State)
00340 {
00341     uint32_t gestureId = 0;
00342     uint8_t ts_status = TS_OK;
00343
00344     /* Get gesture Id */
00345     ft5336_TS_GetGestureID(I2cAddress, &gestureId);
00346
00347     /* Remap gesture Id to a TS_GestureIdTypeDef value */
00348     switch(gestureId)
00349     {
00350         case FT5336_GEST_ID_NO_GESTURE :
00351             TS_State->gestureId = GEST_ID_NO_GESTURE;
00352             break;
```

```
00353     case FT5336_GEST_ID_MOVE_UP :
00354         TS_State->gestureId = GEST_ID_MOVE_UP;
00355         break;
00356     case FT5336_GEST_ID_MOVE_RIGHT :
00357         TS_State->gestureId = GEST_ID_MOVE_RIG
00358         HT;
00359         break;
00360     case FT5336_GEST_ID_MOVE_DOWN :
00361         TS_State->gestureId = GEST_ID_MOVE_DOWN
00362         ;
00363         break;
00364     case FT5336_GEST_ID_MOVE_LEFT :
00365         TS_State->gestureId = GEST_ID_MOVE_LEFT
00366         ;
00367         break;
00368     case FT5336_GEST_ID_ZOOM_IN :
00369         TS_State->gestureId = GEST_ID_ZOOM_IN;
00370         break;
00371     case FT5336_GEST_ID_ZOOM_OUT :
00372         TS_State->gestureId = GEST_ID_ZOOM_OUT
00373         ;
00374         break;
00375     default :
00376         ts_status = TS_ERROR;
00377         break;
00378     } /* of switch(gestureId) */
00379
00380     return(ts_status);
00381 }
00382 #endif /* TS_MULTI_TOUCH_SUPPORTED == 1 */
00383 /**
00384 * @brief Clears all touch screen interrupt.
00385 */
00386 void BSP_TS_ITClear(void)
00387 {
```

```
00385     /* Clear TS IT pending bits */
00386     tsDriver->ClearIT(I2cAddress);
00387 }
00388
00389
00390 /**
00391 * @defgroup STM32756G_DISCOVERY_TS_Private
00392 _Functions TS Private Functions
00393 */
00394
00395 /**
00396 * @brief Function used to reset all touch
00397 data before a new acquisition
00398 * of touch information.
00399 * @param TS_State: Pointer to touch screen
00400 current state structure
00401 * @retval TS_OK if OK, TE_ERROR if problem
00402 found.
00403 */
00404 uint8_t BSP_TS_ResetTouchData(TS_StateTypeDef
00405 *TS_State)
00406 {
00407     uint8_t ts_status = TS_ERROR;
00408     uint32_t index;
00409
00410     if (TS_State != (TS_StateTypeDef *)NULL)
00411     {
00412         TS_State->gestureId = GEST_ID_NO_GESTURE
00413 ;
00414         TS_State->touchDetected = 0;
00415
00416         for(index = 0; index < TS_MAX_NB_TOUCH;
00417 index++)
00418         {
00419             TS_State->touchX[index] = 0;
00420             TS_State->touchY[index] = 0;
```

```
00415     TS_State->touchArea[index] = 0;
00416     TS_State->touchEventId[index] = TOUCH_
EVENT_NO_EVT;
00417     TS_State->touchWeight[index] = 0;
00418 }
00419
00420     ts_status = TS_OK;
00421
00422 } /* of if (TS_State != (TS_StateTypeDef *
)NULL) */
00423
00424     return (ts_status);
00425 }
00426
00427 /**
00428 * @}
00429 */
00430
00431 /**
00432 * @}
00433 */
00434
00435 /**
00436 * @}
00437 */
00438
00439 /**
00440 * @}
00441 */
00442
00443 /**
00444 * @}
00445 */
00446
00447 ***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/
```

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules		

STM32746G_DISCOVERY_CAMERA

[STM32746G_DISCOVERY](#)

Modules

[STM32746G_DISCOVERY_CAMERA Private Types Definitions](#)

[STM32746G_DISCOVERY_CAMERA Private Defines](#)

[STM32746G_DISCOVERY_CAMERA Private Macros](#)

[STM32746G_DISCOVERY_CAMERA Private Variables](#)

[STM32746G_DISCOVERY_CAMERA Private Function Prototypes](#)

[STM32746G_DISCOVERY_CAMERA Exported Functions](#)

[STM32746G_DISCOVERY_CAMERA Exported Types](#)

[STM32746G_DISCOVERY_CAMERA Exported Constants](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules		

STM32746G_DISCOVERY_EEPROM

This file includes the I2C EEPROM driver of STM32746G-Discovery board. [More...](#)

Modules

[**STM32746G_DISCOVERY_EEPROM Private Types**](#)
[**STM32746G_DISCOVERY_EEPROM Private Defines**](#)
[**STM32746G_DISCOVERY_EEPROM Private Macros**](#)
[**STM32746G_DISCOVERY_EEPROM Private Variables**](#)
[**STM32746G_DISCOVERY_EEPROM Private Function Prototypes**](#)
[**STM32746G_DISCOVERY_EEPROM Exported Functions**](#)
[**STM32746G_DISCOVERY_EEPROM Exported Types**](#)
[**STM32746G_DISCOVERY_EEPROM Exported Constants**](#)
[**STM32746G_DISCOVERY_EEPROM Exported Macros**](#)

Detailed Description

This file includes the I2C EEPROM driver of STM32746G-Discovery board.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules		

STM32746G_DISCOVERY_LCD

[STM32746G_DISCOVERY](#)

Modules

[**STM32746G_DISCOVERY_LCD Private Types Definitions**](#)

[**STM32746G_DISCOVERY_LCD Private Defines**](#)

[**STM32746G_DISCOVERY_LCD Private Macros**](#)

[**STM32746G_DISCOVERY_LCD Private Variables**](#)

[**STM32746G_DISCOVERY_LCD Private Function Prototypes**](#)

[**STM32746G_DISCOVERY_LCD Exported Functions**](#)

[**STM32746G_DISCOVERY_LCD Exported Types**](#)

[**STM32746G_DISCOVERY_LCD Exported Constants**](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [**doxygen**](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories		Data Structures Typedefs Enumerations	

STM32746G_DISCOVERY_LCD Exported Types

[STM32746G_DISCOVERY_LCD](#)

Data Structures

```
struct LCD_DrawPropTypeDef  
struct Point
```

TypeDefs

```
typedef struct Point * pPoint
```

Enumerations

enum **Text_AlignModeTypdef** { **CENTER_MODE** = 0x01,
RIGHT_MODE = 0x02, **LEFT_MODE** = 0x03 }

Line mode structures definition. More...

TypeDef Documentation

typedef struct Point * pPoint

Enumeration Type Documentation

enum Text_AlignModeTypdef

Line mode structures definition.

Enumerator:

CENTER_MODE

RIGHT_MODE

LEFT_MODE

Definition at line [89](#) of file [stm32746g_discovery_lcd.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_lcd.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_lcd.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains the common d
efines and functions prototypes for
00008 *           the stm32746g_discovery_lcd.c d
river.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and bin
ary forms, with or without modification,
00015 * are permitted provided that the followin
g conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038      */
00039 /* Define to prevent recursive inclusion ---
00040 -----
00041 #ifndef __STM32746G_DISCOVERY_LCD_H
00042 #define __STM32746G_DISCOVERY_LCD_H
00043
00044 #ifdef __cplusplus
00045   extern "C" {
00046 #endif
00047
00048 /* Includes -----
00049 -----
00050 /* Include LCD component Driver */
00051 /* LCD RK043FN48H-CT672B 4,3" 480x272 pixels
00052 */
00053 #include "../Components/rk043fn48h/rk043fn48
00054 h.h"
00055
00056 /* Include SDRAM Driver */
00057 #include "stm32746g_discovery_sdram.h"
00058
00059 /**
00060 */
00061
00062 /**
00063 */
00064
00065
00066 /**
00067 */
```

```
00067      * @{
00068      */
00069
00070 /** @defgroup STM32746G_DISCOVERY_LCD_Export
00071 ed_Types STM32746G_DISCOVERY_LCD Exported Types
00072      * @{
00073
00074     typedef struct
00075     {
00076         uint32_t TextColor;
00077         uint32_t BackColor;
00078         sFONT      *pFont;
00079     }LCD_DrawPropTypeDef;
00080
00081     typedef struct
00082     {
00083         int16_t X;
00084         int16_t Y;
00085     }Point, * pPoint;
00086
00087     * @brief Line mode structures definition
00088
00089     typedef enum
00090     {
00091         CENTER_MODE          = 0x01,      /* Cent
00092 er mode */
00093         RIGHT_MODE           = 0x02,      /* Righ
00094 t mode */
00095         LEFT_MODE            = 0x03,      /* Left
00096 mode */
00097     }Text_AlignModeTypdef;
00098
00099
00100    * @}
00101
00102
```

```
00099
00100 /** @defgroup STM32746G_DISCOVERY_LCD_Export
00101   ed_Constants STM32746G_DISCOVERY_LCD Exported Cons
00102   tants
00103   * @{
00104   */
00105 #define MAX_LAYER_NUMBER          ((uint32_t)2)
00106
00107 #define LCD_LayerCfgTypeDef      LTDC_LayerCfg
00108 TypeDef
00109
00110 #define LTDC_ACTIVE_LAYER        ((uint3
00111 2_t)1) /* Layer 1 */
00112 /**
00113   * @brief LCD status structure definition
00114
00115 */
00116 #define LCD_OK                  ((uint8_t)0x0
00117 0)
00118 #define LCD_ERROR                ((uint8_t)0x0
1)
00119 #define LCD_TIMEOUT              ((uint8_t)0x0
2)
00120 /**
00121   * @brief LCD FB_StartAddress
00122 */
00123 #define LCD_FB_START_ADDRESS     ((uint32_
00124 t)0xC0000000)
00125
00126 /**
00127   * @brief LCD color
00128 */
00129 #define LCD_COLOR_BLUE           ((uint32_t)0
x000000FF)
00130 #define LCD_COLOR_GREEN          ((uint32_t)0
xFF00FF00)
```

00125 #define LCD_COLOR_RED	((uint32_t)0
xFFFF0000)	
00126 #define LCD_COLOR_CYAN	((uint32_t)0
xFF00FFFF)	
00127 #define LCD_COLOR_MAGENTA	((uint32_t)0
xFFFFF00FF)	
00128 #define LCD_COLOR_YELLOW	((uint32_t)0
xFFFFFFF00)	
00129 #define LCD_COLOR_LIGHTBLUE	((uint32_t)0
xFF8080FF)	
00130 #define LCD_COLOR_LIGHTGREEN	((uint32_t)0
xFF80FF80)	
00131 #define LCD_COLOR_LIGHTRED	((uint32_t)0
xFFFFF8080)	
00132 #define LCD_COLOR_LIGHTCYAN	((uint32_t)0
xFF80FFFF)	
00133 #define LCD_COLOR_LIGHTMAGENTA	((uint32_t)0
xFFFFF80FF)	
00134 #define LCD_COLOR_LIGHTYELLOW	((uint32_t)0
xFFFFFFF80)	
00135 #define LCD_COLOR_DARKBLUE	((uint32_t)0
xFF000080)	
00136 #define LCD_COLOR_DARKGREEN	((uint32_t)0
xFF008000)	
00137 #define LCD_COLOR_DARKRED	((uint32_t)0
xFF800000)	
00138 #define LCD_COLOR_DARKCYAN	((uint32_t)0
xFF008080)	
00139 #define LCD_COLOR_DARKMAGENTA	((uint32_t)0
xFF800080)	
00140 #define LCD_COLOR_DARKYELLOW	((uint32_t)0
xFF808000)	
00141 #define LCD_COLOR_WHITE	((uint32_t)0
xFFFFFFF00)	
00142 #define LCD_COLOR_LIGHTGRAY	((uint32_t)0
xFFD3D3D3)	
00143 #define LCD_COLOR_GRAY	((uint32_t)0

```

00140 #define LCD_COLOR_DARKGRAY ((uint32_t)0
00141 xFF808080)
00142 #define LCD_COLOR_BLACK ((uint32_t)0
00143 xFF000000)
00144 #define LCD_COLOR_BROWN ((uint32_t)0
00145 xFFA52A2A)
00146 #define LCD_COLOR_ORANGE ((uint32_t)0
00147 xFFFFA500)
00148 #define LCD_COLOR_TRANSPARENT ((uint32_t)0
00149 xFF000000)

00150 /**
00151 * @brief LCD default font
00152 */
00153 #define LCD_DEFAULT_FONT Font24
00154
00155 /**
00156 * @brief LCD Reload Types
00157 */
00158 #define LCD_RELOAD_IMMEDIATE ((uint32_t)LTDC_SRCR_IMR)
00159 #define LCD_RELOAD_VERTICAL_BLANKING ((uint32_t)LTDC_SRCR_VBR)
00160
00161
00162 /**
00163 * @brief LCD special pins
00164 */
00165 /* Display enable pin */
00166 #define LCD_DISP_PIN GPIO
00167 _PIN_12
00168 #define LCD_DISP_GPIO_PORT GPIOI
00169 #define LCD_DISP_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()
00170 #define LCD_DISP_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DISABLE()

```

```
L_RCC_GPIOI_CLK_DISABLE()
00170
00171 /* Backlight control pin */
00172 #define LCD_BL_CTRL_PIN          GPIO
0_PIN_3
00173 #define LCD_BL_CTRL_GPIO_PORT    GPIO
OK
00174 #define LCD_BL_CTRL_GPIO_CLK_ENABLE() __H
AL_RCC_GPIOK_CLK_ENABLE()
00175 #define LCD_BL_CTRL_GPIO_CLK_DISABLE() __H
AL_RCC_GPIOK_CLK_DISABLE()

00176
00177 /**
00178 * @}
00179 */
00180
00181 /** @addtogroup STM32746G_DISCOVERY_LCD_Expo
rted_Functions
00182 * @{
00183 */
00184 uint8_t    BSP_LCD_Init(void);
00185 uint8_t    BSP_LCD_DeInit(void);
00186 uint32_t   BSP_LCD_GetXSize(void);
00187 uint32_t   BSP_LCD_GetYSize(void);
00188 void       BSP_LCD_SetXSize(uint32_t imageWidthPixels);
00189 void       BSP_LCD_SetYSize(uint32_t imageHeightPixels);
00190
00191 /* Functions using the LTDC controller */
00192 void       BSP_LCD_LayerDefaultInit(uint16_t LayerIndex, uint32_t FrameBuffer);
00193 void       BSP_LCD_LayerRgb565Init(uint16_t LayerIndex, uint32_t FB_Address);
00194 void       BSP_LCD_SetTransparency(uint32_t LayerIndex, uint8_t Transparency);
00195 void       BSP_LCD_SetTransparency_NoReload(ui
```

```
00195     nt32_t LayerIndex, uint8_t Transparency);
00196 void      BSP_LCD_SetLayerAddress(uint32_t La
yerIndex, uint32_t Address);
00197 void      BSP_LCD_SetLayerAddress_NoReload(ui
nt32_t LayerIndex, uint32_t Address);
00198 void      BSP_LCD_SetColorKeying(uint32_t Lay
erIndex, uint32_t RGBValue);
00199 void      BSP_LCD_SetColorKeying_NoReload(ui
nt32_t LayerIndex, uint32_t RGBValue);
00200 void      BSP_LCD_ResetColorKeying(uint32_t L
ayerIndex);
00201 void      BSP_LCD_ResetColorKeying_NoReload(ui
nt32_t LayerIndex);
00202 void      BSP_LCD_SetLayerWindow(uint16_t Lay
erIndex, uint16_t Xpos, uint16_t Ypos, uint16_t Wi
dth, uint16_t Height);
00203 void      BSP_LCD_SetLayerWindow_NoReload(ui
nt16_t LayerIndex, uint16_t Xpos, uint16_t Ypos, ui
nt16_t Width, uint16_t Height);
00204 void      BSP_LCD_SelectLayer(uint32_t LayerI
ndex);
00205 void      BSP_LCD_SetLayerVisible(uint32_t La
yerIndex, FunctionalState State);
00206 void      BSP_LCD_SetLayerVisible_NoReload(ui
nt32_t LayerIndex, FunctionalState State);
00207 void      BSP_LCD_Reload(uint32_t ReloadType)
;
00208
00209 void      BSP_LCD_SetTextColor(uint32_t Color
);
00210 uint32_t  BSP_LCD_GetTextColor(void);
00211 void      BSP_LCD_SetBackColor(uint32_t Color
);
00212 uint32_t  BSP_LCD_GetBackColor(void);
00213 void      BSP_LCD_SetFont(sFONT *fonts);
00214 sFONT     *BSP_LCD_GetFont(void);
00215
```

```
00216 uint32_t BSP_LCD_ReadPixel(uint16_t Xpos, ui
nt16_t Ypos);
00217 void      BSP_LCD_DrawPixel(uint16_t Xpos, ui
nt16_t Ypos, uint32_t pixel);
00218 void      BSP_LCD_Clear(uint32_t Color);
00219 void      BSP_LCD_ClearStringLine(uint32_t Li
ne);
00220 void      BSP_LCD_DisplayStringAtLine(uint16_
t Line, uint8_t *ptr);
00221 void      BSP_LCD_DisplayStringAt(uint16_t Xp
os, uint16_t Ypos, uint8_t *Text, Text_AlignModeTy
pdef Mode);
00222 void      BSP_LCD_DisplayChar(uint16_t Xpos,
uint16_t Ypos, uint8_t Ascii);
00223
00224 void      BSP_LCD_DrawHLine(uint16_t Xpos, ui
nt16_t Ypos, uint16_t Length);
00225 void      BSP_LCD_DrawVLine(uint16_t Xpos, ui
nt16_t Ypos, uint16_t Length);
00226 void      BSP_LCD_DrawLine(uint16_t x1, uint1
6_t y1, uint16_t x2, uint16_t y2);
00227 void      BSP_LCD_DrawRect(uint16_t Xpos, uin
t16_t Ypos, uint16_t Width, uint16_t Height);
00228 void      BSP_LCD_DrawCircle(uint16_t Xpos, u
int16_t Ypos, uint16_t Radius);
00229 void      BSP_LCD_DrawPolygon(pPoint Points,
uint16_t PointCount);
00230 void      BSP_LCD_DrawEllipse(int Xpos, int Y
pos, int XRadius, int YRadius);
00231 void      BSP_LCD_DrawBitmap(uint32_t Xpos, u
int32_t Ypos, uint8_t *pbmp);
00232
00233 void      BSP_LCD_FillRect(uint16_t Xpos, uin
t16_t Ypos, uint16_t Width, uint16_t Height);
00234 void      BSP_LCD_FillCircle(uint16_t Xpos, u
int16_t Ypos, uint16_t Radius);
00235 void      BSP_LCD_FillPolygon(pPoint Points,
```

```
00235     uint16_t PointCount);  
00236 void      BSP_LCD_FillEllipse(int Xpos, int Y  
pos, int XRadius, int YRadius);  
00237  
00238 void      BSP_LCD_DisplayOff(void);  
00239 void      BSP_LCD_DisplayOn(void);  
00240  
00241 /* These functions can be modified in case t  
he current settings  
00242     need to be changed for specific applicati  
on needs */  
00243 void      BSP_LCD_MspInit(LTDC_HandleTypeDef  
*hltdc, void *Params);  
00244 void      BSP_LCD_MspDeInit(LTDC_HandleTypeDef  
*hltdc, void *Params);  
00245 void      BSP_LCD_ClockConfig(LTDC_HandleTypeDef  
*hltdc, void *Params);  
00246  
00247 /**  
00248     * @}  
00249     */  
00250  
00251 /**  
00252     * @}  
00253     */  
00254  
00255 /**  
00256     * @}  
00257     */  
00258  
00259 /**  
00260     * @}  
00261     */  
00262  
00263 #ifdef __cplusplus  
00264 }  
00265 #endif
```

```
00266
00267 #endif /* __STM32746G_DISCOVERY_LCD_H */
00268
00269 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_lcd.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_lcd.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file includes the driver f
or Liquid Crystal Display (LCD) module
00008 *           mounted on STM32746G-Discovery
board.
00009 @verbatim
00010 1. How To use this driver:
00011 -----
00012      - This driver is used to drive directly
an LCD TFT using the LTDC controller.
00013      - This driver uses timing and setting f
or RK043FN48H LCD.
00014
00015 2. Driver description:
00016 -----
00017      + Initialization steps:
```

```
00018      o Initialize the LCD using the BSP_LC  
D_Init() function.  
00019      o Apply the Layer configuration using  
the BSP_LCD_LayerDefaultInit() function.  
00020      o Select the LCD layer to be used usi  
ng the BSP_LCD_SelectLayer() function.  
00021      o Enable the LCD display using the BS  
P_LCD_DisplayOn() function.  
00022  
00023      + Options  
00024          o Configure and enable the color keyi  
ng functionality using the  
00025              BSP_LCD_SetColorKeying() function.  
00026          o Modify in the fly the transparency  
and/or the frame buffer address  
00027              using the following functions:  
00028                  - BSP_LCD_SetTransparency()  
00029                  - BSP_LCD_SetLayerAddress()  
00030  
00031      + Display on LCD  
00032          o Clear the hole LCD using BSP_LCD_Cl  
ear() function or only one specified string  
00033          line using the BSP_LCD_ClearStringL  
ine() function.  
00034          o Display a character on the specifie  
d line and column using the BSP_LCD_DisplayChar()  
00035          function or a complete string line  
using the BSP_LCD_DisplayStringAtLine() function.  
00036          o Display a string line on the specif  
ied position (x,y in pixel) and align mode  
00037          using the BSP_LCD_DisplayStringAtLi  
ne() function.  
00038          o Draw and fill a basic shapes (dot,  
line, rectangle, circle, ellipse, .. bitmap)  
00039          on LCD using the available set of f  
unctions.  
00040  @endverbatim
```

00041 *****

00042 * @attention
00043 *
00044 * <h2><center>© COPYRIGHT(c) 2016 STMicroelectronics</center></h2>
00045 *
00046 * Redistribution and use in source and binary forms, with or without modification,
00047 * are permitted provided that the following conditions are met:
00048 * 1. Redistributions of source code must retain the above copyright notice,
00049 * this list of conditions and the following disclaimer.
00050 * 2. Redistributions in binary form must reproduce the above copyright notice,
00051 * this list of conditions and the following disclaimer in the documentation
00052 * and/or other materials provided with the distribution.
00053 * 3. Neither the name of STMicroelectronics nor the names of its contributors
00054 * may be used to endorse or promote products derived from this software
00055 * without specific prior written permission.
00056 *
00057 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
00058 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
00059 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
00060 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
00061 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP

ECIAL, EXEMPLARY, OR CONSEQUENTIAL
00062 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR
00063 * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER
00064 * CAUSED AND ON ANY THEORY OF LIABILITY, W
HETHER IN CONTRACT, STRICT LIABILITY,
00065 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI
SE) ARISING IN ANY WAY OUT OF THE USE
00066 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

00067 *

00068 *****

00069 */

00070

00071 /* Includes -----
-----*/

00072 #include "stm32746g_discovery_lcd.h"
00073 #include "../../Utilities/Fonts/fonts.h"
00074 #include "../../Utilities/Fonts/font24.c"
00075 #include "../../Utilities/Fonts/font20.c"
00076 #include "../../Utilities/Fonts/font16.c"
00077 #include "../../Utilities/Fonts/font12.c"
00078 #include "../../Utilities/Fonts/font8.c"
00079

00080 /** @addtogroup BSP
00081 * @{
00082 */

00083

00084 /** @addtogroup STM32746G_DISCOVERY
00085 * @{
00086 */

00087

00088 /** @addtogroup STM32746G_DISCOVERY_LCD
00089 * @{
00090 */

```
00091
00092 /** @defgroup STM32746G_DISCOVERY_LCD_Private_TypesDefinitions STM32746G_DISCOVERY_LCD Private Types Definitions
00093 * @{
00094 */
00095 /**
00096 * @}
00097 */
00098
00099 /** @defgroup STM32746G_DISCOVERY_LCD_PrivateDefines STM32746G_DISCOVERY_LCD Private Defines
00100 * @{
00101 */
00102 #define POLY_X(Z) ((int32_t)((P  
oints + Z)->X))
00103 #define POLY_Y(Z) ((int32_t)((P  
oints + Z)->Y))
00104 /**
00105 * @}
00106 */
00107
00108 /** @defgroup STM32746G_DISCOVERY_LCD_PrivateMacros STM32746G_DISCOVERY_LCD Private Macros
00109 * @{
00110 */
00111 #define ABS(X) ((X) > 0 ? (X) : -(X))
00112 /**
00113 * @}
00114 */
00115
00116 /** @defgroup STM32746G_DISCOVERY_LCD_PrivateVariables STM32746G_DISCOVERY_LCD Private Variables
00117 * @{
00118 */
00119 LTDC_HandleTypeDef hLtdcHandler;
```

```
00120 static DMA2D_HandleTypeDef hDma2dHandler;
00121
00122 /* Default LCD configuration with LCD Layer
1 */
00123 static uint32_t ActiveLayer = 0;
00124 static LCD_DrawPropTypeDef DrawProp[MAX_LAYE
R_NUMBER];
00125 /**
00126 * @}
00127 */
00128
00129 /** @defgroup STM32746G_DISCOVERY_LCD_Privat
e_FunctionPrototypes STM32746G_DISCOVERY_LCD Priva
te Function Prototypes
00130 * @{
00131 */
00132 static void DrawChar(uint16_t Xpos, uint16_t
Ypos, const uint8_t *c);
00133 static void FillTriangle(uint16_t x1, uint16
_t x2, uint16_t x3, uint16_t y1, uint16_t y2, uint
16_t y3);
00134 static void LL_FillBuffer(uint32_t LayerInde
x, void *pDst, uint32_t xSize, uint32_t ySize, ui
n t32_t OffLine, uint32_t ColorIndex);
00135 static void LL_ConvertLineToARGB8888(void *
pSrc, void *pDst, uint32_t xSize, uint32_t ColorMo
de);
00136 /**
00137 * @}
00138 */
00139
00140 /** @defgroup STM32746G_DISCOVERY_LCD_Export
ed_Functions STM32746G_DISCOVERY_LCD Exported Func
tions
00141 * @{
00142 */
00143
```

```
00144 /**
00145  * @brief  Initializes the LCD.
00146  * @retval LCD state
00147 */
00148 uint8_t BSP_LCD_Init(void)
00149 {
00150     /* Select the used LCD */
00151
00152     /* The RK043FN48H LCD 480x272 is selected
00153     */
00154     hLtdcHandler.Init.HorizontalSync = (RK043FN48H_HSYNC - 1);
00155     hLtdcHandler.Init.VerticalSync = (RK043FN48H_VSYNC - 1);
00156     hLtdcHandler.Init.AccumulatedHBP = (RK043FN48H_HSYNC + RK043FN48H_HBP - 1);
00157     hLtdcHandler.Init.AccumulatedVBP = (RK043FN48H_VSYNC + RK043FN48H_VBP - 1);
00158     hLtdcHandler.Init.AccumulatedActiveH = (RK043FN48H_HEIGHT + RK043FN48H_VSYNC + RK043FN48H_VBP - 1);
00159     hLtdcHandler.Init.AccumulatedActiveW = (RK043FN48H_WIDTH + RK043FN48H_HSYNC + RK043FN48H_HBP - 1);
00160     hLtdcHandler.Init.TotalHeigh = (RK043FN48H_HEIGHT + RK043FN48H_VSYNC + RK043FN48H_VBP + RK043FN48H_VFP - 1);
00161     hLtdcHandler.Init.TotalWidth = (RK043FN48H_WIDTH + RK043FN48H_HSYNC + RK043FN48H_HBP + RK043FN48H_HFP - 1);
00162
00163     /* LCD clock configuration */
00164     BSP_LCD_ClockConfig(&hLtdcHandler, NULL);
00165
00166     /* Initialize the LCD pixel width and pixel height */
```

```
00167     hLtdcHandler.LayerCfg->ImageWidth = RK043
FN48H_WIDTH;
00168     hLtdcHandler.LayerCfg->ImageHeight = RK043
FN48H_HEIGHT;
00169
00170     /* Background value */
00171     hLtdcHandler.Init.Backcolor.Blue = 0;
00172     hLtdcHandler.Init.Backcolor.Green = 0;
00173     hLtdcHandler.Init.Backcolor.Red = 0;
00174
00175     /* Polarity */
00176     hLtdcHandler.Init.HSPolarity = LTDC_HSPOLA
RITY_AL;
00177     hLtdcHandler.Init.VSPolarity = LTDC_VSPOLA
RITY_AL;
00178     hLtdcHandler.Init.DEPolarity = LTDC_DEPOLA
RITY_AL;
00179     hLtdcHandler.Init.PCPolarity = LTDC_PCPOLA
RITY_IPC;
00180     hLtdcHandler.Instance = LTDC;
00181
00182     if(HAL_LTDC_GetState(&hLtdcHandler) == HAL
_LTDC_STATE_RESET)
00183     {
00184         /* Initialize the LCD Msp: this __weak f
unction can be rewritten by the application */
00185         BSP_LCD_MspInit(&hLtdcHandler, NULL);
00186     }
00187     HAL_LTDC_Init(&hLtdcHandler);
00188
00189     /* Assert display enable LCD_DISP pin */
00190     HAL_GPIO_WritePin(LCD_DISP_GPIO_PORT, LCD_
DISP_PIN, GPIO_PIN_SET);
00191
00192     /* Assert backlight LCD_BL_CTRL pin */
00193     HAL_GPIO_WritePin(LCD_BL_CTRL_GPIO_PORT, L
CD_BL_CTRL_PIN, GPIO_PIN_SET);
```

```
00194
00195 #if !defined(DATA_IN_ExtSDRAM)
00196     /* Initialize the SDRAM */
00197     BSP_SDRAM_Init();
00198 #endif
00199
00200     /* Initialize the font */
00201     BSP_LCD_SetFont(&LCD_DEFAULT_FONT);
00202
00203     return LCD_OK;
00204 }
00205
00206 /**
00207     * @brief DeInitializes the LCD.
00208     * @retval LCD state
00209 */
00210 uint8_t BSP_LCD_DeInit(void)
00211 {
00212     /* Initialize the hLtdcHandler Instance parameter */
00213     hLtdcHandler.Instance = LTDC;
00214
00215     /* Disable LTDC block */
00216     __HAL_LTDC_DISABLE(&hLtdcHandler);
00217
00218     /* DeInit the LTDC */
00219     HAL_LTDC_DeInit(&hLtdcHandler);
00220
00221     /* DeInit the LTDC MSP : this __weak function can be rewritten by the application */
00222     BSP_LCD_MspDeInit(&hLtdcHandler, NULL);
00223
00224     return LCD_OK;
00225 }
00226
00227 /**
00228     * @brief Gets the LCD X size.
```

```
00229     * @retval Used LCD X size
00230     */
00231 uint32_t BSP_LCD_GetXSize(void)
00232 {
00233     return hLtdcHandler.LayerCfg[ActiveLayer].ImageWidth;
00234 }
00235
00236 /**
00237     * @brief Gets the LCD Y size.
00238     * @retval Used LCD Y size
00239 */
00240 uint32_t BSP_LCD_GetYSize(void)
00241 {
00242     return hLtdcHandler.LayerCfg[ActiveLayer].ImageHeight;
00243 }
00244
00245 /**
00246     * @brief Set the LCD X size.
00247     * @param imageWidthPixels : image width in pixels unit
00248     * @retval None
00249 */
00250 void BSP_LCD_SetXSize(uint32_t imageWidthPixels)
00251 {
00252     hLtdcHandler.LayerCfg[ActiveLayer].ImageWidth = imageWidthPixels;
00253 }
00254
00255 /**
00256     * @brief Set the LCD Y size.
00257     * @param imageHeightPixels : image height in lines unit
00258     * @retval None
00259 */
```

```
00260 void BSP_LCD_SetYSize(uint32_t imageHeightPi
xels)
00261 {
00262     hLtdcHandler.LayerCfg[ActiveLayer].ImageHe
ight = imageHeightPixels;
00263 }
00264
00265 /**
00266     * @brief Initializes the LCD layer in ARG
B8888 format (32 bits per pixel).
00267     * @param LayerIndex: Layer foreground or
background
00268     * @param FB_Address: Layer frame buffer
00269     * @retval None
00270 */
00271 void BSP_LCD_LayerDefaultInit(uint16_t Layer
Index, uint32_t FB_Address)
00272 {
00273     LCD_LayerCfgTypeDef    layer_cfg;
00274
00275     /* Layer Init */
00276     layer_cfg.WindowX0 = 0;
00277     layer_cfg.WindowX1 = BSP_LCD_GetXSize();
00278     layer_cfg.WindowY0 = 0;
00279     layer_cfg.WindowY1 = BSP_LCD_GetYSize();
00280     layer_cfg.PixelFormat = LTDC_PIXEL_FORMAT_
ARGB8888;
00281     layer_cfg.FBStartAdress = FB_Address;
00282     layer_cfg.Alpha = 255;
00283     layer_cfg.Alpha0 = 0;
00284     layer_cfg.Backcolor.Blue = 0;
00285     layer_cfg.Backcolor.Green = 0;
00286     layer_cfg.Backcolor.Red = 0;
00287     layer_cfg.BlendingFactor1 = LTDC_BLENDING_
FACTOR1_PAxCA;
00288     layer_cfg.BlendingFactor2 = LTDC_BLENDING_
FACTOR2_PAxCA;
```

```
00289     layer_cfg.ImageWidth = BSP_LCD_GetXSize();
00290     layer_cfg.ImageHeight = BSP_LCD_GetYSize()
00291 ;
00291
00292     HAL_LTDC_ConfigLayer(&hLtdcHandler, &layer
00293 _cfg, LayerIndex);
00293
00294     DrawProp[LayerIndex].BackColor = LCD_COLOR
00295 _WHITE;
00295     DrawProp[LayerIndex].pFont      = &Font24;
00296     DrawProp[LayerIndex].TextColor = LCD_COLOR
00297 _BLACK;
00297 }
00298
00299 /**
00300 * @brief Initializes the LCD layer in RGB
00301 565 format (16 bits per pixel).
00302 * @param LayerIndex: Layer foreground or
00303 background
00304 * @param FB_Address: Layer frame buffer
00305 * @retval None
00306 */
00305 void BSP_LCD_LayerRgb565Init(uint16_t LayerI
00306 ndex, uint32_t FB_Address)
00306 {
00307     LCD_LayerCfgTypeDef    layer_cfg;
00308
00309     /* Layer Init */
00310     layer_cfg.WindowX0 = 0;
00311     layer_cfg.WindowX1 = BSP_LCD_GetXSize();
00312     layer_cfg.WindowY0 = 0;
00313     layer_cfg.WindowY1 = BSP_LCD_GetYSize();
00314     layer_cfg.PixelFormat = LTDC_PIXEL_FORMAT_
00314 RGB565;
00315     layer_cfg.FBStartAdress = FB_Address;
00316     layer_cfg.Alpha = 255;
00317     layer_cfg.Alpha0 = 0;
```

```
00318     layer_cfg.Backcolor.Blue = 0;
00319     layer_cfg.Backcolor.Green = 0;
00320     layer_cfg.Backcolor.Red = 0;
00321     layer_cfg.BlendingFactor1 = LTDC_BLENDING_
FACTOR1_PAxCA;
00322     layer_cfg.BlendingFactor2 = LTDC_BLENDING_
FACTOR2_PAxCA;
00323     layer_cfg.ImageWidth = BSP_LCD_GetXSize();
00324     layer_cfg.ImageHeight = BSP_LCD_GetYSize()
;
00325
00326     HAL_LTDC_ConfigLayer(&hLtdcHandler, &layer
_cfg, LayerIndex);
00327
00328     DrawProp[LayerIndex].BackColor = LCD_COLOR
_WHITE;
00329     DrawProp[LayerIndex].pFont      = &Font24;
00330     DrawProp[LayerIndex].TextColor = LCD_COLOR
_BLACK;
00331 }
00332
00333 /**
00334     * @brief Selects the LCD Layer.
00335     * @param LayerIndex: Layer foreground or
background
00336     * @retval None
00337 */
00338 void BSP_LCD_SelectLayer(uint32_t LayerIndex
)
00339 {
00340     ActiveLayer = LayerIndex;
00341 }
00342
00343 /**
00344     * @brief Sets an LCD Layer visible
00345     * @param LayerIndex: Visible Layer
00346     * @param State: New state of the specific
```

```
d layer
00347     *           This parameter can be one of th
e following values:
00348     *           @arg  ENABLE
00349     *           @arg  DISABLE
00350     * @retval None
00351   */
00352 void BSP_LCD_SetLayerVisible(uint32_t LayerI
ndex, FunctionalState State)
00353 {
00354     if(State == ENABLE)
00355     {
00356         __HAL_LTDC_LAYER_ENABLE(&hLtdcHandler, L
ayerIndex);
00357     }
00358     else
00359     {
00360         __HAL_LTDC_LAYER_DISABLE(&hLtdcHandler,
LayerIndex);
00361     }
00362     __HAL_LTDC_RELOAD_CONFIG(&hLtdcHandler);
00363 }
00364
00365 /**
00366     * @brief Sets an LCD Layer visible without
t reloading.
00367     * @param LayerIndex: Visible Layer
00368     * @param State: New state of the specific
d layer
00369     *           This parameter can be one of th
e following values:
00370     *           @arg  ENABLE
00371     *           @arg  DISABLE
00372     * @retval None
00373   */
00374 void BSP_LCD_SetLayerVisible_NoReload(uint32
_t LayerIndex, FunctionalState State)
```

```
00375 {
00376     if(State == ENABLE)
00377     {
00378         __HAL_LTDC_LAYER_ENABLE(&hLtdcHandler, LayerIndex);
00379     }
00380     else
00381     {
00382         __HAL_LTDC_LAYER_DISABLE(&hLtdcHandler, LayerIndex);
00383     }
00384     /* Do not Sets the Reload */
00385 }
00386
00387 /**
00388     * @brief Configures the transparency.
00389     * @param LayerIndex: Layer foreground or background.
00390     * @param Transparency: Transparency
00391     *           This parameter must be a number between Min_Data = 0x00 and Max_Data = 0xFF
00392     * @retval None
00393 */
00394 void BSP_LCD_SetTransparency(uint32_t LayerIndex, uint8_t Transparency)
00395 {
00396     HAL_LTDC_SetAlpha(&hLtdcHandler, Transparency, LayerIndex);
00397 }
00398
00399 /**
00400     * @brief Configures the transparency without reloading.
00401     * @param LayerIndex: Layer foreground or background.
00402     * @param Transparency: Transparency
00403     *           This parameter must be a number
```

```
r between Min_Data = 0x00 and Max_Data = 0xFF
00404     * @retval None
00405     */
00406 void BSP_LCD_SetTransparency_NoReload(uint32
_t LayerIndex, uint8_t Transparency)
00407 {
00408     HAL_LTDC_SetAlpha_NoReload(&hLtdcHandler,
Transparency, LayerIndex);
00409 }
00410
00411 /**
00412     * @brief Sets an LCD layer frame buffer a
ddress.
00413     * @param LayerIndex: Layer foreground or
background
00414     * @param Address: New LCD frame buffer va
lue
00415     * @retval None
00416     */
00417 void BSP_LCD_SetLayerAddress(uint32_t LayerI
ndex, uint32_t Address)
00418 {
00419     HAL_LTDC_SetAddress(&hLtdcHandler, Address
, LayerIndex);
00420 }
00421
00422 /**
00423     * @brief Sets an LCD layer frame buffer a
ddress without reloading.
00424     * @param LayerIndex: Layer foreground or
background
00425     * @param Address: New LCD frame buffer va
lue
00426     * @retval None
00427     */
00428 void BSP_LCD_SetLayerAddress_NoReload(uint32
_t LayerIndex, uint32_t Address)
```

```
00429 {
00430     HAL_LTDC_SetAddress_NoReload(&hLtdcHandler,
00431     , Address, LayerIndex);
00432
00433 /**
00434     * @brief Sets display window.
00435     * @param LayerIndex: Layer index
00436     * @param Xpos: LCD X position
00437     * @param Ypos: LCD Y position
00438     * @param Width: LCD window width
00439     * @param Height: LCD window height
00440     * @retval None
00441 */
00442 void BSP_LCD_SetLayerWindow(uint16_t LayerIn
dex, uint16_t Xpos, uint16_t Ypos, uint16_t Width,
uint16_t Height)
00443 {
00444     /* Reconfigure the layer size */
00445     HAL_LTDC_SetWindowSize(&hLtdcHandler, Widt
h, Height, LayerIndex);
00446
00447     /* Reconfigure the layer position */
00448     HAL_LTDC_SetWindowPosition(&hLtdcHandler,
Xpos, Ypos, LayerIndex);
00449 }
00450
00451 /**
00452     * @brief Sets display window without relo
ading.
00453     * @param LayerIndex: Layer index
00454     * @param Xpos: LCD X position
00455     * @param Ypos: LCD Y position
00456     * @param Width: LCD window width
00457     * @param Height: LCD window height
00458     * @retval None
00459 */
```

```
00460 void BSP_LCD_SetLayerWindow_NoReload(uint16_t LayerIndex, uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height)
00461 {
00462     /* Reconfigure the layer size */
00463     HAL_LTDC_SetWindowSize_NoReload(&hLcdcHandler, Width, Height, LayerIndex);
00464
00465     /* Reconfigure the layer position */
00466     HAL_LTDC_SetWindowPosition_NoReload(&hLcdcHandler, Xpos, Ypos, LayerIndex);
00467 }
00468
00469 /**
00470 * @brief Configures and sets the color keying.
00471 * @param LayerIndex: Layer foreground or background
00472 * @param RGBValue: Color reference
00473 * @retval None
00474 */
00475 void BSP_LCD_SetColorKeying(uint32_t LayerIndex, uint32_t RGBValue)
00476 {
00477     /* Configure and Enable the color Keying for LCD Layer */
00478     HAL_LTDC_ConfigColorKeying(&hLcdcHandler, RGBValue, LayerIndex);
00479     HAL_LTDC_EnableColorKeying(&hLcdcHandler, LayerIndex);
00480 }
00481
00482 /**
00483 * @brief Configures and sets the color keying without reloading.
00484 * @param LayerIndex: Layer foreground or background
```

```
00485     * @param  RGBValue: Color reference
00486     * @retval None
00487     */
00488 void BSP_LCD_SetColorKeying_NoReload(uint32_t LayerIndex, uint32_t RGBValue)
00489 {
00490     /* Configure and Enable the color Keying for LCD Layer */
00491     HAL_LTDC_ConfigColorKeying_NoReload(&hLtdcHandler, RGBValue, LayerIndex);
00492     HAL_LTDC_EnableColorKeying_NoReload(&hLtdcHandler, LayerIndex);
00493 }
00494
00495 /**
00496     * @brief Disables the color keying.
00497     * @param LayerIndex: Layer foreground or background
00498     * @retval None
00499 */
00500 void BSP_LCD_ResetColorKeying(uint32_t LayerIndex)
00501 {
00502     /* Disable the color Keying for LCD Layer */
00503     HAL_LTDC_DisableColorKeying(&hLtdcHandler, LayerIndex);
00504 }
00505
00506 /**
00507     * @brief Disables the color keying without reloading.
00508     * @param LayerIndex: Layer foreground or background
00509     * @retval None
00510 */
00511 void BSP_LCD_ResetColorKeying_NoReload(uint3
```

```
00512     2_t LayerIndex)
00513     00512 {
00514         /* Disable the color Keying for LCD Layer */
00515     */
00516     HAL_LTDC_DisableColorKeying_NoReload(&hLtd
00517 cHandler, LayerIndex);
00518 }
00519 /**
00520     * @brief Disables the color keying without
00521     * reloading.
00522     * @param ReloadType: can be one of the following values
00523     * - LCD_RELOAD_IMMEDIATE
00524     * - LCD_RELOAD_VERTICAL_BLANKING
00525     * @retval None
00526 */
00527 void BSP_LCD_Reload(uint32_t ReloadType)
00528 {
00529     HAL_LTDC_Reload (&hLtdcHandler, ReloadType
00530 );
00531 }
00532 /**
00533     * @brief Sets the LCD text color.
00534     * @param Color: Text color code ARGB(8-8-8-8)
00535     * @retval None
00536 */
00537 void BSP_LCD_SetTextColor(uint32_t Color)
00538 {
00539     DrawProp[ActiveLayer].TextColor = Color;
00540 }
00541 /**
00542     * @brief Gets the LCD text color.
00543     * @retval Used text color.
```

```
00542     */
00543 uint32_t BSP_LCD_GetTextColor(void)
00544 {
00545     return DrawProp[ActiveLayer].TextColor;
00546 }
00547
00548 /**
00549 * @brief Sets the LCD background color.
00550 * @param Color: Layer background color code ARGB(8-8-8-8)
00551 * @retval None
00552 */
00553 void BSP_LCD_SetBackColor(uint32_t Color)
00554 {
00555     DrawProp[ActiveLayer].BackColor = Color;
00556 }
00557
00558 /**
00559 * @brief Gets the LCD background color.
00560 * @retval Used background colour
00561 */
00562 uint32_t BSP_LCD_GetBackColor(void)
00563 {
00564     return DrawProp[ActiveLayer].BackColor;
00565 }
00566
00567 /**
00568 * @brief Sets the LCD text font.
00569 * @param fonts: Layer font to be used
00570 * @retval None
00571 */
00572 void BSP_LCD_SetFont(sFONT *fonts)
00573 {
00574     DrawProp[ActiveLayer].pFont = fonts;
00575 }
00576
00577 /**
```

```
00578     * @brief Gets the LCD text font.  
00579     * @retval Used layer font  
00580     */  
00581 sFONT *BSP_LCD_GetFont(void)  
00582 {  
00583     return DrawProp[ActiveLayer].pFont;  
00584 }  
00585  
00586 /**  
00587     * @brief Reads an LCD pixel.  
00588     * @param Xpos: X position  
00589     * @param Ypos: Y position  
00590     * @retval RGB pixel color  
00591     */  
00592 uint32_t BSP_LCD_ReadPixel(uint16_t Xpos, ui  
nt16_t Ypos)  
00593 {  
00594     uint32_t ret = 0;  
00595  
00596     if(hLtdcHandler.LayerCfg[ActiveLayer].Pi  
xelFormat == LTDC_PIXEL_FORMAT_ARGB8888)  
00597     {  
00598         /* Read data value from SDRAM memory */  
00599         ret = *(__IO uint32_t*) (hLtdcHandler.La  
yerCfg[ActiveLayer].FBStartAdress + (4*(Ypos*BSP_L  
CD_GetXSize() + Xpos)));  
00600     }  
00601     else if(hLtdcHandler.LayerCfg[ActiveLayer]  
.PixelFormat == LTDC_PIXEL_FORMAT_RGB888)  
00602     {  
00603         /* Read data value from SDRAM memory */  
00604         ret = (*(__IO uint32_t*) (hLtdcHandler.La  
yerCfg[ActiveLayer].FBStartAdress + (4*(Ypos*BSP_L  
CD_GetXSize() + Xpos))) & 0x00FFFFFF);  
00605     }  
00606     else if((hLtdcHandler.LayerCfg[ActiveLayer]  
.PixelFormat == LTDC_PIXEL_FORMAT_RGB565) || \
```

```
00607             (hLtdcHandler.LayerCfg[ActiveLayer]
00608                 .PixelFormat == LTDC_PIXEL_FORMAT_ARGB4444) || \
00609             (hLtdcHandler.LayerCfg[ActiveLayer]
00610                 .PixelFormat == LTDC_PIXEL_FORMAT_AL88))
00611     {
00612         /* Read data value from SDRAM memory */
00613         ret = *(__IO uint16_t*) (hLtdcHandler.La
00614             yerCfg[ActiveLayer].FBStartAdress + (2*(Ypos*BSP_L
00615                 CD_GetXSize() + Xpos)));
00616     }
00617 }
```

00618

```
00619     return ret;
00620 }
```

00621

```
00622 /**
00623     * @brief Clears the hole LCD.
00624     * @param Color: Color of the background
00625     * @retval None
00626 */
00627 void BSP_LCD_Clear(uint32_t Color)
00628 {
00629     /* Clear the LCD */
00630     LL_FillBuffer(ActiveLayer, (uint32_t *)(hL
00631         tdcHandler.LayerCfg[ActiveLayer].FBStartAdress), B
00632             SP_LCD_GetXSize(), BSP_LCD_GetYSize(), 0, Color);
00633 }
```

00634

```
00635 /**
00636     * @brief Clears the selected line.
00637     * @param Line: Line to be cleared
```

```
00636     * @retval None
00637     */
00638 void BSP_LCD_ClearStringLine(uint32_t Line)
00639 {
00640     uint32_t color_backup = DrawProp[ActiveLayer].TextColor;
00641     DrawProp[ActiveLayer].TextColor = DrawProp[ActiveLayer].BackColor;
00642
00643     /* Draw rectangle with background color */
00644     BSP_LCD_FillRect(0, (Line * DrawProp[ActiveLayer].pFont->Height), BSP_LCD_GetXSize(), DrawProp[ActiveLayer].pFont->Height);
00645
00646     DrawProp[ActiveLayer].TextColor = color_backup;
00647     BSP_LCD_SetTextColor(DrawProp[ActiveLayer].TextColor);
00648 }
00649
00650 /**
00651     * @brief Displays one character.
00652     * @param Xpos: Start column address
00653     * @param Ypos: Line where to display the
00654     * character shape.
00655     * @param Ascii: Character ascii code
00656     *               This parameter must be a number
00657     * between Min_Data = 0x20 and Max_Data = 0x7E
00658     * @retval None
00659 */
00660 void BSP_LCD_DisplayChar(uint16_t Xpos, uint16_t Ypos, uint8_t Ascii)
00661 {
00662     DrawChar(Xpos, Ypos, &DrawProp[ActiveLayer].pFont->table[(Ascii-' ') *\n00663         DrawProp[ActiveLayer].pFont->Height * ((\n00664         DrawProp[ActiveLayer].pFont->Width + 7) / 8)]);
00665 }
```

```

00662 }
00663
00664 /**
00665 * @brief Displays characters on the LCD.
00666 * @param Xpos: X position (in pixel)
00667 * @param Ypos: Y position (in pixel)
00668 * @param Text: Pointer to string to display on LCD
00669 * @param Mode: Display mode
00670 * This parameter can be one of the following values:
00671 * @arg CENTER_MODE
00672 * @arg RIGHT_MODE
00673 * @arg LEFT_MODE
00674 * @retval None
00675 */
00676 void BSP_LCD_DisplayStringAt(uint16_t Xpos,
00677                                uint16_t Ypos, uint8_t *Text, Text_AlignModeTypdef
00678                                Mode)
00679 {
00680     uint16_t ref_column = 1, i = 0;
00681     uint32_t size = 0, xsiz

```

```
00693         break;
00694     }
00695     case LEFT_MODE:
00696     {
00697         ref_column = Xpos;
00698         break;
00699     }
00700     case RIGHT_MODE:
00701     {
00702         ref_column = - Xpos + ((xsize - size)*
DrawProp[ActiveLayer].pFont->Width);
00703         break;
00704     }
00705     default:
00706     {
00707         ref_column = Xpos;
00708         break;
00709     }
00710 }
00711
00712 /* Check that the Start column is located
in the screen */
00713 if ((ref_column < 1) || (ref_column >= 0x8
000))
00714 {
00715     ref_column = 1;
00716 }
00717
00718 /* Send the string character by character
on LCD */
00719 while ((*Text != 0) & (((BSP_LCD_GetXSize(
) - (i*DrawProp[ActiveLayer].pFont->Width)) & 0xFF
FF) >= DrawProp[ActiveLayer].pFont->Width))
00720 {
00721     /* Display one character on LCD */
00722     BSP_LCD_DisplayChar(ref_column, Ypos, *T
ext);
```

```
00723     /* Decrement the column position by 16 */
00724     ref_column += DrawProp[ActiveLayer].pFont
->Width;
00725     /* Point on the next character */
00726     Text++;
00727     i++;
00728 }
00729 }
00730
00731 /**
00732 * @brief Displays a maximum of 60 characters on the LCD.
00733 * @param Line: Line where to display the character shape
00734 * @param ptr: Pointer to string to display on LCD
00735 * @retval None
00736 */
00737 void BSP_LCD_DisplayStringAtLine(uint16_t Line, uint8_t *ptr)
00738 {
00739     BSP_LCD_DisplayStringAt(0, LINE(Line), ptr,
, LEFT_MODE);
00740 }
00741
00742 /**
00743 * @brief Draws an horizontal line.
00744 * @param Xpos: X position
00745 * @param Ypos: Y position
00746 * @param Length: Line length
00747 * @retval None
00748 */
00749 void BSP_LCD_DrawHLine(uint16_t Xpos, uint16_t Ypos, uint16_t Length)
00750 {
00751     uint32_t Xaddress = 0;
```

```
00752
00753     /* Get the line address */
00754     if(hLtdcHandler.LayerCfg[ActiveLayer].PixelFormat == LTDC_PIXEL_FORMAT_RGB565)
00755     { /* RGB565 format */
00756         Xaddress = (hLtdcHandler.LayerCfg[ActiveLayer].FBStartAdress) + 2*(BSP_LCD_GetXSize())*Ypos
00757         + Xpos);
00758     }
00759     else
00760     { /* ARGB8888 format */
00761         Xaddress = (hLtdcHandler.LayerCfg[ActiveLayer].FBStartAdress) + 4*(BSP_LCD_GetXSize())*Ypos
00762         + Xpos);
00763     }
00764     /* Write line */
00765     LL_FillBuffer(ActiveLayer, (uint32_t *)Xaddress, Length, 1, 0, DrawProp[ActiveLayer].TextColor);
00766
00767 /**
00768 * @brief Draws a vertical line.
00769 * @param Xpos: X position
00770 * @param Ypos: Y position
00771 * @param Length: Line length
00772 * @retval None
00773 */
00774 void BSP_LCD_DrawVLine(uint16_t Xpos, uint16_t Ypos, uint16_t Length)
00775 {
00776     uint32_t Xaddress = 0;
00777
00778     /* Get the line address */
00779     if(hLtdcHandler.LayerCfg[ActiveLayer].PixelFormat == LTDC_PIXEL_FORMAT_RGB565)
```

```

00780     { /* RGB565 format */
00781         Xaddress = (hLtdcHandler.LayerCfg[Active
Layer].FBStartAdress) + 2*(BSP_LCD_GetXSize())*Ypos
+ Xpos);
00782     }
00783     else
00784     { /* ARGB8888 format */
00785         Xaddress = (hLtdcHandler.LayerCfg[Active
Layer].FBStartAdress) + 4*(BSP_LCD_GetXSize())*Ypos
+ Xpos);
00786     }
00787
00788     /* Write line */
00789     LL_FillBuffer(ActiveLayer, (uint32_t *)Xad
dress, 1, Length, (BSP_LCD_GetXSize() - 1), DrawPr
op[ActiveLayer].TextColor);
00790 }
00791
00792 /**
00793 * @brief Draws an uni-line (between two p
oints).
00794 * @param x1: Point 1 X position
00795 * @param y1: Point 1 Y position
00796 * @param x2: Point 2 X position
00797 * @param y2: Point 2 Y position
00798 * @retval None
00799 */
00800 void BSP_LCD_DrawLine(uint16_t x1, uint16_t
y1, uint16_t x2, uint16_t y2)
00801 {
00802     int16_t deltax = 0, deltay = 0, x = 0, y =
0, xinc1 = 0, xinc2 = 0,
00803     yinc1 = 0, yinc2 = 0, den = 0, num = 0, nu
m_add = 0, num_pixels = 0,
00804     curpixel = 0;
00805
00806     deltax = ABS(x2 - x1);           /* The diffe

```

```
rence between the x's */
00807    deltax = ABS(x2 - x1);           /* The difference
rence between the y's */
00808    x = x1;                         /* Start x at the first pixel */
00809    y = y1;                         /* Start y at the first pixel */
00810
00811    if (x2 >= x1)                  /* The x-values are increasing */
00812    {
00813        xinc1 = 1;
00814        xinc2 = 1;
00815    }
00816    else                           /* The x-values are decreasing */
00817    {
00818        xinc1 = -1;
00819        xinc2 = -1;
00820    }
00821
00822    if (y2 >= y1)                  /* The y-values are increasing */
00823    {
00824        yinc1 = 1;
00825        yinc2 = 1;
00826    }
00827    else                           /* The y-values are decreasing */
00828    {
00829        yinc1 = -1;
00830        yinc2 = -1;
00831    }
00832
00833    if (deltax >= deltax)          /* There is at least one x-value for every y-value */
00834    {
```

```

00835     xinc1 = 0;                      /* Don't change the x when numerator >= denominator */
00836     yinc2 = 0;                      /* Don't change the y for every iteration */
00837     den = deltax;
00838     num = deltax / 2;
00839     num_add = deltay;
00840     num_pixels = deltax;           /* There are more x-values than y-values */
00841 }
00842 else                                /* There is at least one y-value for every x-value */
00843 {
00844     xinc2 = 0;                      /* Don't change the x for every iteration */
00845     yinc1 = 0;                      /* Don't change the y when numerator >= denominator */
00846     den = deltay;
00847     num = deltay / 2;
00848     num_add = deltax;
00849     num_pixels = deltay;           /* There are more y-values than x-values */
00850 }
00851
00852 for (curpixel = 0; curpixel <= num_pixels;
00853 curpixel++)
00854     BSP_LCD_DrawPixel(x, y, DrawProp[ActiveLayer].TextColor); /* Draw the current pixel */
00855     num += num_add;
00856     /* Increase the numerator by the top of the fraction */
00857     if (num >= den)
00858         /* Check if numerator >= denominator */
00859     {
00860         num -= den;
00861         /* Calculate the new numerator value */

```

```
00859         x += xinc1;
00860         /* Change the x as appropriate */
00860         y += yinc1;
00861         /* Change the y as appropriate */
00861     }
00862     x += xinc2;
00863     /* Change the x as appropriate */
00863     y += yinc2;
00864     /* Change the y as appropriate */
00864 }
00865 }
00866
00867 /**
00868     * @brief Draws a rectangle.
00869     * @param Xpos: X position
00870     * @param Ypos: Y position
00871     * @param Width: Rectangle width
00872     * @param Height: Rectangle height
00873     * @retval None
00874 */
00875 void BSP_LCD_DrawRect(uint16_t Xpos, uint16_
t Ypos, uint16_t Width, uint16_t Height)
00876 {
00877     /* Draw horizontal lines */
00878     BSP_LCD_DrawHLine(Xpos, Ypos, Width);
00879     BSP_LCD_DrawHLine(Xpos, (Ypos+ Height), Wi
dth);
00880
00881     /* Draw vertical lines */
00882     BSP_LCD_DrawVLine(Xpos, Ypos, Height);
00883     BSP_LCD_DrawVLine((Xpos + Width), Ypos, He
ight);
00884 }
00885
00886 /**
00887     * @brief Draws a circle.
00888     * @param Xpos: X position
```

```
00889     * @param Ypos: Y position
00890     * @param Radius: Circle radius
00891     * @retval None
00892     */
00893 void BSP_LCD_DrawCircle(uint16_t Xpos, uint1
6_t Ypos, uint16_t Radius)
00894 {
00895     int32_t    decision;      /* Decision Variabl
e */
00896     uint32_t   current_x;    /* Current X Value
*/
00897     uint32_t   current_y;    /* Current Y Value
*/
00898
00899     decision = 3 - (Radius << 1);
00900     current_x = 0;
00901     current_y = Radius;
00902
00903     while (current_x <= current_y)
00904     {
00905         BSP_LCD_DrawPixel((Xpos + current_x), (Y
pos - current_y), DrawProp[ActiveLayer].TextColor)
;
00906
00907         BSP_LCD_DrawPixel((Xpos - current_x), (Y
pos - current_y), DrawProp[ActiveLayer].TextColor)
;
00908
00909         BSP_LCD_DrawPixel((Xpos + current_y), (Y
pos - current_x), DrawProp[ActiveLayer].TextColor)
;
00910
00911         BSP_LCD_DrawPixel((Xpos - current_y), (Y
pos - current_x), DrawProp[ActiveLayer].TextColor)
;
00912
00913         BSP_LCD_DrawPixel((Xpos + current_x), (Y
```

```
pos + current_y), DrawProp[ActiveLayer].TextColor)
;
00914    BSP_LCD_DrawPixel((Xpos - current_x), (Y
pos + current_y), DrawProp[ActiveLayer].TextColor)
;
00916    BSP_LCD_DrawPixel((Xpos + current_y), (Y
pos + current_x), DrawProp[ActiveLayer].TextColor)
;
00918    BSP_LCD_DrawPixel((Xpos - current_y), (Y
pos + current_x), DrawProp[ActiveLayer].TextColor)
;
00920
00921    if (decision < 0)
00922    {
00923        decision += (current_x << 2) + 6;
00924    }
00925    else
00926    {
00927        decision += ((current_x - current_y) <
< 2) + 10;
00928        current_y--;
00929    }
00930    current_x++;
00931 }
00932 }
00933
00934 /**
00935 * @brief Draws an poly-line (between many
points).
00936 * @param Points: Pointer to the points ar
ray
00937 * @param PointCount: Number of points
00938 * @retval None
00939 */
```

```
00940 void BSP_LCD_DrawPolygon(pPoint Points, uint
16_t PointCount)
00941 {
00942     int16_t x = 0, y = 0;
00943
00944     if(PointCount < 2)
00945     {
00946         return;
00947     }
00948
00949     BSP_LCD_DrawLine(Points->X, Points->Y, (Po
ints+PointCount-1)->X, (Points+PointCount-1)->Y);
00950
00951     while(--PointCount)
00952     {
00953         x = Points->X;
00954         y = Points->Y;
00955         Points++;
00956         BSP_LCD_DrawLine(x, y, Points->X, Points
->Y);
00957     }
00958 }
00959
00960 /**
00961 * @brief Draws an ellipse on LCD.
00962 * @param Xpos: X position
00963 * @param Ypos: Y position
00964 * @param XRadius: Ellipse X radius
00965 * @param YRadius: Ellipse Y radius
00966 * @retval None
00967 */
00968 void BSP_LCD_DrawEllipse(int Xpos, int Ypos,
int XRadius, int YRadius)
00969 {
00970     int x = 0, y = -YRadius, err = 2-2*XRadius
, e2;
00971     float k = 0, rad1 = 0, rad2 = 0;
```

```
00972
00973     rad1 = XRadius;
00974     rad2 = YRadius;
00975
00976     k = (float)(rad2/rad1);
00977
00978     do {
00979         BSP_LCD_DrawPixel((Xpos-(uint16_t)(x/k))
00980 , (Ypos+y), DrawProp[ActiveLayer].TextColor);
00980         BSP_LCD_DrawPixel((Xpos+(uint16_t)(x/k))
00981 , (Ypos+y), DrawProp[ActiveLayer].TextColor);
00981         BSP_LCD_DrawPixel((Xpos+(uint16_t)(x/k))
00982 , (Ypos-y), DrawProp[ActiveLayer].TextColor);
00982         BSP_LCD_DrawPixel((Xpos-(uint16_t)(x/k))
00983 , (Ypos-y), DrawProp[ActiveLayer].TextColor);

00983
00984     e2 = err;
00985     if (e2 <= x) {
00986         err += ++x*2+1;
00987         if (-y == x && e2 <= y) e2 = 0;
00988     }
00989     if (e2 > y) err += ++y*2+1;
00990 }
00991     while (y <= 0);
00992 }
00993
00994 /**
00995     * @brief Draws a pixel on LCD.
00996     * @param Xpos: X position
00997     * @param Ypos: Y position
00998     * @param RGB_Code: Pixel color in ARGB mode (8-8-8-8)
00999     * @retval None
01000 */
01001 void BSP_LCD_DrawPixel(uint16_t Xpos, uint16_t Ypos, uint32_t RGB_Code)
```

```
01002 {
01003     /* Write data value to all SDRAM memory */
01004     if(hLtdcHandler.LayerCfg[ActiveLayer].PixelFormat == LTDC_PIXEL_FORMAT_RGB565)
01005     { /* RGB565 format */
01006         *(__IO uint16_t*) (hLtdcHandler.LayerCfg[ActiveLayer].FBStartAdress + (2*(Ypos*BSP_LCD_GetXSize() + Xpos))) = (uint16_t)RGB_Code;
01007     }
01008     else
01009     { /* ARGB8888 format */
01010         *(__IO uint32_t*) (hLtdcHandler.LayerCfg[ActiveLayer].FBStartAdress + (4*(Ypos*BSP_LCD_GetXSize() + Xpos))) = RGB_Code;
01011     }
01012 }
01013
01014 /**
01015     * @brief Draws a bitmap picture loaded in
01016     * the internal Flash in ARGB888 format (32 bits per
01017     * pixel).
01018     * @param Xpos: Bmp X position in the LCD
01019     * @param Ypos: Bmp Y position in the LCD
01020     * @param pbmp: Pointer to Bmp picture address in the internal Flash
01021     * @retval None
01022 */
01023 void BSP_LCD_DrawBitmap(uint32_t Xpos, uint32_t Ypos, uint8_t *pbmp)
01024 {
01025     uint32_t index = 0, width = 0, height = 0,
01026     bit_pixel = 0;
01027     uint32_t address;
01028     uint32_t input_color_mode = 0;
01029
01030     /* Get bitmap data address offset */
01031     index = *(__IO uint16_t *) (pbmp + 10);
```

```
01029     index |= (*(__IO uint16_t *) (pbmp + 12))<< 16;
01030
01031     /* Read bitmap width */
01032     width = *(uint16_t *) (pbmp + 18);
01033     width |= (*(uint16_t *) (pbmp + 20)) << 16;
01034
01035     /* Read bitmap height */
01036     height = *(uint16_t *) (pbmp + 22);
01037     height |= (*(uint16_t *) (pbmp + 24)) << 16;
01038
01039     /* Read bit/pixel */
01040     bit_pixel = *(uint16_t *) (pbmp + 28);
01041
01042     /* Set the address */
01043     address = hLtdcHandler.LayerCfg[ActiveLayer].FBStartAdress + (((BSP_LCD_GetXSize() * Ypos) + Xpos)*(4));
01044
01045     /* Get the layer pixel format */
01046     if ((bit_pixel/8) == 4)
01047     {
01048         input_color_mode = CM_ARGB8888;
01049     }
01050     else if ((bit_pixel/8) == 2)
01051     {
01052         input_color_mode = CM_RGB565;
01053     }
01054     else
01055     {
01056         input_color_mode = CM_RGB888;
01057     }
01058
01059     /* Bypass the bitmap header */
01060     pbmp += (index + (width * (height - 1)) * (
```

```
bit_pixel/8));
01061
01062     /* Convert picture to ARGB8888 pixel format */
01063     for(index=0; index < height; index++)
01064     {
01065         /* Pixel format conversion */
01066         LL_ConvertLineToARGB8888((uint32_t *)pbm
01067         p, (uint32_t *)address, width, input_color_mode);
01068         /* Increment the source and destination
01069         buffers */
01070         address+= (BSP_LCD_GetXSize()*4);
01071         pbmp -= width*(bit_pixel/8);
01072     }
01073
01074 /**
01075     * @brief    Draws a full rectangle.
01076     * @param    Xpos: X position
01077     * @param    Ypos: Y position
01078     * @param    Width: Rectangle width
01079     * @param    Height: Rectangle height
01080     * @retval   None
01081 */
01082 void BSP_LCD_FillRect(uint16_t Xpos, uint16_
t Ypos, uint16_t Width, uint16_t Height)
01083 {
01084     uint32_t x_address = 0;
01085
01086     /* Set the text color */
01087     BSP_LCD_SetTextColor(DrawProp[ActiveLayer]
.TextColor);
01088
01089     /* Get the rectangle start address */
01090     if(hLtdcHandler.LayerCfg[ActiveLayer].PixelFormat == LTDC_PIXEL_FORMAT_RGB565)
```

```
01091     { /* RGB565 format */
01092         x_address = (hLtdcHandler.LayerCfg[ActiveLayer].FBStartAdress) + 2*(BSP_LCD_GetXSize()*Ypos + Xpos);
01093     }
01094     else
01095     { /* ARGB8888 format */
01096         x_address = (hLtdcHandler.LayerCfg[ActiveLayer].FBStartAdress) + 4*(BSP_LCD_GetXSize()*Ypos + Xpos);
01097     }
01098     /* Fill the rectangle */
01099     LL_FillBuffer(ActiveLayer, (uint32_t *)x_address, Width, Height, (BSP_LCD_GetXSize() - Width), DrawProp[ActiveLayer].TextColor);
01100 }
01101
01102 /**
01103 * @brief Draws a full circle.
01104 * @param Xpos: X position
01105 * @param Ypos: Y position
01106 * @param Radius: Circle radius
01107 * @retval None
01108 */
01109 void BSP_LCD_FillCircle(uint16_t Xpos, uint16_t Ypos, uint16_t Radius)
01110 {
01111     int32_t decision;      /* Decision Variable */
01112     uint32_t current_x;    /* Current X Value */
01113     uint32_t current_y;    /* Current Y Value */
01114
01115     decision = 3 - (Radius << 1);
01116
01117     current_x = 0;
```

```
01118     current_y = Radius;
01119
01120     BSP_LCD_SetTextColor(DrawProp[ActiveLayer]
01121 .TextColor);
01121
01122     while (current_x <= current_y)
01123     {
01124         if(current_y > 0)
01125         {
01126             BSP_LCD_DrawHLine(Xpos - current_y, Yp
01127 os + current_x, 2*current_y);
01127             BSP_LCD_DrawHLine(Xpos - current_y, Yp
01128 os - current_x, 2*current_y);
01128         }
01129
01130         if(current_x > 0)
01131         {
01132             BSP_LCD_DrawHLine(Xpos - current_x, Yp
01133 os - current_y, 2*current_x);
01133             BSP_LCD_DrawHLine(Xpos - current_x, Yp
01134 os + current_y, 2*current_x);
01134         }
01135         if (decision < 0)
01136         {
01137             decision += (current_x << 2) + 6;
01138         }
01139         else
01140         {
01141             decision += ((current_x - current_y) <
01142 < 2) + 10;
01142             current_y--;
01143         }
01144         current_x++;
01145     }
01146
01147     BSP_LCD_SetTextColor(DrawProp[ActiveLayer]
01148 .TextColor);
```

```
01148     BSP_LCD_DrawCircle(Xpos, Ypos, Radius);
01149 }
01150
01151 /**
01152  * @brief Draws a full poly-line (between
01153  * many points).
01154  * @param Points: Pointer to the points ar-
01155  * ray
01156  * @param PointCount: Number of points
01157  * @retval None
01158 */
01159 void BSP_LCD_FillPolygon(pPoint Points, uint
01160 16_t PointCount)
01161 {
01162     int16_t X = 0, Y = 0, X2 = 0, Y2 = 0, X_ce-
01163 nter = 0, Y_center = 0, X_first = 0, Y_first = 0,
01164 pixelX = 0, pixelY = 0, counter = 0;
01165     uint16_t image_left = 0, image_right = 0,
01166     image_top = 0, image_bottom = 0;
01167
01168     image_left = image_right = Points->X;
01169     image_top= image_bottom = Points->Y;
01170
01171     for(counter = 1; counter < PointCount; cou-
01172 nter++)
01173     {
01174         pixelX = POLY_X(counter);
01175         if(pixelX < image_left)
01176         {
01177             image_left = pixelX;
01178         }
01179         if(pixelX > image_right)
01180         {
01181             image_right = pixelX;
01182         }
01183
01184         pixelY = POLY_Y(counter);
```

```
01178     if(pixelY < image_top)
01179     {
01180         image_top = pixelY;
01181     }
01182     if(pixelY > image_bottom)
01183     {
01184         image_bottom = pixelY;
01185     }
01186 }
01187
01188 if(PointCount < 2)
01189 {
01190     return;
01191 }
01192
01193 X_center = (image_left + image_right)/2;
01194 Y_center = (image_bottom + image_top)/2;
01195
01196 X_first = Points->X;
01197 Y_first = Points->Y;
01198
01199 while(--PointCount)
01200 {
01201     X = Points->X;
01202     Y = Points->Y;
01203     Points++;
01204     X2 = Points->X;
01205     Y2 = Points->Y;
01206
01207     FillTriangle(X, X2, X_center, Y, Y2, Y_center);
01208     FillTriangle(X, X_center, X2, Y, Y_center, Y2);
01209     FillTriangle(X_center, X2, X, Y_center, Y2, Y);
01210 }
01211
```

```
01212     FillTriangle(X_first, X2, X_center, Y_firs
t, Y2, Y_center);
01213     FillTriangle(X_first, X_center, X2, Y_firs
t, Y_center, Y2);
01214     FillTriangle(X_center, X2, X_first, Y_cent
er, Y2, Y_first);
01215 }
01216
01217 /**
01218 * @brief Draws a full ellipse.
01219 * @param Xpos: X position
01220 * @param Ypos: Y position
01221 * @param XRadius: Ellipse X radius
01222 * @param YRadius: Ellipse Y radius
01223 * @retval None
01224 */
01225 void BSP_LCD_FillEllipse(int Xpos, int Ypos,
int XRadius, int YRadius)
01226 {
01227     int x = 0, y = -YRadius, err = 2-2*XRadius
, e2;
01228     float k = 0, rad1 = 0, rad2 = 0;
01229
01230     rad1 = XRadius;
01231     rad2 = YRadius;
01232
01233     k = (float)(rad2/rad1);
01234
01235     do
01236     {
01237         BSP_LCD_DrawHLine((Xpos-(uint16_t)(x/k))
, (Ypos+y), (2*(uint16_t)(x/k) + 1));
01238         BSP_LCD_DrawHLine((Xpos-(uint16_t)(x/k))
, (Ypos-y), (2*(uint16_t)(x/k) + 1));
01239
01240         e2 = err;
01241         if (e2 <= x)
```

```

01242     {
01243         err += ++x*2+1;
01244         if (-y == x && e2 <= y) e2 = 0;
01245     }
01246     if (e2 > y) err += ++y*2+1;
01247 }
01248 while (y <= 0);
01249 }
01250
01251 /**
01252 * @brief Enables the display.
01253 * @retval None
01254 */
01255 void BSP_LCD_DisplayOn(void)
01256 {
01257     /* Display On */
01258     __HAL_LTDC_ENABLE(&hLtdcHandler);
01259     HAL_GPIO_WritePin(LCD_DISP_GPIO_PORT, LCD_
DISP_PIN, GPIO_PIN_SET);           /* Assert LCD_DISP
pin */
01260     HAL_GPIO_WritePin(LCD_BL_CTRL_GPIO_PORT, L
CD_BL_CTRL_PIN, GPIO_PIN_SET);    /* Assert LCD_BL_C
TRL pin */
01261 }
01262
01263 /**
01264 * @brief Disables the display.
01265 * @retval None
01266 */
01267 void BSP_LCD_DisplayOff(void)
01268 {
01269     /* Display Off */
01270     __HAL_LTDC_DISABLE(&hLtdcHandler);
01271     HAL_GPIO_WritePin(LCD_DISP_GPIO_PORT, LCD_
DISP_PIN, GPIO_PIN_RESET);        /* De-assert LCD_D
ISP pin */
01272     HAL_GPIO_WritePin(LCD_BL_CTRL_GPIO_PORT, L

```

```
CD_BL_CTRL_PIN, GPIO_PIN_RESET); /* De-assert LCD_B
L_CTRL pin */
01273 }
01274
01275 /**
01276     * @brief Initializes the LTDC MSP.
01277     * @param hltcd: LTDC handle
01278     * @param Params
01279     * @retval None
01280 */
01281 __weak void BSP_LCD_MspInit(LTDC_HandleTypeDef *hltcd, void *Params)
01282 {
01283     GPIO_InitTypeDef gpio_init_structure;
01284
01285     /* Enable the LTDC and DMA2D clocks */
01286     __HAL_RCC_LTDC_CLK_ENABLE();
01287     __HAL_RCC_DMA2D_CLK_ENABLE();
01288
01289     /* Enable GPIOs clock */
01290     __HAL_RCC_GPIOE_CLK_ENABLE();
01291     __HAL_RCC_GPIOG_CLK_ENABLE();
01292     __HAL_RCC_GPIOI_CLK_ENABLE();
01293     __HAL_RCC_GPIOJ_CLK_ENABLE();
01294     __HAL_RCC_GPIOK_CLK_ENABLE();
01295     LCD_DISP_GPIO_CLK_ENABLE();
01296     LCD_BL_CTRL_GPIO_CLK_ENABLE();
01297
01298     /*** LTDC Pins configuration ***/
01299     /* GPIOE configuration */
01300     gpio_init_structure.Pin = GPIO_PIN_4;
01301     gpio_init_structure.Mode = GPIO_MODE_AF_PP;
01302     gpio_init_structure.Pull = GPIO_NOPULL;
01303     gpio_init_structure.Speed = GPIO_SPEED
```

```
_FAST;
01304     gpio_init_structure.Alternate = GPIO_AF14_
LTDC;
01305     HAL_GPIO_Init(GPIOE, &gpio_init_structure)
;
01306
01307 /* GPIOG configuration */
01308     gpio_init_structure.Pin      = GPIO_PIN_1
2;
01309     gpio_init_structure.Mode    = GPIO_MODE_
AF_PP;
01310     gpio_init_structure.Alternate = GPIO_AF9_L
TDC;
01311     HAL_GPIO_Init(GPIOG, &gpio_init_structure)
;
01312
01313 /* GPIOI LTDC alternate configuration */
01314     gpio_init_structure.Pin      = GPIO_PIN_9
| GPIO_PIN_10 | \
01315                               GPIO_PIN_1
3 | GPIO_PIN_14 | GPIO_PIN_15;
01316     gpio_init_structure.Mode    = GPIO_MODE_
AF_PP;
01317     gpio_init_structure.Alternate = GPIO_AF14_
LTDC;
01318     HAL_GPIO_Init(GPIOI, &gpio_init_structure)
;
01319
01320 /* GPIOJ configuration */
01321     gpio_init_structure.Pin      = GPIO_PIN_0
| GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3 | \
01322                               GPIO_PIN_4
| GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7 | \
01323                               GPIO_PIN_8
| GPIO_PIN_9 | GPIO_PIN_10 | GPIO_PIN_11 | \
01324                               GPIO_PIN_1
3 | GPIO_PIN_14 | GPIO_PIN_15;
```

```
01325     gpio_init_structure.Mode      = GPIO_MODE_
AF_PP;
01326     gpio_init_structure.Alternate = GPIO_AF14_
LTDC;
01327     HAL_GPIO_Init(GPIOJ, &gpio_init_structure)
;
01328
01329     /* GPIOK configuration */
01330     gpio_init_structure.Pin      = GPIO_PIN_0
| GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_4 | \
01331                               GPIO_PIN_5
| GPIO_PIN_6 | GPIO_PIN_7;
01332     gpio_init_structure.Mode      = GPIO_MODE_
AF_PP;
01333     gpio_init_structure.Alternate = GPIO_AF14_
LTDC;
01334     HAL_GPIO_Init(GPIOK, &gpio_init_structure)
;
01335
01336     /* LCD_DISP GPIO configuration */
01337     gpio_init_structure.Pin      = LCD_DISP_P
IN; /* LCD_DISP pin has to be manually control
led */
01338     gpio_init_structure.Mode      = GPIO_MODE_
OUTPUT_PP;
01339     HAL_GPIO_Init(LCD_DISP_GPIO_PORT, &gpio_in
it_structure);
01340
01341     /* LCD_BL_CTRL GPIO configuration */
01342     gpio_init_structure.Pin      = LCD_BL_CTR
L_PIN; /* LCD_BL_CTRL pin has to be manually cont
rolled */
01343     gpio_init_structure.Mode      = GPIO_MODE_
OUTPUT_PP;
01344     HAL_GPIO_Init(LCD_BL_CTRL_GPIO_PORT, &gpio
_init_structure);
01345 }
```

```
01346
01347 /**
01348     * @brief  DeInitializes BSP_LCD MSP.
01349     * @param  hltdc: LTDC handle
01350     * @param  Params
01351     * @retval None
01352 */
01353 __weak void BSP_LCD_MspDeInit(LTDC_HandleTypeDefTyp
eDef *hltdc, void *Params)
01354 {
01355     GPIO_InitTypeDef gpio_init_structure;
01356
01357     /* Disable LTDC block */
01358     __HAL_LTDC_DISABLE(hltdc);
01359
01360     /* LTDC Pins deactivation */
01361
01362     /* GPIOE deactivation */
01363     gpio_init_structure.Pin          = GPIO_PIN_4
01364 ;
01365     HAL_GPIO_DeInit(GPIOE, gpio_init_structure
01366 .Pin);
01367
01368     /* GPIOG deactivation */
01369     gpio_init_structure.Pin          = GPIO_PIN_1
01370 2;
01371     HAL_GPIO_DeInit(GPIOG, gpio_init_structure
01372 .Pin);
01373
01374     /* GPIOI deactivation */
01375     gpio_init_structure.Pin          = GPIO_PIN_8
01376 | GPIO_PIN_9 | GPIO_PIN_10 | GPIO_PIN_12 | \
01377                                     GPIO_PIN_1
01378 3 | GPIO_PIN_14 | GPIO_PIN_15;
01379     HAL_GPIO_DeInit(GPIOI, gpio_init_structure
01380 .Pin);
01381
```

```
01375     /* GPIOJ deactivation */
01376     gpio_init_structure.Pin      = GPIO_PIN_0
01377           | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3 | \
01378                           GPIO_PIN_4
01379           | GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7 | \
01380                           GPIO_PIN_8
01381           | GPIO_PIN_9 | GPIO_PIN_10 | GPIO_PIN_11 | \
01382                           GPIO_PIN_1
01383           | GPIO_PIN_14 | GPIO_PIN_15;
01384     HAL_GPIO_DeInit(GPIOJ, gpio_init_structure
01385 .Pin);
01386
01387     /* GPIOK deactivation */
01388     gpio_init_structure.Pin      = GPIO_PIN_0
01389           | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_4 | \
01390                           GPIO_PIN_5
01391           | GPIO_PIN_6 | GPIO_PIN_7;
01392     HAL_GPIO_DeInit(GPIOK, gpio_init_structure
01393 .Pin);
01394
01395     /* Disable LTDC clock */
01396     __HAL_RCC_LTDC_CLK_DISABLE();
01397
01398     /* GPIO pins clock can be shut down in the
01399        application
01400        by surcharging this __weak function */
01401 }
01402
01403
01404 /**
01405  * @brief Clock Config.
01406  * @param hltcd: LTDC handle
01407  * @param Params
01408  * @note This API is called by BSP_LCD_In
01409  * it()
01410  *          Being __weak it can be overwritt
01411  * en by the application
01412  * @retval None
```

```
01401     */
01402 __weak void BSP_LCD_ClockConfig(LTDC_HandleTypeDefDef *hltcd, void *Params)
01403 {
01404     static RCC_PeriphCLKInitTypeDef    periph_clk_init_struct;
01405
01406     /* RK043FN48H LCD clock configuration */
01407     /* PLLSAI_VCO Input = HSE_VALUE/PLL_M = 1
Mhz */
01408     /* PLLSAI_VCO Output = PLLSAI_VCO Input *
PLLSAIN = 192 Mhz */
01409     /* PLLLCDCLK = PLLSAI_VCO Output/PLLSAIR =
192/5 = 38.4 Mhz */
01410     /* LTDC clock frequency = PLLLCDCLK / LTDC
_PLLSAI_DIVR_4 = 38.4/4 = 9.6Mhz */
01411     periph_clk_init_struct.PeriphClockSelection = RCC_PERIPHCLK_LTDC;
01412     periph_clk_init_struct.PLLSAI.PLLSAIN = 19
2;
01413     periph_clk_init_struct.PLLSAI.PLLSAIR = RK
043FN48H_FREQUENCY_DIVIDER;
01414     periph_clk_init_struct.PLLSAIDivR = RCC_PL
LSAIDIVR_4;
01415     HAL_RCCEx_PeriphCLKConfig(&periph_clk_init
_struct);
01416 }
01417
01418
01419 /*****
01420 ***** Static Functions
01421 *****/
01422
01423 /**
01424     * @brief Draws a character on LCD.
```

```
01425     * @param Xpos: Line where to display the
01426     * character shape
01427     * @param Ypos: Start column address
01428     * @param c: Pointer to the character data
01429     * @retval None
01429 */
01430 static void DrawChar(uint16_t Xpos, uint16_t
01431 Ypos, const uint8_t *c)
01431 {
01432     uint32_t i = 0, j = 0;
01433     uint16_t height, width;
01434     uint8_t offset;
01435     uint8_t *pchar;
01436     uint32_t line;
01437
01438     height = DrawProp[ActiveLayer].pFont->Heig
01439 ht;
01440     width = DrawProp[ActiveLayer].pFont->Widt
01441 h;
01442
01443     offset = 8 *((width + 7)/8) - width ;
01444
01445     for(i = 0; i < height; i++)
01446     {
01447         pchar = ((uint8_t *)c + (width + 7)/8 *
01448 i);
01449
01450         switch(((width + 7)/8))
01451         {
01452             case 1:
01453                 line = pchar[0];
01454                 break;
01455
01456             case 2:
01457                 line = (pchar[0]<< 8) | pchar[1];
```

```
01456         break;
01457
01458     case 3:
01459     default:
01460         line = (pchar[0]<< 16) | (pchar[1]<<
8) | pchar[2];
01461         break;
01462     }
01463
01464     for (j = 0; j < width; j++)
01465     {
01466         if(line & (1 << (width- j + offset- 1)
))
01467         {
01468             BSP_LCD_DrawPixel((Xpos + j), Ypos,
DrawProp[ActiveLayer].TextColor);
01469         }
01470         else
01471         {
01472             BSP_LCD_DrawPixel((Xpos + j), Ypos,
DrawProp[ActiveLayer].BackColor);
01473         }
01474     }
01475     Ypos++;
01476 }
01477 }
01478
01479 /**
01480 * @brief Fills a triangle (between 3 points).
01481 * @param x1: Point 1 X position
01482 * @param y1: Point 1 Y position
01483 * @param x2: Point 2 X position
01484 * @param y2: Point 2 Y position
01485 * @param x3: Point 3 X position
01486 * @param y3: Point 3 Y position
01487 * @retval None
```

```
01488  */
01489 static void FillTriangle(uint16_t x1, uint16
_t x2, uint16_t x3, uint16_t y1, uint16_t y2, uint
16_t y3)
01490 {
01491     int16_t deltax = 0, deltay = 0, x = 0, y =
0, xinc1 = 0, xinc2 = 0,
01492     yinc1 = 0, yinc2 = 0, den = 0, num = 0, nu
m_add = 0, num_pixels = 0,
01493     curpixel = 0;
01494
01495     deltax = ABS(x2 - x1);           /* The diffe
rence between the x's */
01496     deltay = ABS(y2 - y1);           /* The diffe
rence between the y's */
01497     x = x1;                         /* Start x o
ff at the first pixel */
01498     y = y1;                         /* Start y o
ff at the first pixel */
01499
01500     if (x2 >= x1)                  /* The x-val
ues are increasing */
01501     {
01502         xinc1 = 1;
01503         xinc2 = 1;
01504     }
01505     else                           /* The x-val
ues are decreasing */
01506     {
01507         xinc1 = -1;
01508         xinc2 = -1;
01509     }
01510
01511     if (y2 >= y1)                  /* The y-val
ues are increasing */
01512     {
01513         yinc1 = 1;
```

```

01514     yinc2 = 1;
01515 }
01516 else /* The y-values are decreasing */
01517 {
01518     yinc1 = -1;
01519     yinc2 = -1;
01520 }
01521
01522 if (deltax >= deltax) /* There is at least one x-value for every y-value */
01523 {
01524     xinc1 = 0; /* Don't change the x when numerator >= denominator */
01525     yinc2 = 0; /* Don't change the y for every iteration */
01526     den = deltax;
01527     num = deltax / 2;
01528     num_add = deltax;
01529     num_pixels = deltax; /* There are more x-values than y-values */
01530 }
01531 else /* There is at least one y-value for every x-value */
01532 {
01533     xinc2 = 0; /* Don't change the x for every iteration */
01534     yinc1 = 0; /* Don't change the y when numerator >= denominator */
01535     den = deltax;
01536     num = deltax / 2;
01537     num_add = deltax;
01538     num_pixels = deltax; /* There are more y-values than x-values */
01539 }
01540
01541 for (curpixel = 0; curpixel <= num_pixels;

```

```

    curpixel++)
01542  {
01543      BSP_LCD_DrawLine(x, y, x3, y3);
01544
01545      num += num_add;           /* Increase
the numerator by the top of the fraction */
01546      if (num >= den)          /* Check if
numerator >= denominator */
01547      {
01548          num -= den;          /* Calculate
the new numerator value */
01549          x += xinc1;          /* Change th
e x as appropriate */
01550          y += yinc1;          /* Change th
e y as appropriate */
01551      }
01552      x += xinc2;          /* Change th
e x as appropriate */
01553      y += yinc2;          /* Change th
e y as appropriate */
01554  }
01555 }
01556
01557 /**
01558  * @brief Fills a buffer.
01559  * @param LayerIndex: Layer index
01560  * @param pDst: Pointer to destination buf
fer
01561  * @param xSize: Buffer width
01562  * @param ySize: Buffer height
01563  * @param OffLine: Offset
01564  * @param ColorIndex: Color index
01565  * @retval None
01566 */
01567 static void LL_FillBuffer(uint32_t LayerInde
x, void *pDst, uint32_t xSize, uint32_t ySize, uin
t32_t OffLine, uint32_t ColorIndex)

```

```
01568 {
01569     /* Register to memory mode with ARGB8888 as color Mode */
01570     hDma2dHandler.Init.Mode          = DMA2D_R2
01571     M;
01571     if(hLtdcHandler.LayerCfg[ActiveLayer].PixelFormat == LTDC_PIXEL_FORMAT_RGB565)
01572     { /* RGB565 format */
01573         hDma2dHandler.Init.ColorMode      = DMA2D_
01574         RGB565;
01574     }
01575     else
01576     { /* ARGB8888 format */
01577         hDma2dHandler.Init.ColorMode      = DMA2D_
01578         ARGB8888;
01578     }
01579     hDma2dHandler.Init.OutputOffset = OffLine;

01580
01581     hDma2dHandler.Instance = DMA2D;
01582
01583     /* DMA2D Initialization */
01584     if(HAL_DMA2D_Init(&hDma2dHandler) == HAL_OK)
01585     {
01586         if(HAL_DMA2D_ConfigLayer(&hDma2dHandler,
01587             LayerIndex) == HAL_OK)
01587         {
01588             if (HAL_DMA2D_Start(&hDma2dHandler, ColorIndex,
01589             (uint32_t)pDst, xSize, ySize) == HAL_OK)
01589             {
01590                 /* Polling For DMA transfer */
01591                 HAL_DMA2D_PollForTransfer(&hDma2dHandler, 10);
01592             }
01593         }
01594     }
```

```
01595 }
01596
01597 /**
01598     * @brief Converts a line to an ARGB8888 pixel format.
01599     * @param pSrc: Pointer to source buffer
01600     * @param pDst: Output color
01601     * @param xSize: Buffer width
01602     * @param ColorMode: Input color mode
01603     * @retval None
01604 */
01605 static void LL_ConvertLineToARGB8888(void *pSrc, void *pDst, uint32_t xSize, uint32_t ColorMode)
01606 {
01607     /* Configure the DMA2D Mode, Color Mode and output offset */
01608     hDMA2DHandler.Init.Mode = DMA2D_M2M_PFC;
01609     hDMA2DHandler.Init.ColorMode = DMA2D_ARGB8888;
01610     hDMA2DHandler.Init.OutputOffset = 0;
01611
01612     /* Foreground Configuration */
01613     hDMA2DHandler.LayerCfg[1].AlphaMode = DMA2D_NO_MODIF_ALPHA;
01614     hDMA2DHandler.LayerCfg[1].InputAlpha = 0xFF;
01615     hDMA2DHandler.LayerCfg[1].InputColorMode = ColorMode;
01616     hDMA2DHandler.LayerCfg[1].InputOffset = 0;
01617
01618     hDMA2DHandler.Instance = DMA2D;
01619
01620     /* DMA2D Initialization */
01621     if(HAL_DMA2D_Init(&hDMA2DHandler) == HAL_OK)
```

```

01622     {
01623         if( HAL_DMA2D_ConfigLayer(&hDma2dHandler,
01624             1) == HAL_OK)
01624     {
01625         if (HAL_DMA2D_Start(&hDma2dHandler, (u
01626             int32_t)pSrc, (uint32_t)pDst, xSize, 1) == HAL_OK)
01626     {
01627         /* Polling For DMA transfer */
01628         HAL_DMA2D_PollForTransfer(&hDma2dHan
01629             dler, 10);
01629     }
01630 }
01631 }
01632 }
01633
01634 /**
01635 * @}
01636 */
01637
01638 /**
01639 * @}
01640 */
01641
01642 /**
01643 * @}
01644 */
01645
01646 /**
01647 * @}
01648 */
01649
01650 /***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/

```

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Defines				STM32746G_DISCOVERY_SDRAM Exported Constants
STM32746G_DISCOVERY_SDRAM Exported Types				

Defines

```
#define SDRAM_DEVICE_ADDR ((uint32_t)0xC0000000)
#define SDRAM_DEVICE_SIZE ((uint32_t)0x800000) /* SDRAM devi
size in MBytes */
#define SDRAM_MEMORY_WIDTH FMC_SDRAM_MEM_BUS_WID
#define SDCLK_PERIOD FMC_SDRAM_CLOCK_PERIOD_2
#define REFRESH_COUNT ((uint32_t)0x0603) /* SDRAM refresh cou
(100Mhz SD clock) */
#define SDRAM_TIMEOUT ((uint32_t)0xFFFF)
#define __DMAx_CLK_ENABLE __HAL_RCC_DMA2_CLK_ENABLE
#define __DMAx_CLK_DISABLE __HAL_RCC_DMA2_CLK_DISABLE
#define SDRAM_DMAx_CHANNEL DMA_CHANNEL_0
#define SDRAM_DMAx_STREAM DMA2_Stream0
#define SDRAM_DMAx_IRQn DMA2_Stream0_IRQn
#define BSP_SDRAM_DMA_IRQHandler DMA2_Stream0_IRQHandler
```

Define Documentation

```
#define __DMAx_CLK_DISABLE __HAL_RCC_DMA2_CLK_DISAB
```

Definition at line [90](#) of file [stm32746g_discovery_sram.h](#).

```
#define __DMAx_CLK_ENABLE __HAL_RCC_DMA2_CLK_ENABL
```

Definition at line [89](#) of file [stm32746g_discovery_sram.h](#).

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

```
#define BSP_SDRAM_DMA_IRQHandler DMA2_Stream0_IRQHandler
```

Definition at line [94](#) of file [stm32746g_discovery_sram.h](#).

```
#define REFRESH_COUNT ((uint32_t)0x0603) /* SDRAM refresh count */
```

Definition at line [84](#) of file [stm32746g_discovery_sram.h](#).

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

```
#define SDCLK_PERIOD FMC_SDRAM_CLOCK_PERIOD_2
```

Definition at line [81](#) of file [stm32746g_discovery_sram.h](#).

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

```
#define SDRAM_DEVICE_ADDR ((uint32_t)0xC0000000)
```

Definition at line [75](#) of file [stm32746g_discovery_sram.h](#).

```
#define SDRAM_DEVICE_SIZE ((uint32_t)0x800000) /* SDRAM dev
```

Definition at line [76](#) of file [stm32746g_discovery_sdram.h](#).

```
#define SDRAM_DMAx_CHANNEL DMA_CHANNEL_0
```

Definition at line [91](#) of file [stm32746g_discovery_sdram.h](#).

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

```
#define SDRAM_DMAx_IRQn DMA2_Stream0_IRQn
```

Definition at line [93](#) of file [stm32746g_discovery_sdram.h](#).

Referenced by [BSP_SDRAM_MspDeInit\(\)](#), and
[BSP_SDRAM_MspInit\(\)](#).

```
#define SDRAM_DMAx_STREAM DMA2_Stream0
```

Definition at line [92](#) of file [stm32746g_discovery_sdram.h](#).

Referenced by [BSP_SDRAM_MspDeInit\(\)](#), and
[BSP_SDRAM_MspInit\(\)](#).

```
#define SDRAM_MEMORY_WIDTH FMC_SDRAM_MEM_BUS_WIDTH
```

Definition at line [79](#) of file [stm32746g_discovery_sdram.h](#).

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

```
#define SDRAM_TIMEOUT ((uint32_t)0xFFFF)
```

Definition at line **86** of file [stm32746g_discovery_sram.h](#).

Referenced by [BSP_SDRAM_Initialization_sequence\(\)](#), and
[BSP_SDRAM_Sendcmd\(\)](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Defines				STM32746G_DISCOVERY_SD Exported Constants
STM32746G_DISCOVERY_SD				

Defines

```
#define SD_PRESENT ((uint8_t)0x01)
#define SD_NOT_PRESENT ((uint8_t)0x00)
#define SD_DATATIMEOUT ((uint32_t)100000000)
#define __DMAx_TxRx_CLK_ENABLE __HAL_RCC_DMA2_CLK_E
#define SD_DMAx_Tx_CHANNEL DMA_CHANNEL_4
#define SD_DMAx_Rx_CHANNEL DMA_CHANNEL_4
#define SD_DMAx_Tx_STREAM DMA2_Stream6
#define SD_DMAx_Rx_STREAM DMA2_Stream3
#define SD_DMAx_Tx_IRQn DMA2_Stream6_IRQn
#define SD_DMAx_Rx_IRQn DMA2_Stream3_IRQn
#define BSP_SDMMC_IRQHandler SDMMC1_IRQHandler
#define BSP_SDMMC_DMA_Tx_IRQHandler DMA2_Stream6_IRQHandler
#define BSP_SDMMC_DMA_Rx_IRQHandler DMA2_Stream3_IRQHandler
#define SD_DetectIRQHandler() HAL_GPIO_EXTI_IRQHandler(SD_
```

Define Documentation

```
#define __DMAx_TxRx_CLK_ENABLE __HAL_RCC_DMA2_CLK_E
```

Definition at line **96** of file [stm32746g_discovery_sd.h](#).

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

```
#define BSP_SDMMC_DMA_Rx_IRQHandler DMA2_Stream3_IRQHandler
```

Definition at line **105** of file [stm32746g_discovery_sd.h](#).

```
#define BSP_SDMMC_DMA_Tx_IRQHandler DMA2_Stream6_IRQHandler
```

Definition at line **104** of file [stm32746g_discovery_sd.h](#).

```
#define BSP_SDMMC_IRQHandler SDMMC1_IRQHandler
```

Definition at line **103** of file [stm32746g_discovery_sd.h](#).

```
#define SD_DATATIMEOUT ((uint32_t)100000000)
```

Definition at line **93** of file [stm32746g_discovery_sd.h](#).

```
#define SD_DetectIRQHandler() HAL_GPIO_EXTI_IRQHandler(S
```

Definition at line **106** of file [stm32746g_discovery_sd.h](#).

```
#define SD_DMAx_Rx_CHANNEL DMA_CHANNEL_4
```

Definition at line **98** of file [stm32746g_discovery_sd.h](#).

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

#define SD_DMAx_Rx_IRQn DMA2_Stream3_IRQn

Definition at line **102** of file [stm32746g_discovery_sd.h](#).

Referenced by [BSP_SD_MspDeInit\(\)](#), and [BSP_SD_MspInit\(\)](#).

#define SD_DMAx_Rx_STREAM DMA2_Stream3

Definition at line **100** of file [stm32746g_discovery_sd.h](#).

Referenced by [BSP_SD_MspDeInit\(\)](#), and [BSP_SD_MspInit\(\)](#).

#define SD_DMAx_Tx_CHANNEL DMA_CHANNEL_4

Definition at line **97** of file [stm32746g_discovery_sd.h](#).

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

#define SD_DMAx_Tx_IRQn DMA2_Stream6_IRQn

Definition at line **101** of file [stm32746g_discovery_sd.h](#).

Referenced by [BSP_SD_MspDeInit\(\)](#), and [BSP_SD_MspInit\(\)](#).

#define SD_DMAx_Tx_STREAM DMA2_Stream6

Definition at line **99** of file [stm32746g_discovery_sd.h](#).

Referenced by [BSP_SD_MspDeInit\(\)](#), and [BSP_SD_MspInit\(\)](#).

#define SD_NOT_PRESENT ((uint8_t)0x00)

Definition at line **91** of file [stm32746g_discovery_sd.h](#).

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

#define SD_PRESENT ((uint8_t)0x01)

Definition at line **90** of file [stm32746g_discovery_sd.h](#).

Referenced by [BSP_SD_Init\(\)](#), and [BSP_SD_IsDetected\(\)](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Defines				STM32746G_DISCOVERY_LOW_LEVEL Private Defines
STM32746G_DISCOVERY_LOW_LEVEL				

Defines

```
#define __STM32746G_DISCO_BSP_VERSION_MAIN (0x02)
    STM32746G DISCOVERY BSP Driver version number
    V2.0.0.

#define __STM32746G_DISCO_BSP_VERSION_SUB1 (0x00)
#define __STM32746G_DISCO_BSP_VERSION_SUB2 (0x00)
#define __STM32746G_DISCO_BSP_VERSION_RC (0x00)
#define __STM32746G_DISCO_BSP_VERSION
```

Define Documentation

#define __STM32746G_DISCO_BSP_VERSION

Value:

```
((__STM32746G_DISCO_BSP_VERSION_MAIN << 24) \  
 | ( __  
| STM32746G_DISCO_BSP_VERSION_SUB1 << 16) \  
 | ( __  
| STM32746G_DISCO_BSP_VERSION_SUB2 << 8 ) \  
 | ( __  
| STM32746G_DISCO_BSP_VERSION_RC))
```

Definition at line **72** of file [stm32746g_discovery.c](#).

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

#define __STM32746G_DISCO_BSP_VERSION_MAIN (0x02)

STM32746G DISCOVERY BSP Driver version number V2.0.0.

[31:24] main version

Definition at line **68** of file [stm32746g_discovery.c](#).

#define __STM32746G_DISCO_BSP_VERSION_RC (0x00)

[7:0] release candidate

Definition at line **71** of file [stm32746g_discovery.c](#).

#define __STM32746G_DISCO_BSP_VERSION_SUB1 (0x00)

[23:16] sub1 version

Definition at line **69** of file [stm32746g_discovery.c](#).

```
#define __STM32746G_DISCO_BSP_VERSION_SUB2 (0x00)
```

[15:8] sub2 version

Definition at line **70** of file [stm32746g_discovery.c](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Defines

STM32746G_DISCOVERY_LCD Private Macros

[STM32746G_DISCOVERY_LCD](#)

Defines

```
#define ABS(X) ((X) > 0 ? (X) : -(X))
```

Define Documentation

```
#define ABS( X ) ((X) > 0 ? (X) : -(X))
```

Definition at line [111](#) of file [stm32746g_discovery_lcd.c](#).

Referenced by [BSP_LCD_DrawLine\(\)](#), and [FillTriangle\(\)](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Variables
STM32746G_DISCOVERY_LCD Private Variables				
STM32746G_DISCOVERY_LCD				

Variables

```
LTDC_HandleTypeDef hLtdcHandler  
static DMA2D_HandleTypeDef hDma2dHandler  
    static uint32_t ActiveLayer = 0  
static LCD_DrawPropTypeDef DrawProp [MAX_LAYER_NUMBER]
```

Variable Documentation

uint32_t ActiveLayer = 0 [static]

Definition at line 123 of file [stm32746g_discovery_lcd.c](#).

Referenced by [BSP_LCD_Clear\(\)](#), [BSP_LCD_ClearStringLine\(\)](#), [BSP_LCD_DisplayChar\(\)](#), [BSP_LCD_DisplayStringAt\(\)](#), [BSP_LCD_DrawBitmap\(\)](#), [BSP_LCD_DrawCircle\(\)](#), [BSP_LCD_DrawEllipse\(\)](#), [BSP_LCD_DrawHLine\(\)](#), [BSP_LCD_DrawLine\(\)](#), [BSP_LCD_DrawPixel\(\)](#), [BSP_LCD_DrawVLine\(\)](#), [BSP_LCD_FillCircle\(\)](#), [BSP_LCD_FillRect\(\)](#), [BSP_LCD_GetBackColor\(\)](#), [BSP_LCD_GetFont\(\)](#), [BSP_LCD_GetTextColor\(\)](#), [BSP_LCDGetXSize\(\)](#), [BSP_LCDGetYSize\(\)](#), [BSP_LCD_ReadPixel\(\)](#), [BSP_LCD_SelectLayer\(\)](#), [BSP_LCD_SetBackColor\(\)](#), [BSP_LCD_SetFont\(\)](#), [BSP_LCD_SetTextColor\(\)](#), [BSP_LCD_SetXSize\(\)](#), [BSP_LCD_SetYSize\(\)](#), [DrawChar\(\)](#), and [LL_FillBuffer\(\)](#).

LCD_DrawPropTypeDef DrawProp[MAX_LAYER_NUMBER] [static]

Definition at line 124 of file [stm32746g_discovery_lcd.c](#).

DMA2D_HandleTypeDef hDma2dHandler [static]

Definition at line 120 of file [stm32746g_discovery_lcd.c](#).

Referenced by [LL_ConvertLineToARGB8888\(\)](#), and [LL_FillBuffer\(\)](#).

LTDC_HandleTypeDef hLtdcHandler

Definition at line 119 of file [stm32746g_discovery_lcd.c](#).

Referenced by [BSP_LCD_Clear\(\)](#), [BSP_LCD_DeInit\(\)](#),
[BSP_LCD_DisplayOff\(\)](#), [BSP_LCD_DisplayOn\(\)](#),
[BSP_LCD_DrawBitmap\(\)](#), [BSP_LCD_DrawHLine\(\)](#),
[BSP_LCD_DrawPixel\(\)](#), [BSP_LCD_DrawVLine\(\)](#),
[BSP_LCD_FillRect\(\)](#), [BSP_LCD_GetXSize\(\)](#),
[BSP_LCD_GetYSize\(\)](#), [BSP_LCD_Init\(\)](#),
[BSP_LCD_LayerDefaultInit\(\)](#), [BSP_LCD_LayerRgb565Init\(\)](#),
[BSP_LCD_ReadPixel\(\)](#), [BSP_LCD_Reload\(\)](#),
[BSP_LCD_ResetColorKeying\(\)](#),
[BSP_LCD_ResetColorKeying_NoReload\(\)](#),
[BSP_LCD_SetColorKeying\(\)](#),
[BSP_LCD_SetColorKeying_NoReload\(\)](#),
[BSP_LCD_SetLayerAddress\(\)](#),
[BSP_LCD_SetLayerAddress_NoReload\(\)](#),
[BSP_LCD_SetLayerVisible\(\)](#),
[BSP_LCD_SetLayerVisible_NoReload\(\)](#),
[BSP_LCD_SetLayerWindow\(\)](#),
[BSP_LCD_SetLayerWindow_NoReload\(\)](#),
[BSP_LCD_SetTransparency\(\)](#),
[BSP_LCD_SetTransparency_NoReload\(\)](#), [BSP_LCD_SetXSize\(\)](#),
[BSP_LCD_SetYSize\(\)](#), and [LL_FillBuffer\(\)](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Variables
STM32746G_DISCOVERY AUDIO Private Variables				
STM32746G_DISCOVERY AUDIO				

Variables

```
AUDIO_DrvTypeDef * audio_drv  
SAI_HandleTypeDef haudio_out_sai = {0}  
SAI_HandleTypeDef haudio_in_sai = {0}  
TIM_HandleTypeDef haudio_tim  
    uint16_t __IO AudioInVolume =  
        DEFAULT_AUDIO_IN_VOLUME
```

Variable Documentation

AUDIO_DrvTypeDef* audio_drv

Definition at line [136](#) of file `stm32746g_discovery_audio.c`.

Referenced by `BSP_AUDIO_IN_InitEx()`,
`BSP_AUDIO_IN_OUT_Init()`, `BSP_AUDIO_IN_SetVolume()`,
`BSP_AUDIO_IN_Stop()`, `BSP_AUDIO_OUT_Init()`,
`BSP_AUDIO_OUT_Pause()`, `BSP_AUDIO_OUT_Play()`,
`BSP_AUDIO_OUT_Resume()`, `BSP_AUDIO_OUT_SetMute()`,
`BSP_AUDIO_OUT_SetOutputMode()`,
`BSP_AUDIO_OUT_SetVolume()`, and `BSP_AUDIO_OUT_Stop()`.

uint16_t __IO AudioInVolume = DEFAULT_AUDIO_IN_VOLUME

Definition at line [141](#) of file `stm32746g_discovery_audio.c`.

Referenced by `BSP_AUDIO_IN_SetVolume()`.

SAI_HandleTypeDef haudio_in_sai = {0}

Definition at line [138](#) of file `stm32746g_discovery_audio.c`.

Referenced by `BSP_AUDIO_IN_Delnit()`, `BSP_AUDIO_IN_InitEx()`,
`BSP_AUDIO_IN_OUT_Init()`, `BSP_AUDIO_IN_Pause()`,
`BSP_AUDIO_IN_Record()`, `BSP_AUDIO_IN_Resume()`,
`BSP_AUDIO_IN_Stop()`, `HAL_SAI_ErrorCallback()`,
`SAIx_In_Delnit()`, and `SAIx_In_Init()`.

SAI_HandleTypeDef haudio_out_sai = {0}

Definition at line [137](#) of file `stm32746g_discovery_audio.c`.

Referenced by [BSP_AUDIO_IN_OUT_Init\(\)](#),
[BSP_AUDIO_OUT_ChangeBuffer\(\)](#), [BSP_AUDIO_OUT_DeInit\(\)](#),
[BSP_AUDIO_OUT_Init\(\)](#), [BSP_AUDIO_OUT_Pause\(\)](#),
[BSP_AUDIO_OUT_Play\(\)](#), [BSP_AUDIO_OUT_Resume\(\)](#),
[BSP_AUDIO_OUT_SetAudioFrameSlot\(\)](#),
[BSP_AUDIO_OUT_SetFrequency\(\)](#), [BSP_AUDIO_OUT_Stop\(\)](#),
[HAL_SAI_ErrorCallback\(\)](#), [SAIx_In_Init\(\)](#), [SAIx_Out_DeInit\(\)](#), and
[SAIx_Out_Init\(\)](#).

TIM_HandleTypeDefDef [haudio_tim](#)

Definition at line [139](#) of file [stm32746g_discovery_audio.c](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Defines				STM32746G_DISCOVERY_AUDIO Exported Constants
STM32746G_DISCOVERY AUDIO				

Defines

```
#define CODEC_AUDIOFRAME_SLOT_0123 SAI_SLOTACTIVE_0 |  
SAI_SLOTACTIVE_2 | SAI_SLOTACTIVE_3  
#define CODEC_AUDIOFRAME_SLOT_02 SAI_SLOTACTIVE_0 | SAI_SLOTACTIVE_1  
#define CODEC_AUDIOFRAME_SLOT_13 SAI_SLOTACTIVE_1 | SAI_SLOTACTIVE_2  
#define AUDIO_OUT_SAIx SAI2_Block_A  
#define AUDIO_OUT_SAIx_CLK_ENABLE() __HAL_RCC_SAI2_CLK_ENABLE()  
#define AUDIO_OUT_SAIx_CLK_DISABLE() __HAL_RCC_SAI2_CLK_DISABLE()  
#define AUDIO_OUT_SAIx_SCK_AF GPIO_AF10_SAI2  
#define AUDIO_OUT_SAIx_FS_SD_MCLK_AF GPIO_AF10_SAI2  
#define AUDIO_OUT_SAIx_MCLK_ENABLE() __HAL_RCC_GPIOI_ENABLE()  
#define AUDIO_OUT_SAIx_MCLK_GPIO_PORT GPIOI  
#define AUDIO_OUT_SAIx_MCLK_PIN GPIO_PIN_4  
#define AUDIO_OUT_SAIx_SCK_SD_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()  
#define AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT GPIOI  
#define AUDIO_OUT_SAIx_SCK_PIN GPIO_PIN_5  
#define AUDIO_OUT_SAIx_SD_PIN GPIO_PIN_6  
#define AUDIO_OUT_SAIx_FS_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()  
#define AUDIO_OUT_SAIx_FS_GPIO_PORT GPIOI  
#define AUDIO_OUT_SAIx_FS_PIN GPIO_PIN_7  
#define AUDIO_OUT_SAIx_DMAx_CLK_ENABLE() __HAL_RCC_DMA2_CLK_ENABLE()  
#define AUDIO_OUT_SAIx_DMAx_STREAM DMA2_Stream4  
#define AUDIO_OUT_SAIx_DMAx_CHANNEL DMA_CHANNEL_3  
#define AUDIO_OUT_SAIx_DMAx_IRQ DMA2_Stream4_IRQn  
#define AUDIO_OUT_SAIx_DMAx_PERIPH_DATA_SIZE DMA_PDATA_SIZE  
#define AUDIO_OUT_SAIx_DMAx_MEM_DATA_SIZE DMA_MDATA_SIZE  
#define DMA_MAX_SZE ((uint16_t)0xFFFF)  
#define AUDIO_OUT_SAIx_DMAx_IRQHandler DMA2_Stream4_IRQHandler  
#define AUDIO_OUT_IRQ_PREPRIO ((uint32_t)0x0E) /* Select the priority level(0 is the highest) */  
#define AUDIO_IN_SAIx SAI2_Block_B  
#define AUDIO_IN_SAIx_CLK_ENABLE() __HAL_RCC_SAI2_CLK_ENABLE()
```

```
#define AUDIO_IN_SAIx_CLK_DISABLE() __HAL_RCC_SAI2_CLK_DISABLE()
#define AUDIO_IN_SAIx_SD_AF GPIO_AF10_SAI2
#define AUDIO_IN_SAIx_SD_ENABLE() __HAL_RCC_GPIOG_CLK_ENABLE()
#define AUDIO_IN_SAIx_SD_GPIO_PORT GPIOG
#define AUDIO_IN_SAIx_SD_PIN GPIO_PIN_10
#define AUDIO_IN_INT_GPIO_ENABLE() __HAL_RCC_GPIOH_CLK_ENABLE()
#define AUDIO_IN_INT_GPIO_PORT GPIOH
#define AUDIO_IN_INT_GPIO_PIN GPIO_PIN_15
#define AUDIO_IN_INT_IRQ EXTI15_10_IRQHandler
#define AUDIO_IN_SAIx_DMAx_CLK_ENABLE() __HAL_RCC_DMA2_CLK_ENABLE()
#define AUDIO_IN_SAIx_DMAx_STREAM DMA2_Stream7
#define AUDIO_IN_SAIx_DMAx_CHANNEL DMA_CHANNEL_0
#define AUDIO_IN_SAIx_DMAx_IRQ DMA2_Stream7_IRQHandler
#define AUDIO_IN_SAIx_DMAx_PERIPH_DATA_SIZE DMA_PDATAALIGN
#define AUDIO_IN_SAIx_DMAx_MEM_DATA_SIZE DMA_MDATAALIGN
#define AUDIO_IN_SAIx_DMAx_IRQHandler DMA2_Stream7_IRQHandler
#define AUDIO_IN_INT_IRQHandler EXTI15_10_IRQHandler
#define AUDIO_IN_IRQ_PREPRIO ((uint32_t)0x0F) /* Select the preemption priority
is the highest) */
#define AUDIODATA_SIZE ((uint16_t)2) /* 16-bits audio data size */
#define AUDIO_OK ((uint8_t)0)
#define AUDIO_ERROR ((uint8_t)1)
#define AUDIO_TIMEOUT ((uint8_t)2)
#define DEFAULT_AUDIO_IN_FREQ I2S_AUDIOFREQ_16K
#define DEFAULT_AUDIO_IN_BIT_RESOLUTION ((uint8_t)16)
#define DEFAULT_AUDIO_IN_CHANNEL_NBR ((uint8_t)2) /* Mono */
#define DEFAULT_AUDIO_IN_VOLUME ((uint16_t)64)
#define CODEC_RESET_DELAY ((uint8_t)5)
#define OUTPUT_DEVICE_HEADPHONE1 OUTPUT_DEVICE_HEADPHONE1
#define OUTPUT_DEVICE_HEADPHONE2 OUTPUT_DEVICE_SPEAKER
/* connected to Speaker output of the wm8994 */
```

Define Documentation

#define AUDIO_ERROR ((uint8_t)1)

Definition at line [164](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_InitEx()`,
`BSP_AUDIO_IN_OUT_Init()`, `BSP_AUDIO_IN_Record()`,
`BSP_AUDIO_IN_SetVolume()`, `BSP_AUDIO_IN_Stop()`,
`BSP_AUDIO_OUT_Init()`, `BSP_AUDIO_OUT_Pause()`,
`BSP_AUDIO_OUT_Play()`, `BSP_AUDIO_OUT_Resume()`,
`BSP_AUDIO_OUT_SetMute()`,
`BSP_AUDIO_OUT_SetOutputMode()`,
`BSP_AUDIO_OUT_SetVolume()`, and `BSP_AUDIO_OUT_Stop()`.

#define AUDIO_IN_INT_GPIO_ENABLE() __HAL_RCC_GPIOH_C

Definition at line [137](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

#define AUDIO_IN_INT_GPIO_PIN GPIO_PIN_15

Definition at line [139](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

#define AUDIO_IN_INT_GPIO_PORT GPIOH

Definition at line [138](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_INT_IRQ  EXTI15_10_IRQn
```

Definition at line [140](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_INT_IRQHandler  EXTI15_10_IRQHandler
```

Definition at line [151](#) of file `stm32746g_discovery_audio.h`.

```
#define AUDIO_IN_IRQ_PREPRIO ((uint32_t)0x0F) /* Select the pre-
```

Definition at line [154](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_SAI2_Block_B
```

Definition at line [128](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_InitEx()`,
`BSP_AUDIO_IN_MspDelInit()`, `BSP_AUDIO_IN_MspInit()`,
`BSP_AUDIO_IN_OUT_Init()`, `SAIx_In_DelInit()`, and `SAIx_In_Init()`.

```
#define AUDIO_IN_SAIx_CLK_DISABLE( ) __HAL_RCC_SAI2_CLK
```

Definition at line [130](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspDelInit()`.

```
#define AUDIO_IN_SAIx_CLK_ENABLE( ) __HAL_RCC_SAI2_CLK
```

Definition at line [129](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_DMAx_CHANNEL DMA_CHANNEL_0
```

Definition at line [145](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_DMAx_CLK_ENABLE() __HAL_RCC_D
```

Definition at line [143](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_DMAx_IRQ DMA2_Stream7_IRQn
```

Definition at line [146](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspDeInit()`, and
`BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_DMAx_IRQHandler DMA2_Stream7_IRQHandler
```

Definition at line [150](#) of file `stm32746g_discovery_audio.h`.

```
#define AUDIO_IN_SAIx_DMAx_MEM_DATA_SIZE DMA_MDATAAI
```

Definition at line [148](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_DMAx_PERIPH_DATA_SIZE DMA_PDATA
```

Definition at line [147](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_DMAx_STREAM DMA2_Stream7
```

Definition at line [144](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_SD_AF GPIO_AF10_SAI2
```

Definition at line [131](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_SD_ENABLE() __HAL_RCC_GPIOG_CLK
```

Definition at line [133](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_SD_GPIO_PORT GPIOG
```

Definition at line [134](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspDelInit()`, and
`BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_IN_SAIx_SD_PIN GPIO_PIN_10
```

Definition at line [135](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_MspDelInit()`, and
`BSP_AUDIO_IN_MspInit()`.

```
#define AUDIO_OK ((uint8_t)0)
```

Definition at line [163](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_InitEx()`,
`BSP_AUDIO_IN_OUT_Init()`, `BSP_AUDIO_IN_Pause()`,
`BSP_AUDIO_IN_Record()`, `BSP_AUDIO_IN_Resume()`,
`BSP_AUDIO_IN_SetVolume()`, `BSP_AUDIO_IN_Stop()`,
`BSP_AUDIO_OUT_Init()`, `BSP_AUDIO_OUT_Pause()`,
`BSP_AUDIO_OUT_Play()`, `BSP_AUDIO_OUT_Resume()`,
`BSP_AUDIO_OUT_SetMute()`,
`BSP_AUDIO_OUT_SetOutputMode()`,
`BSP_AUDIO_OUT_SetVolume()`, and `BSP_AUDIO_OUT_Stop()`.

```
#define AUDIO_OUT_IRQ_PREPRIO ((uint32_t)0x0E) /* Select the
```

Definition at line [122](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_OUT_MspInit()`.

```
#define AUDIO_OUT_SAIx SAI2_Block_A
```

Definition at line [93](#) of file `stm32746g_discovery_audio.h`.

Referenced by `BSP_AUDIO_IN_OUT_Init()`,
`BSP_AUDIO_OUT_Init()`, `BSP_AUDIO_OUT_MspDelInit()`,
`BSP_AUDIO_OUT_MspInit()`, `SAIx_In_Init()`, `SAIx_Out_DelInit()`,

and [SAIx_Out_Init\(\)](#).

```
#define AUDIO_OUT_SAIx_CLK_DISABLE() __HAL_RCC_SAI2_
```

Definition at line [95](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_CLK_ENABLE() __HAL_RCC_SAI2_
```

Definition at line [94](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_DMAx_CHANNEL DMA_CHANNEL_3
```

Definition at line [113](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_DMAx_CLK_ENABLE() __HAL_RCC_
```

Definition at line [111](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_DMAx_IRQ DMA2_Stream4_IRQn
```

Definition at line [114](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_OUT_MspDeInit\(\)](#), and
[BSP_AUDIO_OUT_MspInit\(\)](#).

```
#define AUDIO_OUT_SAIx_DMAx_IRQHandler DMA2_Stream4_IRQHandler
```

Definition at line 119 of file [stm32746g_discovery_audio.h](#).

```
#define AUDIO_OUT_SAIx_DMAx_MEM_DATA_SIZE DMA_MDATA
```

Definition at line 116 of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_DMAx_PERIPH_DATA_SIZE DMA_PDATA
```

Definition at line 115 of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_DMAx_STREAM DMA2_Stream4
```

Definition at line 112 of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_FS_ENABLE( ) __HAL_RCC_GPIOI_C
```

Definition at line 106 of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_FS_GPIO_PORT GPIOI
```

Definition at line 107 of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_OUT_MspDeInit\(\)](#), and [BSP_AUDIO_OUT_MspInit\(\)](#).

```
#define AUDIO_OUT_SAIx_FS_PIN GPIO_PIN_7
```

Definition at line [108](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_OUT_MspDeInit\(\)](#), and [BSP_AUDIO_OUT_MspInit\(\)](#).

```
#define AUDIO_OUT_SAIx_FS_SD_MCLK_AF GPIO_AF10_SAI2
```

Definition at line [97](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_MCLK_ENABLE() __HAL_RCC_GPIC
```

Definition at line [99](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_MCLK_GPIO_PORT GPIOI
```

Definition at line [100](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_OUT_MspDeInit\(\)](#), and [BSP_AUDIO_OUT_MspInit\(\)](#).

```
#define AUDIO_OUT_SAIx_MCLK_PIN GPIO_PIN_4
```

Definition at line [101](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_OUT_MspDeInit\(\)](#), and [BSP_AUDIO_OUT_MsplInit\(\)](#).

```
#define AUDIO_OUT_SAIx_SCK_AF GPIO_AF10_SAI2
```

Definition at line [96](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_SCK_PIN GPIO_PIN_5
```

Definition at line [104](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_OUT_MspDeInit\(\)](#), and [BSP_AUDIO_OUT_MsplInit\(\)](#).

```
#define AUDIO_OUT_SAIx_SCK_SD_ENABLE() __HAL_RCC_GF
```

Definition at line [102](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT GPIOI
```

Definition at line [103](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_OUT_MspDeInit\(\)](#), and [BSP_AUDIO_OUT_MsplInit\(\)](#).

```
#define AUDIO_OUT_SAIx_SD_PIN GPIO_PIN_6
```

Definition at line [105](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_OUT_MspDeInit\(\)](#), and [BSP_AUDIO_OUT_MsplInit\(\)](#).

```
#define AUDIO_TIMEOUT ((uint8_t)2)
```

Definition at line [165](#) of file [stm32746g_discovery_audio.h](#).

```
#define AUDIODATA_SIZE ((uint16_t)2) /* 16-bits audio data size */
```

Definition at line [160](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define CODEC_AUDIOFRAME_SLOT_0123 SAI_SLOTACTIVE_0 |
```

Definition at line [86](#) of file [stm32746g_discovery_audio.h](#).

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

```
#define CODEC_AUDIOFRAME_SLOT_02 SAI_SLOTACTIVE_0 | S.
```

Definition at line [88](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_IN_InitEx\(\)](#), and [BSP_AUDIO_IN_OUT_Init\(\)](#).

```
#define CODEC_AUDIOFRAME_SLOT_13 SAI_SLOTACTIVE_1 | S.
```

Definition at line [90](#) of file [stm32746g_discovery_audio.h](#).

Referenced by [BSP_AUDIO_IN_InitEx\(\)](#), and [BSP_AUDIO_IN_OUT_Init\(\)](#).

```
#define CODEC_RESET_DELAY ((uint8_t)5)
```

Definition at line 178 of file `stm32746g_discovery_audio.h`.

```
#define DEFAULT_AUDIO_IN_BIT_RESOLUTION ((uint8_t)16)
```

Definition at line 169 of file `stm32746g_discovery_audio.h`.

```
#define DEFAULT_AUDIO_IN_CHANNEL_NBR ((uint8_t)2) /* Mono
```

Definition at line 170 of file `stm32746g_discovery_audio.h`.

```
#define DEFAULT_AUDIO_IN_FREQ I2S_AUDIOFREQ_16K
```

Definition at line 168 of file `stm32746g_discovery_audio.h`.

```
#define DEFAULT_AUDIO_IN_VOLUME ((uint16_t)64)
```

Definition at line 171 of file `stm32746g_discovery_audio.h`.

```
#define DMA_MAX_SZE ((uint16_t)0xFFFF)
```

Definition at line 117 of file `stm32746g_discovery_audio.h`.

```
#define OUTPUT_DEVICE_HEADPHONE1 OUTPUT_DEVICE_HEA
```

Definition at line 185 of file `stm32746g_discovery_audio.h`.

```
#define OUTPUT_DEVICE_HEADPHONE2 OUTPUT_DEVICE_SPEAKER
```

Definition at line **186** of file **stm32746g_discovery_audio.h**.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Defines

STM32746G_DISCOVERY_LOW_LEVEL_COM

[STM32746G_DISCOVERY_LOW_LEVEL Exported Constants](#)

Defines

```
#define COMn ((uint8_t)1)
#define DISCOVERY_COM1 USART1
    Definition for COM port1, connected to USART1.

#define DISCOVERY_COM1_CLK_ENABLE() __HAL_RCC_USART1_ENABLE()
#define DISCOVERY_COM1_CLK_DISABLE() __HAL_RCC_USART1_DISABLE()
#define DISCOVERY_COM1_TX_PIN GPIO_PIN_9
#define DISCOVERY_COM1_TX_GPIO_PORT GPIOA
#define DISCOVERY_COM1_TX_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_ENABLE()
#define DISCOVERY_COM1_TX_GPIO_CLK_DISABLE() __HAL_RCC_GPIOA_DISABLE()
#define DISCOVERY_COM1_TX_AF GPIO_AF7_USART1
#define DISCOVERY_COM1_RX_PIN GPIO_PIN_7
#define DISCOVERY_COM1_RX_GPIO_PORT GPIOB
#define DISCOVERY_COM1_RX_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_ENABLE()
#define DISCOVERY_COM1_RX_GPIO_CLK_DISABLE() __HAL_RCC_GPIOB_DISABLE()
#define DISCOVERY_COM1_RX_AF GPIO_AF7_USART1
#define DISCOVERY_COM1_IRQn USART1_IRQn
#define DISCOVERY_COMx_CLK_ENABLE(__INDEX__) do { if((__INDEX__ > 0) && ( __INDEX__ <= 1)) DISCOVERY_COM1_CLK_ENABLE(); } while(0)
#define DISCOVERY_COMx_CLK_DISABLE(__INDEX__) (((__INDEX__ > 0) && ( __INDEX__ <= 1)) ? DISCOVERY_COM1_CLK_DISABLE() : 0)
#define DISCOVERY_COMx_TX_GPIO_CLK_ENABLE(__INDEX__)
#define DISCOVERY_COM1_TX_GPIO_CLK_ENABLE(); } while(0)
#define DISCOVERY_COMx_TX_GPIO_CLK_DISABLE(__INDEX__)
#define DISCOVERY_COM1_TX_GPIO_CLK_DISABLE(); } while(0)
#define DISCOVERY_COMx_RX_GPIO_CLK_ENABLE(__INDEX__)
#define DISCOVERY_COM1_RX_GPIO_CLK_ENABLE(); } while(0)
#define DISCOVERY_COMx_RX_GPIO_CLK_DISABLE(__INDEX__)
#define DISCOVERY_COM1_RX_GPIO_CLK_DISABLE(); } while(0)
#define LCD_I2C_ADDRESS ((uint16_t)0x70)
#define CAMERA_I2C_ADDRESS ((uint16_t)0x60)
#define AUDIO_I2C_ADDRESS ((uint16_t)0x34)
#define EEPROM_I2C_ADDRESS_A01 ((uint16_t)0xA0)
```

```
#define EEPROM_I2C_ADDRESS_A02 ((uint16_t)0xA6)
#define TS_I2C_ADDRESS ((uint16_t)0x70)
#define DISCOVERY_AUDIO_I2Cx I2C3
#define DISCOVERY_AUDIO_I2Cx_CLK_ENABLE() __HAL_RCC_I2C3_CLK_ENABLE()
#define DISCOVERY_AUDIO_I2Cx_DMAX_CLK_ENABLE() __HAL_RCC_I2C3_CLK_ENABLE()
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()
#define DISCOVERY_AUDIO_I2Cx_FORCE_RESET() __HAL_RCC_I2C3_FORCE_RESET()
#define DISCOVERY_AUDIO_I2Cx_RELEASE_RESET() __HAL_RCC_I2C3_RELEASE_RESET()
#define DISCOVERY_AUDIO_I2Cx_SCL_PIN GPIO_PIN_7
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PORT GPIOH
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_AF GPIO_AF4_I2C3
#define DISCOVERY_AUDIO_I2Cx_SDA_PIN GPIO_PIN_8
#define DISCOVERY_AUDIO_I2Cx_EV IRQn I2C3_EV_IRQn
#define DISCOVERY_AUDIO_I2Cx_ER IRQn I2C3_ER_IRQn
#define DISCOVERY_EXT_I2Cx I2C1
#define DISCOVERY_EXT_I2Cx_CLK_ENABLE() __HAL_RCC_I2C1_CLK_ENABLE()
#define DISCOVERY_EXT_DMAX_CLK_ENABLE() __HAL_RCC_I2C1_CLK_ENABLE()
#define DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()
#define DISCOVERY_EXT_I2Cx_FORCE_RESET() __HAL_RCC_I2C1_FORCE_RESET()
#define DISCOVERY_EXT_I2Cx_RELEASE_RESET() __HAL_RCC_I2C1_RELEASE_RESET()
#define DISCOVERY_EXT_I2Cx_SCL_PIN GPIO_PIN_8
#define DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_PORT GPIOB
#define DISCOVERY_EXT_I2Cx_SCL_SDA_AF GPIO_AF4_I2C1
#define DISCOVERY_EXT_I2Cx_SDA_PIN GPIO_PIN_9
#define DISCOVERY_EXT_I2Cx_EV IRQn I2C1_EV_IRQn
#define DISCOVERY_EXT_I2Cx_ER IRQn I2C1_ER_IRQn
```

Define Documentation

#define AUDIO_I2C_ADDRESS ((uint16_t)0x34)

Definition at line [228](#) of file `stm32746g_discovery.h`.

Referenced by `BSP_AUDIO_IN_InitEx()`,
`BSP_AUDIO_IN_OUT_Init()`, `BSP_AUDIO_IN_SetVolume()`,
`BSP_AUDIO_IN_Stop()`, `BSP_AUDIO_OUT_Init()`,
`BSP_AUDIO_OUT_Pause()`, `BSP_AUDIO_OUT_Play()`,
`BSP_AUDIO_OUT_Resume()`, `BSP_AUDIO_OUT_SetMute()`,
`BSP_AUDIO_OUT_SetOutputMode()`,
`BSP_AUDIO_OUT_SetVolume()`, and `BSP_AUDIO_OUT_Stop()`.

#define CAMERA_I2C_ADDRESS ((uint16_t)0x60)

Definition at line [227](#) of file `stm32746g_discovery.h`.

Referenced by `BSP_CAMERA_Init()`.

#define COMn ((uint8_t)1)

Definition at line [192](#) of file `stm32746g_discovery.h`.

#define DISCOVERY_AUDIO_DMAt_CLK_ENABLE() __HAL_RC

Definition at line [246](#) of file `stm32746g_discovery.h`.

#define DISCOVERY_AUDIO_I2Cx I2C3

Definition at line [244](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_Init\(\)](#).

```
#define DISCOVERY_AUDIO_I2Cx_CLK_ENABLE() __HAL_RCC
```

Definition at line [245](#) of file [stm32746g_discovery.h](#).

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_AUDIO_I2Cx_ER IRQn I2C3_ER IRQn
```

Definition at line [260](#) of file [stm32746g_discovery.h](#).

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_AUDIO_I2Cx_EV IRQn I2C3_EV IRQn
```

Definition at line [259](#) of file [stm32746g_discovery.h](#).

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_AUDIO_I2Cx_FORCE_RESET() __HAL_RC
```

Definition at line [249](#) of file [stm32746g_discovery.h](#).

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_AUDIO_I2Cx_RELEASE_RESET() __HAL_F
```

Definition at line [250](#) of file [stm32746g_discovery.h](#).

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_AUDIO_I2Cx_SCL_PIN GPIO_PIN_7
```

Definition at line [253](#) of file `stm32746g_discovery.h`.

Referenced by `I2Cx_MspInit()`.

```
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_AF GPIO_AF4_I2C3
```

Definition at line [255](#) of file `stm32746g_discovery.h`.

Referenced by `I2Cx_MspInit()`.

```
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_CLK_ENABLE
```

Definition at line [247](#) of file `stm32746g_discovery.h`.

Referenced by `I2Cx_MspInit()`.

```
#define DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PORT GPIOH
```

Definition at line [254](#) of file `stm32746g_discovery.h`.

Referenced by `I2Cx_MspInit()`.

```
#define DISCOVERY_AUDIO_I2Cx_SDA_PIN GPIO_PIN_8
```

Definition at line [256](#) of file `stm32746g_discovery.h`.

Referenced by `I2Cx_MspInit()`.

```
#define DISCOVERY_COM1 USART1
```

Definition for COM port1, connected to USART1.

Definition at line [197](#) of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_CLK_DISABLE() __HAL_RCC_USART1
```

Definition at line [199](#) of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_CLK_ENABLE() __HAL_RCC_USART1
```

Definition at line [198](#) of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_IRQn USART1_IRQn
```

Definition at line [213](#) of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_RX_AF GPIO_AF7_USART1
```

Definition at line [211](#) of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_RX_GPIO_CLK_DISABLE() __HAL
```

Definition at line [210](#) of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_RX_GPIO_CLK_ENABLE() __HAL
```

Definition at line [209](#) of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_RX_GPIO_PORT GPIOB
```

Definition at line 208 of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_RX_PIN GPIO_PIN_7
```

Definition at line 207 of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_TX_AF GPIO_AF7_USART1
```

Definition at line 205 of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_TX_GPIO_CLK_DISABLE() __HAL_
```

Definition at line 204 of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_TX_GPIO_CLK_ENABLE() __HAL_F
```

Definition at line 203 of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_TX_GPIO_PORT GPIOA
```

Definition at line 202 of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COM1_TX_PIN GPIO_PIN_9
```

Definition at line 201 of file `stm32746g_discovery.h`.

```
#define DISCOVERY_COMx_CLK_DISABLE(__INDEX__) (((__IN
```

Definition at line 216 of file `stm32746g_discovery.h`.

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

```
#define DISCOVERY_COMx_CLK_ENABLE( __INDEX__ ) do { if(
```

Definition at line [215](#) of file [stm32746g_discovery.h](#).

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

```
#define DISCOVERY_COMx_RX_GPIO_CLK_DISABLE( __INDEX__
```

Definition at line [222](#) of file [stm32746g_discovery.h](#).

```
#define DISCOVERY_COMx_RX_GPIO_CLK_ENABLE( __INDEX__
```

Definition at line [221](#) of file [stm32746g_discovery.h](#).

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

```
#define DISCOVERY_COMx_TX_GPIO_CLK_DISABLE( __INDEX__
```

Definition at line [219](#) of file [stm32746g_discovery.h](#).

```
#define DISCOVERY_COMx_TX_GPIO_CLK_ENABLE( __INDEX__
```

Definition at line [218](#) of file [stm32746g_discovery.h](#).

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

```
#define DISCOVERY_EXT_DMAx_CLK_ENABLE( ) __HAL_RCC_
```

Definition at line [265](#) of file [stm32746g_discovery.h](#).

```
#define DISCOVERY_EXT_I2Cx_I2C1
```

Definition at line [263](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_Init\(\)](#).

```
#define DISCOVERY_EXT_I2Cx_CLK_ENABLE() __HAL_RCC_I2
```

Definition at line [264](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_EXT_I2Cx_ER IRQn I2C1_ER IRQn
```

Definition at line [279](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_EXT_I2Cx_EV IRQn I2C1_EV IRQn
```

Definition at line [278](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_EXT_I2Cx_FORCE_RESET() __HAL_RCC_I
```

Definition at line [268](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

```
#define DISCOVERY_EXT_I2Cx_RELEASE_RESET() __HAL_RCC_I
```

Definition at line [269](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

#define DISCOVERY_EXT_I2Cx_SCL_PIN GPIO_PIN_8

Definition at line [272](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

#define DISCOVERY_EXT_I2Cx_SCL_SDA_AF GPIO_AF4_I2C1

Definition at line [274](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

#define DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_CLK_ENABLE()

Definition at line [266](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

#define DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_PORT GPIOB

Definition at line [273](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

#define DISCOVERY_EXT_I2Cx_SDA_PIN GPIO_PIN_9

Definition at line [275](#) of file `stm32746g_discovery.h`.

Referenced by [I2Cx_MspInit\(\)](#).

#define EEPROM_I2C_ADDRESS_A01 ((uint16_t)0xA0)

Definition at line [229](#) of file [stm32746g_discovery.h](#).

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

#define EEPROM_I2C_ADDRESS_A02 ((uint16_t)0xA6)

Definition at line [230](#) of file [stm32746g_discovery.h](#).

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

#define LCD_I2C_ADDRESS ((uint16_t)0x70)

Definition at line [226](#) of file [stm32746g_discovery.h](#).

#define TS_I2C_ADDRESS ((uint16_t)0x70)

Definition at line [231](#) of file [stm32746g_discovery.h](#).

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

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
STM32746G_DISCOVERY_LOW_LEVELSTM3274				Functions
Exported Functions				
STM32746G_DISCOVERY_LOW_LEVEL				

Functions

uint32_t	BSP_GetVersion (void)	This method returns the STM32746G DISCOVERY BSP Driver revision.
void	BSP_LED_Init (Led_TypeDef Led)	Configures LED on 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.
void	BSP_COM_Init (COM_TypeDef COM, UART_HandleTypeDef *huart)	Configures COM port.
void	BSP_COM_DeInit (COM_TypeDef COM, UART_HandleTypeDef *huart)	DeInit COM port.
static void	I2Cx_MspInit (I2C_HandleTypeDef *i2c_handler)	Initializes I2C MSP.

static void	I2Cx_Init (I2C_HandleTypeDef *i2c_handler) Initializes I2C HAL.
static HAL_StatusTypeDef	I2Cx_ReadMultiple (I2C_HandleTypeDef *i2c_handler, uint8_t Addr, uint16_t Reg, uint16_t MemAddress, uint8_t *Buffer, uint16_t Length) Reads multiple data.
static HAL_StatusTypeDef	I2Cx_WriteMultiple (I2C_HandleTypeDef *i2c_handler, uint8_t Addr, uint16_t Reg, uint16_t MemAddress, uint8_t *Buffer, uint16_t Length) Writes a value in a register of the device through BUS in using DMA mode.
static HAL_StatusTypeDef	I2Cx_IsDeviceReady (I2C_HandleTypeDef *i2c_handler, uint16_t DevAddress, uint32_t Trials) Checks if target device is ready for communication.
static void	I2Cx_Error (I2C_HandleTypeDef *i2c_handler, uint8_t Addr) Manages error callback by re-initializing I2C.
void	AUDIO_IO_Init (void) Initializes Audio low level.
void	AUDIO_IO_DeInit (void) Deinitializes Audio low level.
void	AUDIO_IO_Write (uint8_t Addr, uint16_t Reg, uint16_t Value) Writes a single data.
uint16_t	AUDIO_IO_Read (uint8_t Addr, uint16_t Reg) Reads a single data.
	AUDIO_IO_Delay (uint32_t Delay)

		AUDIO Codec delay.
	void	CAMERA_IO_Init (void) Initializes Camera low level.
	void	CAMERA_IO_Write (uint8_t Addr, uint8_t Reg, uint8_t Value) Camera writes single data.
	uint8_t	CAMERA_IO_Read (uint8_t Addr, uint8_t Reg) Camera reads single data.
	void	CAMERA_Delay (uint32_t Delay) Camera delay.
	void	EEPROM_IO_Init (void) Initializes peripherals used by the I2C EEPROM driver.
HAL_StatusTypeDef		EEPROM_IO_WriteData (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize) Write data to I2C EEPROM driver in using DMA channel.
HAL_StatusTypeDef		EEPROM_IO_ReadData (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize) Read data from I2C EEPROM driver in using DMA channel.
HAL_StatusTypeDef		EEPROM_IO_IsDeviceReady (uint16_t DevAddress, uint32_t Trials) Checks if target device is ready for communication.
	void	TS_IO_Init (void) Initializes Touchscreen low level.
	void	TS_IO_Write (uint8_t Addr, uint8_t Reg, uint8_t Value) Writes a single data.
	uint8_t	TS_IO_Read (uint8_t Addr, uint8_t Reg)

Reads a single data.

void **TS_IO_Delay** (uint32_t Delay)
TS delay.

Function Documentation

void [AUDIO_IO_DelInit](#)(void)

Deinitializes Audio low level.

Return values:

None

Definition at line [683](#) of file [stm32746g_discovery.c](#).

void [AUDIO_IO_Delay](#)(uint32_t Delay)

AUDIO Codec delay.

Parameters:

Delay,: Delay in ms

Return values:

None

Definition at line [731](#) of file [stm32746g_discovery.c](#).

void [AUDIO_IO_Init](#)(void)

Initializes Audio low level.

Return values:

None

Definition at line [674](#) of file [stm32746g_discovery.c](#).

References [hI2cAudioHandler](#), and [I2Cx_Init\(\)](#).

```
uint16_t AUDIO_IO_Read( uint8_t Addr,  
                        uint16_t Reg  
                      )
```

Reads a single data.

Parameters:

Addr,: I2C address
Reg,: Reg address

Return values:

Data to be read

Definition at line [711](#) of file [stm32746g_discovery.c](#).

References [hI2cAudioHandler](#), and [I2Cx_ReadMultiple\(\)](#).

```
void AUDIO_IO_Write( uint8_t Addr,  
                      uint16_t Reg,  
                      uint16_t Value  
                    )
```

Writes a single data.

Parameters:

Addr,: I2C address
Reg,: Reg address
Value,: Data to be written

Return values:

None

Definition at line [694](#) of file [stm32746g_discovery.c](#).

References [hi2cAudioHandler](#), and [I2Cx_WriteMultiple\(\)](#).

```
void BSP_COM_DeInit ( COM_TypeDef           COM,
                      UART_HandleTypeDef * huart
                    )
```

DeInit COM port.

Parameters:

COM,: COM port to be configured. This parameter can be one of the following values:

- COM1
- COM2

huart,: Pointer to a `UART_HandleTypeDef` structure that contains the configuration information for the specified USART peripheral.

Return values:

None

Definition at line [435](#) of file [stm32746g_discovery.c](#).

References [COM_USART](#), and
[DISCOVERY_COMx_CLK_DISABLE](#).

```
void BSP_COM_Init ( COM_TypeDef           COM,
                     UART_HandleTypeDef * huart
                   )
```

Configures COM port.

Parameters:

COM,: COM port to be configured. This parameter can be one of the following values:

- COM1
- COM2

huart,: Pointer to a UART_HandleTypeDef structure that contains the configuration information for the specified USART peripheral.

Return values:

None

Definition at line [395](#) of file [stm32746g_discovery.c](#).

References [COM_RX_AF](#), [COM_RX_PIN](#), [COM_RX_PORT](#), [COM_TX_AF](#), [COM_TX_PIN](#), [COM_TX_PORT](#), [COM_USART](#), [DISCOVERY_COMx_CLK_ENABLE](#), [DISCOVERY_COMx_RX_GPIO_CLK_ENABLE](#), and [DISCOVERY_COMx_TX_GPIO_CLK_ENABLE](#).

`uint32_t BSP_GetVersion(void)`

This method returns the STM32746G DISCOVERY BSP Driver revision.

Return values:

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

Definition at line [173](#) of file [stm32746g_discovery.c](#).

References [__STM32746G_DISCO_BSP_VERSION](#).

`void BSP_LED_DeInit(Led_TypeDef Led)`

DeInit LEDs.

Parameters:

Led,: LED to be configured. This parameter can be one of the

following values:

- LED1

Note:

Led DeInit does not disable the GPIO clock

Return values:

None

Definition at line [217](#) of file [stm32746g_discovery.c](#).

References [GPIO_PIN](#), [LED1](#), and [LED1_GPIO_PORT](#).

void [BSP_LED_Init](#)(Led_TypeDef Led)

Configures LED on GPIO.

Parameters:

Led,: LED to be configured. This parameter can be one of the following values:

- LED1

Return values:

None

Definition at line [185](#) of file [stm32746g_discovery.c](#).

References [GPIO_PIN](#), [LED1](#), [LED1_GPIO_CLK_ENABLE](#), and [LED1_GPIO_PORT](#).

void [BSP_LED_Off](#)(Led_TypeDef Led)

Turns selected LED Off.

Parameters:

Led,: LED to be set off This parameter can be one of the following values:

- LED1

Return values:

None

Definition at line [258](#) of file [stm32746g_discovery.c](#).

References [GPIO_PIN](#), [LED1](#), and [LED1_GPIO_PORT](#).

void BSP_LED_On (Led_TypeDef Led)

Turns selected LED On.

Parameters:

Led,: LED to be set on This parameter can be one of the following values:

- LED1

Return values:

None

Definition at line [240](#) of file [stm32746g_discovery.c](#).

References [GPIO_PIN](#), [LED1](#), and [LED1_GPIO_PORT](#).

void BSP_LED_Toggle (Led_TypeDef Led)

Toggles the selected LED.

Parameters:

Led,: LED to be toggled This parameter can be one of the following values:

- LED1

Return values:

None

Definition at line 276 of file [stm32746g_discovery.c](#).

References [GPIO_PIN](#), [LED1](#), and [LED1_GPIO_PORT](#).

void BSP_PB_DelInit (Button_TypeDef Button)

Push Button DelInit.

Parameters:

Button,: Button to be configured This parameter can be one of the following values:

- **BUTTON_WAKEUP**: Wakeup Push Button
- **BUTTON_TAMPER**: Tamper Push Button
- **BUTTON_KEY**: Key Push Button

Note:

On STM32746G-Discovery board, the three buttons (Wakeup, Tamper and key buttons) are mapped on the same push button named "User" on the board serigraphy.

PB DelInit does not disable the GPIO clock

Return values:

None

Definition at line 358 of file [stm32746g_discovery.c](#).

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

uint32_t BSP_PB_GetState (Button_TypeDef Button)

Returns the selected button state.

Parameters:

Button,: Button to be checked This parameter can be one of the following values:

- BUTTON_WAKEUP: Wakeup Push Button
- BUTTON_TAMPER: Tamper Push Button
- BUTTON_KEY: Key Push Button

Note:

On STM32746G-Discovery board, the three buttons (Wakeup, Tamper and key buttons) are mapped on the same push button named "User" on the board serigraphy.

Return values:

The Button GPIO pin value

Definition at line 380 of file [stm32746g_discovery.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,: Button to be configured This parameter can be one of the following values:

- BUTTON_WAKEUP: Wakeup Push Button
- BUTTON_TAMPER: Tamper Push Button
- BUTTON_KEY: Key Push Button

ButtonMode,: Button mode This parameter can be one of the following values:

- BUTTON_MODE_GPIO: Button will be used as simple IO
- BUTTON_MODE_EXTI: Button will be

connected to EXTI line with interrupt generation capability

Note:

On STM32746G-Discovery board, the three buttons (Wakeup, Tamper and key buttons) are mapped on the same push button named "User" on the board serigraphy.

Return values:

None

Definition at line [304](#) of file [stm32746g_discovery.c](#).

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

void [CAMERA_Delay](#)(uint32_t Delay)

Camera delay.

Parameters:

Delay,: Delay in ms

Return values:

None

Definition at line [779](#) of file [stm32746g_discovery.c](#).

void [CAMERA_IO_Init](#)(void)

Initializes Camera low level.

Return values:

None

Definition at line 742 of file `stm32746g_discovery.c`.

References [hi2cExtHandler](#), and [I2Cx_Init\(\)](#).

```
uint8_t CAMERA_IO_Read ( uint8_t Addr,  
                         uint8_t Reg  
                       )
```

Camera reads single data.

Parameters:

Addr,: I2C address

Reg,: Register address

Return values:

Read data

Definition at line 765 of file `stm32746g_discovery.c`.

References [hi2cExtHandler](#), and [I2Cx_ReadMultiple\(\)](#).

```
void CAMERA_IO_Write ( uint8_t Addr,  
                       uint8_t Reg,  
                       uint8_t Value  
                     )
```

Camera writes single data.

Parameters:

Addr,: I2C address

Reg,: Register address

Value,: Data to be written

Return values:

None

Definition at line [754](#) of file [stm32746g_discovery.c](#).

References [hI2cExtHandler](#), and [I2Cx_WriteMultiple\(\)](#).

void EEPROM_IO_Init (void)

Initializes peripherals used by the I2C EEPROM driver.

Return values:

None

Definition at line [790](#) of file [stm32746g_discovery.c](#).

References [hI2cExtHandler](#), and [I2Cx_Init\(\)](#).

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

**HAL_StatusTypeDef EEPROM_IO_IsDeviceReady (uint16_t DevAd
 uint32_t Trials
)**

Checks if target device is ready for communication.

Note:

This function is used with Memory devices

Parameters:

DevAddress,: Target device address

Trials,: Number of trials

Return values:

HAL status

Definition at line 828 of file [stm32746g_discovery.c](#).

References [hI2cExtHandler](#), and [I2Cx_IsDeviceReady\(\)](#).

Referenced by [BSP_EEPROM_Init\(\)](#), and
[BSP_EEPROM_WaitEepromStandbyState\(\)](#).

```
HAL_StatusTypeDef EEPROM_IO_ReadData ( uint16_t DevAddress,  
                                      uint16_t MemAddress,  
                                      uint8_t * pBuffer,  
                                      uint32_t BufferSize  
                                    )
```

Read data from I2C EEPROM driver in using DMA channel.

Parameters:

DevAddress,: Target device address
MemAddress,: Internal memory address
pBuffer,: Pointer to data buffer
BufferSize,: Amount of data to be read

Return values:

HAL status

Definition at line 816 of file [stm32746g_discovery.c](#).

References [hI2cExtHandler](#), and [I2Cx_ReadMultiple\(\)](#).

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

```
HAL_StatusTypeDef EEPROM_IO_WriteData ( uint16_t DevAddress,  
                                       uint16_t MemAddress,  
                                       uint8_t * pBuffer,  
                                       uint32_t BufferSize  
                                     )
```

)

Write data to I2C EEPROM driver in using DMA channel.

Parameters:

DevAddress,: Target device address
MemAddress,: Internal memory address
pBuffer,: Pointer to data buffer
BufferSize,: Amount of data to be sent

Return values:

HAL status

Definition at line [803](#) of file [stm32746g_discovery.c](#).

References [hI2cExtHandler](#), and [I2Cx_WriteMultiple\(\)](#).

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

```
static void I2Cx_Error ( I2C_HandleTypeDef * i2c_handler,
                         uint8_t                               Addr
                         )                                         [static]
```

Manages error callback by re-initializing I2C.

Parameters:

i2c_handler : I2C handler
Addr,: I2C Address

Return values:

None

Definition at line [655](#) of file [stm32746g_discovery.c](#).

References [I2Cx_Init\(\)](#).

Referenced by [I2Cx_ReadMultiple\(\)](#), and [I2Cx_WriteMultiple\(\)](#).

static void I2Cx_Init(I2C_HandleTypeDef * i2c_handler) [static]

Initializes I2C HAL.

Parameters:

i2c_handler : I2C handler

Return values:

None

Definition at line [548](#) of file [stm32746g_discovery.c](#).

References [DISCOVERY_AUDIO_I2Cx](#), [DISCOVERY_EXT_I2Cx](#), [hI2cAudioHandler](#), and [I2Cx_MspInit\(\)](#).

Referenced by [AUDIO_IO_Init\(\)](#), [CAMERA_IO_Init\(\)](#), [EEPROM_IO_Init\(\)](#), [I2Cx_Error\(\)](#), and [TS_IO_Init\(\)](#).

static HAL_StatusTypeDef I2Cx_IsDeviceReady(I2C_HandleTypeDef * i2c_handler, uint16_t DevAddress, uint32_t Trials,)

Checks if target device is ready for communication.

Note:

This function is used with Memory devices

Parameters:

i2c_handler : I2C handler

DevAddress, Target device address

Trials,: Number of trials

Return values:

HAL status

Definition at line 644 of file [stm32746g_discovery.c](#).

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

static void I2Cx_MspInit (I2C_HandleTypeDef * i2c_handler) [stat:]

Initializes I2C MSP.

Parameters:

i2c_handler : I2C handler

Return values:

None

Definition at line 461 of file [stm32746g_discovery.c](#).

References [DISCOVERY_AUDIO_I2Cx_CLK_ENABLE](#),
[DISCOVERY_AUDIO_I2Cx_ER IRQn](#),
[DISCOVERY_AUDIO_I2Cx_EV IRQn](#),
[DISCOVERY_AUDIO_I2Cx_FORCE_RESET](#),
[DISCOVERY_AUDIO_I2Cx_RELEASE_RESET](#),
[DISCOVERY_AUDIO_I2Cx_SCL_PIN](#),
[DISCOVERY_AUDIO_I2Cx_SCL_SDA_AF](#),
[DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_CLK_ENABLE](#),
[DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PORT](#),
[DISCOVERY_AUDIO_I2Cx_SDA_PIN](#),
[DISCOVERY_EXT_I2Cx_CLK_ENABLE](#),
[DISCOVERY_EXT_I2Cx_ER IRQn](#),
[DISCOVERY_EXT_I2Cx_EV IRQn](#),
[DISCOVERY_EXT_I2Cx_FORCE_RESET](#),
[DISCOVERY_EXT_I2Cx_RELEASE_RESET](#),
[DISCOVERY_EXT_I2Cx_SCL_PIN](#),
[DISCOVERY_EXT_I2Cx_SCL_SDA_AF](#),

DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_CLK_ENABLE,
DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_PORT,
DISCOVERY_EXT_I2Cx_SDA_PIN, and **hI2cAudioHandler**.

Referenced by **I2Cx_Init()**.

```
static HAL_StatusTypeDef I2Cx_ReadMultiple ( I2C_HandleTypeDef
                                            uint8_t
                                            uint16_t
                                            uint16_t
                                            uint8_t *
                                            uint16_t
)
}
```

Reads multiple data.

Parameters:

i2c_handler	: I2C handler
Addr,:	I2C address
Reg,:	Reg address
MemAddress,:	Memory address
Buffer,:	Pointer to data buffer
Length,:	Length of the data

Return values:

Number of read data

Definition at line 586 of file **stm32746g_discovery.c**.

References **I2Cx_Error()**.

Referenced by **AUDIO_IO_Read()**, **CAMERA_IO_Read()**,
EEPROM_IO_ReadData(), and **TS_IO_Read()**.

```
static HAL_StatusTypeDef I2Cx_WriteMultiple ( I2C_HandleTypeDef
                                              uint8_t
                                              uint16_t
                                              uint16_t
                                              uint8_t *
                                              uint16_t
)

```

Writes a value in a register of the device through BUS in using DMA mode.

Parameters:

i2c_handler : I2C handler
Addr,: Device address on BUS Bus.
Reg,: The target register address to write
MemAddress,: Memory address
Buffer,: The target register value to be written
Length,: buffer size to be written

Return values:

HAL status

Definition at line [616](#) of file [stm32746g_discovery.c](#).

References [I2Cx_Error\(\)](#).

Referenced by [AUDIO_IO_Write\(\)](#), [CAMERA_IO_Write\(\)](#),
[EEPROM_IO_WriteData\(\)](#), and [TS_IO_Write\(\)](#).

```
void TS_IO_Delay ( uint32_t Delay )

```

TS delay.

Parameters:

Delay,: Delay in ms

Return values:

None

Definition at line 876 of file [stm32746g_discovery.c](#).

void TS_IO_Init(void)

Initializes Touchscreen low level.

Return values:

None

Definition at line 839 of file [stm32746g_discovery.c](#).

References [hI2cAudioHandler](#), and [I2Cx_Init\(\)](#).

```
uint8_t TS_IO_Read( uint8_t Addr,  
                     uint8_t Reg  
)
```

Reads a single data.

Parameters:

Addr,: I2C address

Reg,: Reg address

Return values:

Data to be read

Definition at line 862 of file [stm32746g_discovery.c](#).

References [hI2cAudioHandler](#), and [I2Cx_ReadMultiple\(\)](#).

```
void TS_IO_Write( uint8_t Addr,  
                  uint8_t Reg,  
                  uint8_t Value  
)
```

Writes a single data.

Parameters:

Addr,: I2C address

Reg,: Reg address

Value,: Data to be written

Return values:

None

Definition at line [851](#) of file [stm32746g_discovery.c](#).

References [hI2cAudioHandler](#), and [I2Cx_WriteMultiple\(\)](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories	Variables			
STM32746G_DISCOVERY_AUDIO Exported Variables				STM32746G_DISCOVERY AUDIO

Variables

`__IO uint16_t AudioInVolume`

Variable Documentation

`__IO uint16_t AudioInVolume`

Definition at line [141](#) of file `stm32746g_discovery_audio.c`.

Referenced by `BSP_AUDIO_IN_SetVolume()`.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Functions		
STM32746G_DISCOVERY_AUDIO Out Private Functions			
STM32746G_DISCOVERY AUDIO			

Functions

uint8_t	BSP_AUDIO_IN_Init (uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording.
uint8_t	BSP_AUDIO_IN_InitEx (uint16_t InputDevice, uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording.
uint8_t	BSP_AUDIO_IN_OUT_Init (uint16_t InputDevice, uint16_t OutputDevice, uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording and playback in parallel.
uint8_t	BSP_AUDIO_IN_Record (uint16_t *pbuf, uint32_t size)	Starts audio recording.
uint8_t	BSP_AUDIO_IN_Stop (uint32_t Option)	Stops audio recording.
uint8_t	BSP_AUDIO_IN_Pause (void)	Pauses the audio file stream.
uint8_t	BSP_AUDIO_IN_Resume (void)	Resumes the audio file stream.
uint8_t	BSP_AUDIO_IN_SetVolume (uint8_t Volume)	Controls the audio in volume level.
void	BSP_AUDIO_IN_DeInit (void)	Deinit the audio IN peripherals.
void	HAL_SAI_RxCpltCallback (SAI_HandleTypeDefDef *hsai)	Rx Transfer completed callbacks.
void	HAL_SAI_RxHalfCpltCallback (SAI_HandleTypeDefDef *hsai)	Rx Half Transfer completed callbacks.
__weak void	BSP_AUDIO_IN_TransferComplete_CallBack (void)	User callback when record buffer is filled.
__weak void	BSP_AUDIO_IN_HalfTransfer_CallBack (void)	Manages the DMA Half Transfer complete event.

<code>__weak void</code>	BSP_AUDIO_IN_Error_CallBack (void) Audio IN Error callback function.
<code>__weak void</code>	BSP_AUDIO_IN_MspInit (SAI_HandleTypeDef *hsai, void *Params) Initializes BSP_AUDIO_IN MSP.
<code>__weak void</code>	BSP_AUDIO_IN_MspDelInit (SAI_HandleTypeDef *hsai, void *Params) DeInitializes BSP_AUDIO_IN MSP.
<code>static void</code>	SAIx_In_Init (uint32_t SaiOutMode, uint32_t SlotActive, uint32_t AudioFreq) Initializes the input Audio Codec audio interface (SAI).
<code>static void</code>	SAIx_In_DelInit (void) Deinitializes the output Audio Codec audio interface (SAI).

Function Documentation

void [BSP_AUDIO_IN_DelInit](#)(void)

Deinit the audio IN peripherals.

Return values:

None

Definition at line [1056](#) of file [stm32746g_discovery_audio.c](#).

References [BSP_AUDIO_IN_MspDeInit\(\)](#), [haudio_in_sai](#), and [SAIx_In_DelInit\(\)](#).

__weak void [BSP_AUDIO_IN_Error_CallBack](#)(void)

Audio IN Error callback function.

Return values:

None

Definition at line [1112](#) of file [stm32746g_discovery_audio.c](#).

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

__weak void [BSP_AUDIO_IN_HalfTransfer_CallBack](#)(void)

Manages the DMA Half Transfer complete event.

Return values:

None

Definition at line [1101](#) of file [stm32746g_discovery_audio.c](#).

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

```
uint8_t BSP_AUDIO_IN_Init( uint32_t AudioFreq,  
                           uint32_t BitRes,  
                           uint32_t ChnlNbr  
                         )
```

Initializes wave recording.

Parameters:

- AudioFreq,:** Audio frequency to be configured for the SAI peripheral.
- BitRes,:** Audio frequency to be configured.
- ChnlNbr,:** Channel number.

Return values:

- AUDIO_OK** if correct communication, else wrong communication

Definition at line [788](#) of file [stm32746g_discovery_audio.c](#).

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

```
uint8_t BSP_AUDIO_IN_InitEx( uint16_t InputDevice,  
                            uint32_t AudioFreq,  
                            uint32_t BitRes,  
                            uint32_t ChnlNbr  
                          )
```

Initializes wave recording.

Parameters:

- InputDevice,:** INPUT_DEVICE_DIGITAL_MICROPHONE_2 or INPUT_DEVICE_INPUT_LINE_1

AudioFreq,: Audio frequency to be configured for the SAI peripheral.

BitRes,: Audio frequency to be configured.

ChnlNbr,: Channel number.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [801](#) of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), [AUDIO_IN_SAIx](#), [AUDIO_OK](#), [BSP_AUDIO_IN_MspInit\(\)](#), [BSP_AUDIO_OUT_ClockConfig\(\)](#), [BSP_AUDIO_OUT_MspInit\(\)](#), [CODEC_AUDIOFRAME_SLOT_02](#), [CODEC_AUDIOFRAME_SLOT_13](#), [haudio_in_sai](#), [SAIx_In_DelInit\(\)](#), and [SAIx_In_Init\(\)](#).

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

```
__weak void BSP_AUDIO_IN_MspDelInit ( SAI_HandleTypeDef * hsa
                                      void *                               Par
                                      )
```

DeInitializes BSP_AUDIO_IN MSP.

Parameters:

hsai,: SAI handle

Params

Return values:

None

Definition at line [1197](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_IN_SAIx](#), [AUDIO_IN_SAIx_CLK_DISABLE](#), [AUDIO_IN_SAIx_DMAx_IRQ](#), [AUDIO_IN_SAIx_SD_GPIO_PORT](#), and [AUDIO_IN_SAIx_SD_PIN](#).

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

```
__weak void BSP_AUDIO_IN_MspInit( SAI_HandleTypeDef * hsaI,  
                                  void * Parans  
                                )
```

Initializes BSP_AUDIO_IN MSP.

Parameters:

hsai,: SAI handle
Params

Return values:

None

Definition at line 1124 of file [stm32746g_discovery_audio.c](#).

References [AUDIO_IN_INT_GPIO_ENABLE](#), [AUDIO_IN_INT_GPIO_PIN](#), [AUDIO_IN_INT_GPIO_PORT](#), [AUDIO_IN_INT_IRQ](#), [AUDIO_IN_IRQ_PREPRIO](#), [AUDIO_IN_SAIx](#), [AUDIO_IN_SAIx_CLK_ENABLE](#), [AUDIO_IN_SAIx_DMAx_CHANNEL](#), [AUDIO_IN_SAIx_DMAx_CLK_ENABLE](#), [AUDIO_IN_SAIx_DMAx_IRQ](#), [AUDIO_IN_SAIx_DMAx_MEM_DATA_SIZE](#), [AUDIO_IN_SAIx_DMAx_PERIPH_DATA_SIZE](#), [AUDIO_IN_SAIx_DMAx_STREAM](#), [AUDIO_IN_SAIx_SD_AF](#), [AUDIO_IN_SAIx_SD_ENABLE](#), [AUDIO_IN_SAIx_SD_GPIO_PORT](#), and [AUDIO_IN_SAIx_SD_PIN](#).

Referenced by [BSP_AUDIO_IN_InitEx\(\)](#), and [BSP_AUDIO_IN_OUT_Init\(\)](#).

```
uint8_t BSP_AUDIO_IN_OUT_Init( uint16_t InputDevice,  
                               uint16_t OutputDevice,  
                               uint32_t AudioFreq,  
                               uint32_t BitRes,  
                               uint32_t ChnlNbr  
)  
{
```

Initializes wave recording and playback in parallel.

Parameters:

- InputDevice,: :** INPUT_DEVICE_DIGITAL_MICROPHONE_2
- OutputDevice,: :** OUTPUT_DEVICE_SPEAKER,
OUTPUT_DEVICE_HEADPHONE, or
OUTPUT_DEVICE_BOTH.
- AudioFreq,: :** Audio frequency to be configured for the SAI peripheral.
- BitRes,: :** Audio frequency to be configured.
- ChnlNbr,: :** Channel number.

Return values:

- AUDIO_OK** if correct communication, else wrong communication

Definition at line [879](#) of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), [AUDIO_IN_SAIx](#), [AUDIO_OK](#), [AUDIO_OUT_SAIx](#), [BSP_AUDIO_IN_MspInit\(\)](#), [BSP_AUDIO_OUT_ClockConfig\(\)](#), [BSP_AUDIO_OUT_MspInit\(\)](#), [CODEC_AUDIOFRAME_SLOT_02](#), [CODEC_AUDIOFRAME_SLOT_13](#), [haudio_in_sai](#), [haudio_out_sai](#), [SAIx_In_DelInit\(\)](#), [SAIx_In_Init\(\)](#), and [SAIx_Out_DelInit\(\)](#).

`uint8_t BSP_AUDIO_IN_Pause(void)`

Pauses the audio file stream.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [1011](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_OK](#), and [haudio_in_sai](#).

`uint8_t BSP_AUDIO_IN_Record(uint16_t * pbuf, uint32_t size)`

Starts audio recording.

Parameters:

pbuf,: Main buffer pointer for the recorded data storing
size,: size of the recorded buffer in number of elements
(typically number of half-words) Be careful that it is not
the same unit than `BSP_AUDIO_OUT_Play` function

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [963](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_ERROR](#), [AUDIO_OK](#), and [haudio_in_sai](#).

`uint8_t BSP_AUDIO_IN_Resume(void)`

Resumes the audio file stream.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [1023](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_OK](#), and [haudio_in_sai](#).

uint8_t BSP_AUDIO_IN_SetVolume (uint8_t Volume)

Controls the audio in volume level.

Parameters:

Volume,: Volume level in range
0(Mute)..80(+0dB)..100(+17.625dB)

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [1036](#) of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), [AUDIO_OK](#), and [AudioInVolume](#).

uint8_t BSP_AUDIO_IN_Stop (uint32_t Option)

Stops audio recording.

Parameters:

Option,: could be one of the following parameters

- CODEC_PDWN_SW: for software power off (by

writing registers). Then no need to reconfigure the Codec after power on.

- CODEC_PDWN_HW: completely shut down the codec (physically). Then need to reconfigure the Codec after power on.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line **985** of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), [AUDIO_OK](#), and [haudio_in_sai](#).

__weak void BSP_AUDIO_IN_TransferComplete_CallBack (void)

User callback when record buffer is filled.

Return values:

None

Definition at line **1090** of file [stm32746g_discovery_audio.c](#).

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

void HAL_SAI_RxCpltCallback (SAI_HandleTypeDef * hsai)

Rx Transfer completed callbacks.

Parameters:

hsai: SAI handle

Return values:

None

Definition at line [1068](#) of file [stm32746g_discovery_audio.c](#).

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

void HAL_SAI_RxHalfCpltCallback (SAI_HandleTypeDef * hsai)

Rx Half Transfer completed callbacks.

Parameters:

hsai,: SAI handle

Return values:

None

Definition at line [1079](#) of file [stm32746g_discovery_audio.c](#).

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

static void SAIx_In_DelInit (void) [static]

Deinitializes the output Audio Codec audio interface (SAI).

Return values:

None

Definition at line [1344](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_IN_SAIx](#), and [haudio_in_sai](#).

Referenced by [BSP_AUDIO_IN_DelInit\(\)](#), [BSP_AUDIO_IN_InitEx\(\)](#), and [BSP_AUDIO_IN_OUT_Init\(\)](#).

**static void SAIx_In_Init (uint32_t SaiOutMode,
 uint32_t SlotActive,**

```
    uint32_t AudioFreq  
)  
    [static]
```

Initializes the input Audio Codec audio interface (SAI).

Parameters:

SaiOutMode,: SAI_MODEMASTER_TX (for record and playback in parallel) or SAI_MODEMASTER_RX (for record only).

SlotActive,: CODEC_AUDIOFRAME_SLOT_02 or CODEC_AUDIOFRAME_SLOT_13

AudioFreq,: Audio frequency to be configured for the SAI peripheral.

Return values:

None

Definition at line 1239 of file [stm32746g_discovery_audio.c](#).

References [AUDIO_IN_SAIx](#), [AUDIO_OUT_SAIx](#), [haudio_in_sai](#), and [haudio_out_sai](#).

Referenced by [BSP_AUDIO_IN_InitEx\(\)](#), and [BSP_AUDIO_IN_OUT_Init\(\)](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Functions
STM32746G_DISCOVERY_AUDIO_IN Exported Functions				
STM32746G_DISCOVERY AUDIO				

Functions

uint8_t	BSP_AUDIO_IN_Init (uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording.
uint8_t	BSP_AUDIO_IN_InitEx (uint16_t InputDevice, uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording.
uint8_t	BSP_AUDIO_IN_OUT_Init (uint16_t InputDevice, uint16_t OutputDevice, uint32_t AudioFreq, uint32_t BitRes, uint32_t ChnlNbr)	Initializes wave recording and playback in parallel.
uint8_t	BSP_AUDIO_IN_Record (uint16_t *pData, uint32_t Size)	Starts audio recording.
uint8_t	BSP_AUDIO_IN_Stop (uint32_t Option)	Stops audio recording.
uint8_t	BSP_AUDIO_IN_Pause (void)	Pauses the audio file stream.
uint8_t	BSP_AUDIO_IN_Resume (void)	Resumes the audio file stream.
uint8_t	BSP_AUDIO_IN_SetVolume (uint8_t Volume)	Controls the audio in volume level.
void	BSP_AUDIO_IN_DeInit (void)	Deinit the audio IN peripherals.
void	BSP_AUDIO_IN_TransferComplete_CallBack (void)	User callback when record buffer is filled.
void	BSP_AUDIO_IN_HalfTransfer_CallBack (void)	Manages the DMA Half Transfer complete event.
void	BSP_AUDIO_IN_Error_CallBack (void)	Audio IN Error callback function.
void	BSP_AUDIO_IN_MspInit (SAI_HandleTypeDefDef *hsai, void *Params)	Initializes BSP_AUDIO_IN MSP.

```
void BSP_AUDIO_IN_MspDelInit (SAI_HandleTypeDef *hsai, void  
*Params)  
DeInitializes BSP_AUDIO_IN MSP.
```

Function Documentation

void [BSP_AUDIO_IN_DelInit](#)(void)

Deinit the audio IN peripherals.

Return values:

None

Definition at line [1056](#) of file [stm32746g_discovery_audio.c](#).

References [BSP_AUDIO_IN_MspDelInit\(\)](#), [haudio_in_sai](#), and [SAIx_In_DelInit\(\)](#).

void [BSP_AUDIO_IN_Error_CallBack](#)(void)

Audio IN Error callback function.

Return values:

None

Definition at line [1112](#) of file [stm32746g_discovery_audio.c](#).

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

void [BSP_AUDIO_IN_HalfTransfer_CallBack](#)(void)

Manages the DMA Half Transfer complete event.

Return values:

None

Definition at line [1101](#) of file [stm32746g_discovery_audio.c](#).

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

```
uint8_t BSP_AUDIO_IN_Init( uint32_t AudioFreq,  
                           uint32_t BitRes,  
                           uint32_t ChnlNbr  
                         )
```

Initializes wave recording.

Parameters:

- AudioFreq,:** Audio frequency to be configured for the SAI peripheral.
- BitRes,:** Audio frequency to be configured.
- ChnlNbr,:** Channel number.

Return values:

- AUDIO_OK** if correct communication, else wrong communication

Definition at line [788](#) of file [stm32746g_discovery_audio.c](#).

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

```
uint8_t BSP_AUDIO_IN_InitEx( uint16_t InputDevice,  
                            uint32_t AudioFreq,  
                            uint32_t BitRes,  
                            uint32_t ChnlNbr  
                          )
```

Initializes wave recording.

Parameters:

- InputDevice,:** INPUT_DEVICE_DIGITAL_MICROPHONE_2 or INPUT_DEVICE_INPUT_LINE_1

AudioFreq,: Audio frequency to be configured for the SAI peripheral.

BitRes,: Audio frequency to be configured.

ChnlNbr,: Channel number.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [801](#) of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), [AUDIO_IN_SAIx](#), [AUDIO_OK](#), [BSP_AUDIO_IN_MspInit\(\)](#), [BSP_AUDIO_OUT_ClockConfig\(\)](#), [BSP_AUDIO_OUT_MspInit\(\)](#), [CODEC_AUDIOFRAME_SLOT_02](#), [CODEC_AUDIOFRAME_SLOT_13](#), [haudio_in_sai](#), [SAIx_In_DelInit\(\)](#), and [SAIx_In_Init\(\)](#).

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

```
void BSP_AUDIO_IN_MspDelInit( SAI_HandleTypeDef * hsai,  
                                void * Params  
                            )
```

DeInitializes BSP_AUDIO_IN MSP.

Parameters:

hsai,: SAI handle

Params

Return values:

None

Definition at line [1197](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_IN_SAIx](#), [AUDIO_IN_SAIx_CLK_DISABLE](#), [AUDIO_IN_SAIx_DMAx_IRQ](#), [AUDIO_IN_SAIx_SD_GPIO_PORT](#), and [AUDIO_IN_SAIx_SD_PIN](#).

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

```
void BSP_AUDIO_IN_MspInit( SAI_HandleTypeDef * hsaI,  
                           void *                                     Params  
                         )
```

Initializes BSP_AUDIO_IN MSP.

Parameters:

hsai,: SAI handle
Params

Return values:

None

Definition at line 1124 of file [stm32746g_discovery_audio.c](#).

References [AUDIO_IN_INT_GPIO_ENABLE](#), [AUDIO_IN_INT_GPIO_PIN](#), [AUDIO_IN_INT_GPIO_PORT](#), [AUDIO_IN_INT_IRQ](#), [AUDIO_IN_IRQ_PREPRIO](#), [AUDIO_IN_SAIx](#), [AUDIO_IN_SAIx_CLK_ENABLE](#), [AUDIO_IN_SAIx_DMAx_CHANNEL](#), [AUDIO_IN_SAIx_DMAx_CLK_ENABLE](#), [AUDIO_IN_SAIx_DMAx_IRQ](#), [AUDIO_IN_SAIx_DMAx_MEM_DATA_SIZE](#), [AUDIO_IN_SAIx_DMAx_PERIPH_DATA_SIZE](#), [AUDIO_IN_SAIx_DMAx_STREAM](#), [AUDIO_IN_SAIx_SD_AF](#), [AUDIO_IN_SAIx_SD_ENABLE](#), [AUDIO_IN_SAIx_SD_GPIO_PORT](#), and [AUDIO_IN_SAIx_SD_PIN](#).

Referenced by [BSP_AUDIO_IN_InitEx\(\)](#), and [BSP_AUDIO_IN_OUT_Init\(\)](#).

```
uint8_t BSP_AUDIO_IN_OUT_Init( uint16_t InputDevice,  
                               uint16_t OutputDevice,  
                               uint32_t AudioFreq,  
                               uint32_t BitRes,  
                               uint32_t ChnlNbr  
)  
{
```

Initializes wave recording and playback in parallel.

Parameters:

- InputDevice,: :** INPUT_DEVICE_DIGITAL_MICROPHONE_2
- OutputDevice,: :** OUTPUT_DEVICE_SPEAKER,
OUTPUT_DEVICE_HEADPHONE, or
OUTPUT_DEVICE_BOTH.
- AudioFreq,: :** Audio frequency to be configured for the SAI peripheral.
- BitRes,: :** Audio frequency to be configured.
- ChnlNbr,: :** Channel number.

Return values:

- AUDIO_OK** if correct communication, else wrong communication

Definition at line [879](#) of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), [AUDIO_IN_SAIx](#), [AUDIO_OK](#), [AUDIO_OUT_SAIx](#), [BSP_AUDIO_IN_MspInit\(\)](#), [BSP_AUDIO_OUT_ClockConfig\(\)](#), [BSP_AUDIO_OUT_MspInit\(\)](#), [CODEC_AUDIOFRAME_SLOT_02](#), [CODEC_AUDIOFRAME_SLOT_13](#), [haudio_in_sai](#), [haudio_out_sai](#), [SAIx_In_DelInit\(\)](#), [SAIx_In_Init\(\)](#), and [SAIx_Out_DelInit\(\)](#).

`uint8_t BSP_AUDIO_IN_Pause(void)`

Pauses the audio file stream.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [1011](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_OK](#), and [haudio_in_sai](#).

`uint8_t BSP_AUDIO_IN_Record(uint16_t * pbuf, uint32_t size)`

Starts audio recording.

Parameters:

pbuf,: Main buffer pointer for the recorded data storing
size,: size of the recorded buffer in number of elements
(typically number of half-words) Be careful that it is not
the same unit than `BSP_AUDIO_OUT_Play` function

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [963](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_ERROR](#), [AUDIO_OK](#), and [haudio_in_sai](#).

`uint8_t BSP_AUDIO_IN_Resume(void)`

Resumes the audio file stream.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [1023](#) of file [stm32746g_discovery_audio.c](#).

References [AUDIO_OK](#), and [haudio_in_sai](#).

uint8_t BSP_AUDIO_IN_SetVolume (uint8_t Volume)

Controls the audio in volume level.

Parameters:

Volume,: Volume level in range
0(Mute)..80(+0dB)..100(+17.625dB)

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line [1036](#) of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), [AUDIO_OK](#), and [AudioInVolume](#).

uint8_t BSP_AUDIO_IN_Stop (uint32_t Option)

Stops audio recording.

Parameters:

Option,: could be one of the following parameters

- CODEC_PDWN_SW: for software power off (by

writing registers). Then no need to reconfigure the Codec after power on.

- CODEC_PDWN_HW: completely shut down the codec (physically). Then need to reconfigure the Codec after power on.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line **985** of file **stm32746g_discovery_audio.c**.

References **audio_drv**, **AUDIO_ERROR**, **AUDIO_I2C_ADDRESS**, **AUDIO_OK**, and **haudio_in_sai**.

void BSP_AUDIO_IN_TransferComplete_CallBack (void)

User callback when record buffer is filled.

Return values:

None

Definition at line **1090** of file **stm32746g_discovery_audio.c**.

Referenced by **HAL_SAI_RxCpltCallback()**.

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Functions		
STM32746G_DISCOVERY AUDIO Out Exported Functions			
STM32746G_DISCOVERY AUDIO			

Functions

uint8_t	BSP_AUDIO_OUT_Init (uint16_t OutputDevice, uint8_t Volume, uint32_t AudioFreq)	Configures the audio peripherals.
uint8_t	BSP_AUDIO_OUT_Play (uint16_t *pBuffer, uint32_t Size)	Starts playing audio stream from a data buffer for a determined size.
void	BSP_AUDIO_OUT_ChangeBuffer (uint16_t *pData, uint16_t Size)	Sends n-Bytes on the SAI interface.
uint8_t	BSP_AUDIO_OUT_Pause (void)	This function Pauses the audio file stream.
uint8_t	BSP_AUDIO_OUT_Resume (void)	This function Resumes the audio file stream.
uint8_t	BSP_AUDIO_OUT_Stop (uint32_t Option)	Stops audio playing and Power down the Audio Codec.
uint8_t	BSP_AUDIO_OUT_SetVolume (uint8_t Volume)	Controls the current audio volume level.
uint8_t	BSP_AUDIO_OUT_SetMute (uint32_t Cmd)	Enables or disables the MUTE mode by software.
uint8_t	BSP_AUDIO_OUT_SetOutputMode (uint8_t Output)	Switch dynamically (while audio file is played) the output target (speaker or headphone).
void	BSP_AUDIO_OUT_SetFrequency (uint32_t AudioFreq)	Updates the audio frequency.
void	BSP_AUDIO_OUT_SetAudioFrameSlot (uint32_t AudioFrameSlot)	Updates the Audio frame slot configuration.
void	BSP_AUDIO_OUT_DeInit (void)	Deinit the audio peripherals.

void	HAL_SAI_TxCpltCallback (SAI_HandleTypeDef *hsai)	Tx Transfer completed callbacks.
void	HAL_SAI_TxHalfCpltCallback (SAI_HandleTypeDef *hsai)	Tx Half Transfer completed callbacks.
void	HAL_SAI_ErrorCallback (SAI_HandleTypeDef *hsai)	SAI error callbacks.
<u>weak void</u>	BSP_AUDIO_OUT_TransferComplete_CallBack (void)	Manages the DMA full Transfer complete event.
<u>weak void</u>	BSP_AUDIO_OUT_HalfTransfer_CallBack (void)	Manages the DMA Half Transfer complete event.
<u>weak void</u>	BSP_AUDIO_OUT_Error_CallBack (void)	Manages the DMA FIFO error event.
<u>weak void</u>	BSP_AUDIO_OUT_MspInit (SAI_HandleTypeDef *hsai, void *Params)	Initializes BSP_AUDIO_OUT MSP.
<u>weak void</u>	BSP_AUDIO_OUT_MspDeInit (SAI_HandleTypeDef *hsai, void *Params)	Deinitializes SAI MSP.
<u>weak void</u>	BSP_AUDIO_OUT_ClockConfig (SAI_HandleTypeDef *hsai, uint32_t AudioFreq, void *Params)	Clock Config.
static void	SAIx_Out_Init (uint32_t AudioFreq)	Initializes the output Audio Codec audio interface (SAI).
static void	SAIx_Out_DeInit (void)	Deinitializes the output Audio Codec audio interface (SAI).

Function Documentation

```
void BSP_AUDIO_OUT_ChangeBuffer ( uint16_t * pData,  
                                  uint16_t Size  
                                )
```

Sends n-Bytes on the SAI interface.

Parameters:

pData,: pointer on data address
Size,: number of data to be written

Return values:

None

Definition at line 246 of file [stm32746g_discovery_audio.c](#).

References [haudio_out_sai](#).

```
void BSP_AUDIO_OUT_ClockConfig ( SAI_HandleTypeDef * hsaI,  
                                 uint32_t          AudioFreq,  
                                 void *           Params  
                               )
```

Clock Config.

Parameters:

hsai,: might be required to set audio peripheral predivider if any.
AudioFreq,: Audio frequency used to play the audio stream.
Params

Note:

This API is called by [BSP_AUDIO_OUT_Init\(\)](#) and [BSP_AUDIO_OUT_SetFrequency\(\)](#). Being `__weak` it can be overwritten by the application.

Return values:

None

Definition at line [657](#) of file [stm32746g_discovery_audio.c](#).

Referenced by [BSP_AUDIO_IN_InitEx\(\)](#), [BSP_AUDIO_IN_OUT_Init\(\)](#), [BSP_AUDIO_OUT_Init\(\)](#), and [BSP_AUDIO_OUT_SetFrequency\(\)](#).

void [BSP_AUDIO_OUT_DeInit](#)([void](#))

Deinit the audio peripherals.

Return values:

None

Definition at line [443](#) of file [stm32746g_discovery_audio.c](#).

References [BSP_AUDIO_OUT_MspDeInit\(\)](#), [haudio_out_sai](#), and [SAIx_Out_DeInit\(\)](#).

void [BSP_AUDIO_OUT_Error_CallBack](#)([void](#))

Manages the DMA FIFO error event.

Return values:

None

Definition at line [519](#) of file [stm32746g_discovery_audio.c](#).

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

```
void BSP_AUDIO_OUT_HalfTransfer_CallBack( void )
```

Manages the DMA Half Transfer complete event.

Return values:

None

Definition at line [511](#) of file [stm32746g_discovery_audio.c](#).

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

```
uint8_t BSP_AUDIO_OUT_Init( uint16_t OutputDevice,  
                           uint8_t Volume,  
                           uint32_t AudioFreq  
                         )
```

Configures the audio peripherals.

Parameters:

OutputDevice,: OUTPUT_DEVICE_SPEAKER,
OUTPUT_DEVICE_HEADPHONE, or
OUTPUT_DEVICE_BOTH.

Volume,: Initial volume level (from 0 (Mute) to 100
(Max))

AudioFreq,: Audio frequency used to play the audio
stream.

Note:

The I2S PLL input clock must be done in the user application.

Return values:

AUDIO_OK if correct communication, else wrong
communication

Definition at line [171](#) of file [stm32746g_discovery_audio.c](#).

References `audio_drv`, `AUDIO_ERROR`, `AUDIO_I2C_ADDRESS`, `AUDIO_OK`, `AUDIO_OUT_SAIx`, `BSP_AUDIO_OUT_ClockConfig()`, `BSP_AUDIO_OUT_MspInit()`, `haudio_out_sai`, `SAIx_Out_DeInit()`, and `SAIx_Out_Init()`.

```
void BSP_AUDIO_OUT_MspDeInit( SAI_HandleTypeDef * hsaI,  
                                void * Params  
                            )
```

Deinitializes SAI MSP.

Parameters:

`hsaI`: SAI handle
`Params`

Return values:

`None`

Definition at line 612 of file `stm32746g_discovery_audio.c`.

References `AUDIO_OUT_SAIx`, `AUDIO_OUT_SAIx_CLK_DISABLE`, `AUDIO_OUT_SAIx_DMAx_IRQ`, `AUDIO_OUT_SAIx_FS_GPIO_PORT`, `AUDIO_OUT_SAIx_FS_PIN`, `AUDIO_OUT_SAIx_MCLK_GPIO_PORT`, `AUDIO_OUT_SAIx_MCLK_PIN`, `AUDIO_OUT_SAIx_SCK_PIN`, `AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT`, and `AUDIO_OUT_SAIx_SD_PIN`.

Referenced by `BSP_AUDIO_OUT_DeInit()`.

```
void BSP_AUDIO_OUT_MspInit( SAI_HandleTypeDef * hsaI,  
                                void * Params  
                            )
```

Initializes BSP_AUDIO_OUT MSP.

Parameters:

hsai,: SAI handle

Params

Return values:

None

Definition at line 529 of file [stm32746g_discovery_audio.c](#).

References [AUDIO_OUT_IRQ_PREPRIO](#), [AUDIO_OUT_SAIx](#),
[AUDIO_OUT_SAIx_CLK_ENABLE](#),
[AUDIO_OUT_SAIx_DMAx_CHANNEL](#),
[AUDIO_OUT_SAIx_DMAx_CLK_ENABLE](#),
[AUDIO_OUT_SAIx_DMAx_IRQ](#),
[AUDIO_OUT_SAIx_DMAx_MEM_DATA_SIZE](#),
[AUDIO_OUT_SAIx_DMAx_PERIPH_DATA_SIZE](#),
[AUDIO_OUT_SAIx_DMAx_STREAM](#),
[AUDIO_OUT_SAIx_FS_ENABLE](#),
[AUDIO_OUT_SAIx_FS_GPIO_PORT](#), [AUDIO_OUT_SAIx_FS_PIN](#),
[AUDIO_OUT_SAIx_FS_SD_MCLK_AF](#),
[AUDIO_OUT_SAIx_MCLK_ENABLE](#),
[AUDIO_OUT_SAIx_MCLK_GPIO_PORT](#),
[AUDIO_OUT_SAIx_MCLK_PIN](#), [AUDIO_OUT_SAIx_SCK_AF](#),
[AUDIO_OUT_SAIx_SCK_PIN](#),
[AUDIO_OUT_SAIx_SCK_SD_ENABLE](#),
[AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT](#), and
[AUDIO_OUT_SAIx_SD_PIN](#).

Referenced by [BSP_AUDIO_IN_InitEx\(\)](#),
[BSP_AUDIO_IN_OUT_Init\(\)](#), and [BSP_AUDIO_OUT_Init\(\)](#).

uint8_t [BSP_AUDIO_OUT_Pause](#)(void)

This function Pauses the audio file stream.

In case of using DMA, the DMA Pause feature is used.

Note:

When calling **BSP_AUDIO_OUT_Pause()** function for pause, only **BSP_AUDIO_OUT_Resume()** function should be called for resume (use of **BSP_AUDIO_OUT_Play()** function for resume could lead to unexpected behaviour).

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line **259** of file **stm32746g_discovery_audio.c**.

References **audio_drv**, **AUDIO_ERROR**, **AUDIO_I2C_ADDRESS**, **AUDIO_OK**, and **haudio_out_sai**.

```
uint8_t BSP_AUDIO_OUT_Play ( uint16_t* pBuffer,  
                            uint32_t  Size  
                          )
```

Starts playing audio stream from a data buffer for a determined size.

Parameters:

pBuffer,: Pointer to the buffer

Size,: Number of audio data in BYTES unit. In memory, first element is for left channel, second element is for right channel

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line 224 of file `stm32746g_discovery_audio.c`.

References `audio_drv`, `AUDIO_ERROR`, `AUDIO_I2C_ADDRESS`, `AUDIO_OK`, `AUDIODATA_SIZE`, `DMA_MAX`, and `haudio_out_sai`.

`uint8_t BSP_AUDIO_OUT_Resume(void)`

This function Resumes the audio file stream.

Note:

When calling `BSP_AUDIO_OUT_Pause()` function for pause, only `BSP_AUDIO_OUT_Resume()` function should be called for resume (use of `BSP_AUDIO_OUT_Play()` function for resume could lead to unexpected behaviour).

Return values:

`AUDIO_OK` if correct communication, else wrong communication

Definition at line 283 of file `stm32746g_discovery_audio.c`.

References `audio_drv`, `AUDIO_ERROR`, `AUDIO_I2C_ADDRESS`, `AUDIO_OK`, and `haudio_out_sai`.

`void BSP_AUDIO_OUT_SetAudioFrameSlot(uint32_t AudioFrame`

Updates the Audio frame slot configuration.

Parameters:

`AudioFrameSlot`,: specifies the audio Frame slot This parameter can be one of the following values

- `CODEC_AUDIOFRAME_SLOT_0123`
- `CODEC_AUDIOFRAME_SLOT_02`
- `CODEC_AUDIOFRAME_SLOT_13`

Note:

This API should be called after the [BSP_AUDIO_OUT_Init\(\)](#) to adjust the audio frame slot.

Return values:

None

Definition at line [426](#) of file [stm32746g_discovery_audio.c](#).

References [haudio_out_sai](#).

void BSP_AUDIO_OUT_SetFrequency (uint32_t AudioFreq)

Updates the audio frequency.

Parameters:

AudioFreq,: Audio frequency used to play the audio stream.

Note:

This API should be called after the [BSP_AUDIO_OUT_Init\(\)](#) to adjust the audio frequency.

Return values:

None

Definition at line [399](#) of file [stm32746g_discovery_audio.c](#).

References [BSP_AUDIO_OUT_ClockConfig\(\)](#), and [haudio_out_sai](#).

uint8_t BSP_AUDIO_OUT_SetMute (uint32_t Cmd)

Enables or disables the MUTE mode by software.

Parameters:

Cmd,: Could be AUDIO_MUTE_ON to mute sound or

AUDIO_MUTE_OFF to unmute the codec and restore previous volume level.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line 357 of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), and [AUDIO_OK](#).

uint8_t BSP_AUDIO_OUT_SetOutputMode (uint8_t Output)

Switch dynamically (while audio file is played) the output target (speaker or headphone).

Parameters:

Output,: The audio output target:
OUTPUT_DEVICE_SPEAKER,
OUTPUT_DEVICE_HEADPHONE or
OUTPUT_DEVICE_BOTH

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line 378 of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), and [AUDIO_OK](#).

uint8_t BSP_AUDIO_OUT_SetVolume (uint8_t Volume)

Controls the current audio volume level.

Parameters:

Volume,: Volume level to be set in percentage from 0% to 100% (0 for Mute and 100 for Max volume level).

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line 337 of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), and [AUDIO_OK](#).

uint8_t BSP_AUDIO_OUT_Stop (uint32_t Option)

Stops audio playing and Power down the Audio Codec.

Parameters:

Option,: could be one of the following parameters

- CODEC_PDWN_SW: for software power off (by writing registers). Then no need to reconfigure the Codec after power on.
- CODEC_PDWN_HW: completely shut down the codec (physically). Then need to reconfigure the Codec after power on.

Return values:

AUDIO_OK if correct communication, else wrong communication

Definition at line 309 of file [stm32746g_discovery_audio.c](#).

References [audio_drv](#), [AUDIO_ERROR](#), [AUDIO_I2C_ADDRESS](#), [AUDIO_OK](#), and [haudio_out_sai](#).

void [BSP_AUDIO_OUT_TransferComplete_CallBack](#)(void)

Manages the DMA full Transfer complete event.

Return values:

None

Definition at line [503](#) of file [stm32746g_discovery_audio.c](#).

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

void [HAL_SAI_ErrorCallback](#)(SAI_HandleTypeDef * hsai)

SAI error callbacks.

Parameters:

hsai,: SAI handle

Return values:

None

Definition at line [479](#) of file [stm32746g_discovery_audio.c](#).

References [BSP_AUDIO_IN_Error_Callback\(\)](#),
[BSP_AUDIO_OUT_Error_Callback\(\)](#), [haudio_in_sai](#), and
[haudio_out_sai](#).

void [HAL_SAI_TxCpltCallback](#)(SAI_HandleTypeDef * hsai)

Tx Transfer completed callbacks.

Parameters:

hsai,: SAI handle

Return values:

None

Definition at line [455](#) of file `stm32746g_discovery_audio.c`.

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

void HAL_SAI_TxHalfCpltCallback (SAI_HandleTypeDef * hsai)

Tx Half Transfer completed callbacks.

Parameters:

hsai,: SAI handle

Return values:

None

Definition at line [467](#) of file `stm32746g_discovery_audio.c`.

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

static void SAIx_Out_DelInit (void) [static]

Deinitializes the output Audio Codec audio interface (SAI).

Return values:

None

Definition at line [762](#) of file `stm32746g_discovery_audio.c`.

References [AUDIO_OUT_SAIx](#), and [haudio_out_sai](#).

Referenced by [BSP_AUDIO_IN_OUT_Init\(\)](#),
[BSP_AUDIO_OUT_DelInit\(\)](#), and [BSP_AUDIO_OUT_Init\(\)](#).

static void SAIx_Out_Init (uint32_t AudioFreq) [static]

Initializes the output Audio Codec audio interface (SAI).

Parameters:

AudioFreq,: Audio frequency to be configured for the SAI peripheral.

Note:

The default SlotActive configuration is set to CODEC_AUDIOFRAME_SLOT_0123 and user can update this configuration using

Return values:

None

Definition at line 706 of file [stm32746g_discovery_audio.c](#).

References [AUDIO_OUT_SAIx](#),
[CODEC_AUDIOFRAME_SLOT_0123](#), and [haudio_out_sai](#).

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

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Functions
STM32746G_DISCOVERY_CAMERA Exported Functions				
<u>STM32746G_DISCOVERY_CAMERA</u>				

Functions

uint8_t	BSP_CAMERA_Init (uint32_t Resolution)	Initializes the camera.
uint8_t	BSP_CAMERA_DelInit (void)	DeInitializes the camera.
void	BSP_CAMERA_ContinuousStart (uint8_t *buff)	Starts the camera capture in continuous mode.
void	BSP_CAMERA_SnapshotStart (uint8_t *buff)	Starts the camera capture in snapshot mode.
void	BSP_CAMERA_Suspend (void)	Suspend the CAMERA capture.
void	BSP_CAMERA_Resume (void)	Resume the CAMERA capture.
uint8_t	BSP_CAMERA_Stop (void)	Stop the CAMERA capture.
void	BSP_CAMERA_PwrUp (void)	CAMERA power up.
void	BSP_CAMERA_PwrDown (void)	CAMERA power down.
void	BSP_CAMERA_ContrastBrightnessConfig (uint32_t contrast_level, uint32_t brightness_level)	Configures the camera contrast and brightness.
void	BSP_CAMERA_BlackWhiteConfig (uint32_t Mode)	Configures the camera white balance.
void	BSP_CAMERA_ColorEffectConfig (uint32_t Effect)	Configures the camera color effect.
__weak void	BSP_CAMERA_MspInit (DCMI_HandleTypeDefDef *hdcmi, void *Params)	Initializes the DCMI MSP.
__weak void	BSP_CAMERA_MspDelInit (DCMI_HandleTypeDefDef *hdcmi, void *Params)	DeInitializes the DCMI MSP.

void	HAL_DCMI_LineEventCallback (DCMI_HandleTypeDef *hdcmi) Line event callback.
<u>weak</u> void	BSP_CAMERA_LineEventCallback (void) Line Event callback.
void	HAL_DCMI_VsyncEventCallback (DCMI_HandleTypeDef *hdcmi) VSYNC event callback.
<u>weak</u> void	BSP_CAMERA_VsyncEventCallback (void) VSYNC Event callback.
void	HAL_DCMI_FrameEventCallback (DCMI_HandleTypeDef *hdcmi) Frame event callback.
<u>weak</u> void	BSP_CAMERA_FrameEventCallback (void) Frame Event callback.
void	HAL_DCMI_ErrorCallback (DCMI_HandleTypeDef *hdcmi) Error callback.
<u>weak</u> void	BSP_CAMERA_ErrorCallback (void) Error callback.
static uint32_t	GetSize (uint32_t resolution) Get the capture size in pixels unit.

Function Documentation

`void BSP_CAMERA_BlackWhiteConfig (uint32_t Mode)`

Configures the camera white balance.

Parameters:

Mode,: black_white mode This parameter can be one of the following values:

- CAMERA_BLACK_WHITE_BW
- CAMERA_BLACK_WHITE_NEGATIVE
- CAMERA_BLACK_WHITE_BW_NEGATIVE
- CAMERA_BLACK_WHITE_NORMAL

Return values:

None

Definition at line 363 of file [stm32746g_discovery_camera.c](#).

References [camera_drv](#), and [CameraHwAddress](#).

`void BSP_CAMERA_ColorEffectConfig (uint32_t Effect)`

Configures the camera color effect.

Parameters:

Effect,: Color effect This parameter can be one of the following values:

- CAMERA_COLOR_EFFECT_ANTIQUE
- CAMERA_COLOR_EFFECT_BLUE
- CAMERA_COLOR_EFFECT_GREEN
- CAMERA_COLOR_EFFECT_RED

Return values:

None

Definition at line [381](#) of file `stm32746g_discovery_camera.c`.

References `camera_drv`, and `CameraHwAddress`.

```
void BSP_CAMERA_ContinuousStart( uint8_t * buff )
```

Starts the camera capture in continuous mode.

Parameters:

buff,: pointer to the camera output buffer

Return values:

None

Definition at line 223 of file `stm32746g_discovery_camera.c`.

References [CameraCurrentResolution](#), [GetSize\(\)](#), and [hDcmiHandler](#).

```
void BSP_CAMERA_ContrastBrightnessConfig ( uint32_t contrast_
                                            uint32_t brightness_
                                         )
```

Configures the camera contrast and brightness.

Parameters:

contrast_level,: Contrast level This parameter can be one of the following values:

- CAMERA_CONTRAST_LEVEL4: for contrast +2
 - CAMERA_CONTRAST_LEVEL3: for contrast +1
 - CAMERA_CONTRAST_LEVEL2: for contrast 0
 - CAMERA_CONTRAST_LEVEL1: for

- contrast -1
- CAMERA_CONTRAST_LEVEL0: for contrast -2

brightness_level,: Contrast level This parameter can be one of the following values:

- CAMERA_BRIGHTNESS_LEVEL4: for brightness +2
- CAMERA_BRIGHTNESS_LEVEL3: for brightness +1
- CAMERA_BRIGHTNESS_LEVEL2: for brightness 0
- CAMERA_BRIGHTNESS_LEVEL1: for brightness -1
- CAMERA_BRIGHTNESS_LEVEL0: for brightness -2

Return values:

None

Definition at line [345](#) of file [stm32746g_discovery_camera.c](#).

References [camera_drv](#), and [CameraHwAddress](#).

uint8_t BSP_CAMERA_DeInit(void)

DeInitializes the camera.

Return values:

Camera status

Definition at line [209](#) of file [stm32746g_discovery_camera.c](#).

References [BSP_CAMERA_MspDeInit\(\)](#), [CAMERA_OK](#), and [hDcmiHandler](#).

void BSP_CAMERA_ErrorCallback(void)

Error callback.

Return values:

None

Definition at line [628](#) of file [stm32746g_discovery_camera.c](#).

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

void BSP_CAMERA_FrameEventCallback(void)

Frame Event callback.

Return values:

None

Definition at line [607](#) of file [stm32746g_discovery_camera.c](#).

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

uint8_t BSP_CAMERA_Init(uint32_t Resolution)

Initializes the camera.

Parameters:

Resolution : camera sensor requested resolution (x, y) :
standard resolution naming QQVGA, QVGA, VGA
...

Return values:

Camera status

Definition at line [140](#) of file [stm32746g_discovery_camera.c](#).

References [BSP_CAMERA_MspInit\(\)](#), [BSP_CAMERA_PwrUp\(\)](#), [CAMERA_480x272_RES_X](#), [CAMERA_480x272_RES_Y](#), [camera_drv](#), [CAMERA_ERROR](#), [CAMERA_I2C_ADDRESS](#), [CAMERA_NOT_SUPPORTED](#), [CAMERA_OK](#), [CAMERA_VGA_RES_X](#), [CAMERA_VGA_RES_Y](#), [CameraCurrentResolution](#), [CameraHwAddress](#), and [hDcmiHandler](#).

void BSP_CAMERA_LineEventCallback (void)

Line Event callback.

Return values:

None

Definition at line [565](#) of file [stm32746g_discovery_camera.c](#).

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

```
void BSP_CAMERA_MspDeInit ( DCMI_HandleTypeDef * hdcmi,  
                           void *                                     Params  
                         )
```

DeInitializes the DCMI MSP.

Parameters:

hdcmi,: HDMI handle
Params

Return values:

None

Definition at line [533](#) of file [stm32746g_discovery_camera.c](#).

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

```
void BSP_CAMERA_MspInit ( DCMI_HandleTypeDef * hdcmi,  
                           void * Params  
                         )
```

Initializes the DCMI MSP.

Parameters:

hdcmi,: HDMI handle
Params

Return values:

None

Definition at line 436 of file [stm32746g_discovery_camera.c](#).

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

```
void BSP_CAMERA_PwrDown ( void )
```

CAMERA power down.

Return values:

None

Definition at line 308 of file [stm32746g_discovery_camera.c](#).

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

```
void BSP_CAMERA_PwrUp ( void )
```

CAMERA power up.

Return values:

None

Definition at line [283](#) of file [stm32746g_discovery_camera.c](#).

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

void BSP_CAMERA_Resume(void)

Resume the CAMERA capture.

Return values:

None

Definition at line [254](#) of file [stm32746g_discovery_camera.c](#).

References [hDcmiHandler](#).

void BSP_CAMERA_SnapshotStart(uint8_t * buff)

Starts the camera capture in snapshot mode.

Parameters:

buff,: pointer to the camera output buffer

Return values:

None

Definition at line [234](#) of file [stm32746g_discovery_camera.c](#).

References [CameraCurrentResolution](#), [GetSize\(\)](#), and [hDcmiHandler](#).

uint8_t BSP_CAMERA_Stop(void)

Stop the CAMERA capture.

Return values:

Camera status

Definition at line [264](#) of file [stm32746g_discovery_camera.c](#).

References [BSP_CAMERA_PwrDown\(\)](#), [CAMERA_ERROR](#), [CAMERA_OK](#), and [hDcmiHandler](#).

void [BSP_CAMERA_Suspend](#)(void)

Suspend the CAMERA capture.

Return values:

None

Definition at line [244](#) of file [stm32746g_discovery_camera.c](#).

References [hDcmiHandler](#).

void [BSP_CAMERA_VsyncEventCallback](#)(void)

VSYNC Event callback.

Return values:

None

Definition at line [586](#) of file [stm32746g_discovery_camera.c](#).

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

static uint32_t [GetSize](#)(uint32_t resolution) [static]

Get the capture size in pixels unit.

Parameters:

resolution,: the current resolution.

Return values:

capture size in pixels unit.

Definition at line 394 of file [stm32746g_discovery_camera.c](#).

Referenced by [BSP_CAMERA_ContinuousStart\(\)](#), and
[BSP_CAMERA_SnapshotStart\(\)](#).

void HAL_DCMI_ErrorCallback (DCMI_HandleTypeDef * hdcmi)

Error callback.

Parameters:

hdcmi,: pointer to the DCMI handle

Return values:

None

Definition at line 619 of file [stm32746g_discovery_camera.c](#).

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

void HAL_DCMI_FrameEventCallback (DCMI_HandleTypeDef * hdcmi)

Frame event callback.

Parameters:

hdcmi,: pointer to the DCMI handle

Return values:

None

Definition at line 598 of file [stm32746g_discovery_camera.c](#).

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

void HAL_DCMI_LineEventCallback (DCMI_HandleTypeDef * hdcmi)

Line event callback.

Parameters:

hdcmi: pointer to the DCMI handle

Return values:

None

Definition at line 556 of file [stm32746g_discovery_camera.c](#).

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

void HAL_DCMI_VsyncEventCallback (DCMI_HandleTypeDef * hdcmi)

VSYNC event callback.

Parameters:

hdcmi: pointer to the DCMI handle

Return values:

None

Definition at line 577 of file [stm32746g_discovery_camera.c](#).

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

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Defines				STM32746G_DISCOVERY_CAMERA Exported Constants
STM32746G_DISCOVERY_CAMERA				

Defines

```
#define BSP_CAMERA_IRQHandler DCMI_IRQHandler  
#define BSP_CAMERA_DMA_IRQHandler DMA2_Stream1_IRQHandler
```

Define Documentation

```
#define BSP_CAMERA_DMA_IRQHandler DMA2_Stream1_IRQHandler
```

Definition at line [93](#) of file [stm32746g_discovery_camera.h](#).

```
#define BSP_CAMERA_IRQHandler DCMI_IRQHandler
```

Definition at line [92](#) of file [stm32746g_discovery_camera.h](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Functions
STM32746G_DISCOVERY_EEPROM Exported Functions				
STM32746G_DISCOVERY_EEPROM				

Functions

uint32_t	BSP_EEPROM_Init (void) Initializes peripherals used by the I2C EEPROM driver.
uint8_t	BSP_EEPROM_DelInit (void) DeInitializes the EEPROM.
uint32_t	BSP_EEPROM_ReadBuffer (uint8_t *pBuffer, uint16_t ReadAddr, uint16_t *NumByteToRead) Reads a block of data from the EEPROM.
uint32_t	BSP_EEPROM_WritePage (uint8_t *pBuffer, uint16_t WriteAddr, uint8_t *NumByteToWrite) Writes more than one byte to the EEPROM with a single WRITE cycle.
uint32_t	BSP_EEPROM_WriteBuffer (uint8_t *pBuffer, uint16_t WriteAddr, uint16_t NumByteToWrite) Writes buffer of data to the I2C EEPROM.
uint32_t	BSP_EEPROM_WaitEepromStandbyState (void) Wait for EEPROM Standby state.
__weak void	BSP_EEPROM_TIMEOUT_UserCallback (void) Basic management of the timeout situation.
void	EEPROM_IO_Init (void) Initializes peripherals used by the I2C EEPROM driver.
HAL_StatusTypeDef	EEPROM_IO_WriteData (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize) Write data to I2C EEPROM driver in using DMA channel.
HAL_StatusTypeDef	EEPROM_IO_ReadData (uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize)

	Read data from I2C EEPROM driver in using DMA channel.
HAL_StatusTypeDef	EEPROM_IO_IsDeviceReady (uint16_t DevAddress, uint32_t Trials) Checks if target device is ready for communication.

Function Documentation

`uint8_t BSP_EEPROM_DeInit(void)`

DeInitializes the EEPROM.

Return values:

EEPROM state

Definition at line [178](#) of file `stm32746g_discovery_eeprom.c`.

References [EEPROM_OK](#).

`uint32_t BSP_EEPROM_Init(void)`

Initializes peripherals used by the I2C EEPROM driver.

Note:

There are 2 different versions of M24LR64 (A01 & A02). Then try to connect on 1st one (EEPROM_I2C_ADDRESS_A01) and if problem, check the 2nd one (EEPROM_I2C_ADDRESS_A02)

Return values:

EEPROM_OK (0) if operation is correctly performed, else return value different from EEPROM_OK (0)

Definition at line [155](#) of file `stm32746g_discovery_eeprom.c`.

References [EEPROM_FAIL](#), [EEPROM_I2C_ADDRESS_A01](#), [EEPROM_I2C_ADDRESS_A02](#), [EEPROM_IO_Init\(\)](#), [EEPROM_IO_IsDeviceReady\(\)](#), [EEPROM_MAX_TRIALS](#), [EEPROM_OK](#), and [EEPROMAddress](#).

`uint32_t BSP_EEPROM_ReadBuffer(uint8_t * pBuffer,`

```
    uint16_t ReadAddr,  
    uint16_t * NumByteToRead  
)
```

Reads a block of data from the EEPROM.

Parameters:

- pBuffer,:** pointer to the buffer that receives the data read from the EEPROM.
- ReadAddr,:** EEPROM's internal address to start reading from.
- NumByteToRead,:** pointer to the variable holding number of bytes to be read from the EEPROM.

Note:

The variable pointed by NumByteToRead is reset to 0 when all the data are read from the EEPROM. Application should monitor this variable in order know when the transfer is complete.

Return values:

- EEPROM_OK** (0) if operation is correctly performed, else return value different from EEPROM_OK (0) or the timeout user callback.

Definition at line 199 of file [stm32746g_discovery_eeprom.c](#).

References [BSP_EEPROM_TIMEOUT_UserCallback\(\)](#), [EEPROM_FAIL](#), [EEPROM_IO_ReadData\(\)](#), [EEPROM_OK](#), [EEPROMAddress](#), and [EEPROMDataRead](#).

```
void BSP_EEPROM_TIMEOUT_UserCallback ( void )
```

Basic management of the timeout situation.

Return values:

None

Definition at line 454 of file `stm32746g_discovery_eeprom.c`.

Referenced by `BSP_EEPROM_ReadBuffer()`,
`BSP_EEPROM_WaitEepromStandbyState()`, and
`BSP_EEPROM_WritePage()`.

```
uint32_t BSP_EEPROM_WaitEepromStandbyState( void )
```

Wait for EEPROM Standby state.

Note:

This function allows to wait and check that EEPROM has finished the last operation. It is mostly used after Write operation: after receiving the buffer to be written, the EEPROM may need additional time to actually perform the write operation. During this time, it doesn't answer to I2C packets addressed to it. Once the write operation is complete the EEPROM responds to its address.

Return values:

EEPROM_OK (0) if operation is correctly performed, else return value different from EEPROM_OK (0) or the timeout user callback.

Definition at line 438 of file [stm32746g_discovery_eeprom.c](#).

References `BSP_EEPROM_TIMEOUT_UserCallback()`, `EPPROM_IO_IsDeviceReady()`, `EPPROM_MAX_TRIALS`, `EPPROM_OK`, `EPPROM_TIMEOUT`, and `EPPROMAddress`.

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

```
    uint16_t NumByteToWrite
```

```
)
```

Writes buffer of data to the I2C EEPROM.

Parameters:

pBuffer,: pointer to the buffer containing the data to be written to the EEPROM.

WriteAddr,: EEPROM's internal address to write to.

NumByteToWrite,: number of bytes to write to the EEPROM.

Return values:

EEPROM_OK (0) if operation is correctly performed, else return value different from EEPROM_OK (0) or the timeout user callback.

Definition at line [281](#) of file [stm32746g_discovery_eeprom.c](#).

References [BSP_EEPROM_WritePage\(\)](#), [EEPROM_OK](#), and [EEPROM_PAGESIZE](#).

```
uint32_t BSP_EEPROM_WritePage ( uint8_t * pBuffer,  
                                uint16_t WriteAddr,  
                                uint8_t * NumByteToWrite  
)
```

Writes more than one byte to the EEPROM with a single WRITE cycle.

Note:

The number of bytes (combined to write start address) must not cross the EEPROM page boundary. This function can only write into the boundaries of an EEPROM page. This function doesn't check on boundaries condition (in this driver the function [BSP_EEPROM_WriteBuffer\(\)](#) which calls

BSP_EEPROM_WritePage() is responsible of checking on Page boundaries).

Parameters:

pBuffer,: pointer to the buffer containing the data to be written to the EEPROM.

WriteAddr,: EEPROM's internal address to write to.

NumByteToWrite,: pointer to the variable holding number of bytes to be written into the EEPROM.

Note:

The variable pointed by NumByteToWrite is reset to 0 when all the data are written to the EEPROM. Application should monitor this variable in order know when the transfer is complete.

This function just configure the communication and enable the DMA channel to transfer data. Meanwhile, the user application may perform other tasks in parallel.

Return values:

EEPROM_OK (0) if operation is correctly performed, else return value different from EEPROM_OK (0) or the timeout user callback.

Definition at line [246](#) of file [stm32746g_discovery_eeprom.c](#).

References [BSP_EEPROM_TIMEOUT_UserCallback\(\)](#), [BSP_EEPROM_WaitEepromStandbyState\(\)](#), [EEPROM_FAIL](#), [EEPROM_IO_WriteData\(\)](#), [EEPROM_OK](#), [EEPROMAddress](#), and [EEPROMDataWrite](#).

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

void **EEPROM_IO_Init (void)**

Initializes peripherals used by the I2C EEPROM driver.

Return values:

None

Definition at line 790 of file `stm32746g_discovery.c`.

References `hi2cExtHandler`, and `I2Cx_Init()`.

Referenced by **BSP_EEPROM_Init()**.

```
HAL_StatusTypeDef EEPROM_IO_IsDeviceReady( uint16_t DevAd  
                                         uint32_t Trials  
                                         )
```

Checks if target device is ready for communication.

Note:

This function is used with Memory devices

Parameters:

DevAddress,: Target device address

Trials,: Number of trials

Return values:

HAL status

Definition at line 828 of file [stm32746g_discovery.c](#).

References `hi2cExtHandler`, and `I2Cx_IsDeviceReady()`.

Referenced by **BSP_EEPROM_Init()**, and
BSP_EEPROM_WaitEepromStandbyState().

```
        uint32_t BufferSize  
    )
```

Read data from I2C EEPROM driver in using DMA channel.

Parameters:

DevAddress,: Target device address
MemAddress,: Internal memory address
pBuffer,: Pointer to data buffer
BufferSize,: Amount of data to be read

Return values:

HAL status

Definition at line 816 of file [stm32746g_discovery.c](#).

References [hI2cExtHandler](#), and [I2Cx_ReadMultiple\(\)](#).

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

```
HAL_StatusTypeDef EEPROM_IO_WriteData ( uint16_t DevAddress,  
                                         uint16_t MemAddress,  
                                         uint8_t * pBuffer,  
                                         uint32_t BufferSize  
    )
```

Write data to I2C EEPROM driver in using DMA channel.

Parameters:

DevAddress,: Target device address
MemAddress,: Internal memory address
pBuffer,: Pointer to data buffer
BufferSize,: Amount of data to be sent

Return values:

HAL status

Definition at line 803 of file [stm32746g_discovery.c](#).

References [hi2cExtHandler](#), and [I2Cx_WriteMultiple\(\)](#).

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

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Functions		
STM32746G_DISCOVERY_LCD Exported Functions			
STM32746G_DISCOVERY_LCD			

Functions

uint8_t **BSP_LCD_Init** (void)

Initializes the LCD.

uint8_t **BSP_LCD_DelInit** (void)

DeInitializes the LCD.

uint32_t **BSP_LCD_GetXSize** (void)

Gets the LCD X size.

uint32_t **BSP_LCD_GetYSize** (void)

Gets the LCD Y size.

void **BSP_LCD_SetXSize** (uint32_t imageWidthPixels)

Set the LCD X size.

void **BSP_LCD_SetYSize** (uint32_t imageHeightPixels)

Set the LCD Y size.

void **BSP_LCD_LayerDefaultInit** (uint16_t LayerIndex,
uint32_t FB_Address)

Initializes the LCD layer in ARGB8888 format (32 bits per pixel).

void **BSP_LCD_LayerRgb565Init** (uint16_t LayerIndex,
uint32_t FB_Address)

Initializes the LCD layer in RGB565 format (16 bits per pixel).

void **BSP_LCD_SelectLayer** (uint32_t LayerIndex)

Selects the LCD Layer.

void **BSP_LCD_SetLayerVisible** (uint32_t LayerIndex,
FunctionalState State)

Sets an LCD Layer visible.

void **BSP_LCD_SetLayerVisible_NoReload** (uint32_t
LayerIndex, FunctionalState State)

Sets an LCD Layer visible without reloading.

void **BSP_LCD_SetTransparency** (uint32_t LayerIndex,
uint8_t Transparency)

Configures the transparency.

void	BSP_LCD_SetTransparency_NoReload (uint32_t LayerIndex, uint8_t Transparency)	Configures the transparency without reloading.
void	BSP_LCD_SetLayerAddress (uint32_t LayerIndex, uint32_t Address)	Sets an LCD layer frame buffer address.
void	BSP_LCD_SetLayerAddress_NoReload (uint32_t LayerIndex, uint32_t Address)	Sets an LCD layer frame buffer address without reloading.
void	BSP_LCD_SetLayerWindow (uint16_t LayerIndex, uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height)	Sets display window.
void	BSP_LCD_SetLayerWindow_NoReload (uint16_t LayerIndex, uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height)	Sets display window without reloading.
void	BSP_LCD_SetColorKeying (uint32_t LayerIndex, uint32_t RGBValue)	Configures and sets the color keying.
void	BSP_LCD_SetColorKeying_NoReload (uint32_t LayerIndex, uint32_t RGBValue)	Configures and sets the color keying without reloading.
void	BSP_LCD_ResetColorKeying (uint32_t LayerIndex)	Disables the color keying.
void	BSP_LCD_ResetColorKeying_NoReload (uint32_t LayerIndex)	Disables the color keying without reloading.
void	BSP_LCD_Reload (uint32_t ReloadType)	Disables the color keying without reloading.
void	BSP_LCD_SetTextColor (uint32_t Color)	Sets the LCD text color.
uint32_t	BSP_LCD_GetTextColor (void)	

	Gets the LCD text color.
void	BSP_LCD_SetBackColor (uint32_t Color) Sets the LCD background color.
uint32_t	BSP_LCD_GetBackColor (void) Gets the LCD background color.
void	BSP_LCD_SetFont (sFONT *fonts) Sets the LCD text font.
sFONT *	BSP_LCD_GetFont (void) Gets the LCD text font.
uint32_t	BSP_LCD_ReadPixel (uint16_t Xpos, uint16_t Ypos) Reads an LCD pixel.
void	BSP_LCD_Clear (uint32_t Color) Clears the hole LCD.
void	BSP_LCD_ClearStringLine (uint32_t Line) Clears the selected line.
void	BSP_LCD_DisplayChar (uint16_t Xpos, uint16_t Ypos, uint8_t Ascii) Displays one character.
void	BSP_LCD_DisplayStringAt (uint16_t Xpos, uint16_t Ypos, uint8_t *Text, Text_AlignModeTypdef Mode) Displays characters on the LCD.
void	BSP_LCD_DisplayStringAtLine (uint16_t Line, uint8_t *ptr) Displays a maximum of 60 characters on the LCD.
void	BSP_LCD_DrawHLine (uint16_t Xpos, uint16_t Ypos, uint16_t Length) Draws an horizontal line.
void	BSP_LCD_DrawVLine (uint16_t Xpos, uint16_t Ypos, uint16_t Length) Draws a vertical line.
void	BSP_LCD_DrawLine (uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2) Draws an uni-line (between two points).
	BSP_LCD_DrawRect (uint16_t Xpos, uint16_t Ypos,

void	BSP_LCD_DrawRect (uint16_t Width, uint16_t Height)	Draws a rectangle.
void	BSP_LCD_DrawCircle (uint16_t Xpos, uint16_t Ypos, uint16_t Radius)	Draws a circle.
void	BSP_LCD_DrawPolygon (pPoint Points, uint16_t PointCount)	Draws an poly-line (between many points).
void	BSP_LCD_DrawEllipse (int Xpos, int Ypos, int XRadius, int YRadius)	Draws an ellipse on LCD.
void	BSP_LCD_DrawPixel (uint16_t Xpos, uint16_t Ypos, uint32_t RGB_Code)	Draws a pixel on LCD.
void	BSP_LCD_DrawBitmap (uint32_t Xpos, uint32_t Ypos, uint8_t *pbmp)	Draws a bitmap picture loaded in the internal Flash in ARGB888 format (32 bits per pixel).
void	BSP_LCD_FillRect (uint16_t Xpos, uint16_t Ypos, uint16_t Width, uint16_t Height)	Draws a full rectangle.
void	BSP_LCD_FillCircle (uint16_t Xpos, uint16_t Ypos, uint16_t Radius)	Draws a full circle.
void	BSP_LCD_FillPolygon (pPoint Points, uint16_t PointCount)	Draws a full poly-line (between many points).
void	BSP_LCD_FillEllipse (int Xpos, int Ypos, int XRadius, int YRadius)	Draws a full ellipse.
void	BSP_LCD_DisplayOn (void)	Enables the display.
void	BSP_LCD_DisplayOff (void)	Disables the display.

__weak void	BSP_LCD_MspInit (LTDC_HandleTypeDef *hltcd, void *Params)
	Initializes the LTDC MSP.
__weak void	BSP_LCD_MspDeInit (LTDC_HandleTypeDef *hltcd, void *Params)
	DeInitializes BSP_LCD MSP.
__weak void	BSP_LCD_ClockConfig (LTDC_HandleTypeDef *hltcd, void *Params)
	Clock Config.
static void	DrawChar (uint16_t Xpos, uint16_t Ypos, const uint8_t *c)
	Draws a character on LCD.
static void	FillTriangle (uint16_t x1, uint16_t x2, uint16_t x3, uint16_t y1, uint16_t y2, uint16_t y3)
	Fills a triangle (between 3 points).
static void	LL_FillBuffer (uint32_t LayerIndex, void *pDst, uint32_t xSize, uint32_t ySize, uint32_t OffLine, uint32_t ColorIndex)
	Fills a buffer.
static void	LL_ConvertLineToARGB8888 (void *pSrc, void *pDst, uint32_t xSize, uint32_t ColorMode)
	Converts a line to an ARGB8888 pixel format.

Function Documentation

void [BSP_LCD_Clear](#)(uint32_t Color)

Clears the hole LCD.

Parameters:

Color,: Color of the background

Return values:

None

Definition at line [627](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_GetXSize\(\)](#),
[BSP_LCD_GetYSize\(\)](#), [hLtdcHandler](#), and [LL_FillBuffer\(\)](#).

void [BSP_LCD_ClearStringLine](#)(uint32_t Line)

Clears the selected line.

Parameters:

Line,: Line to be cleared

Return values:

None

Definition at line [638](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [LCD_DrawPropTypeDef::BackColor](#),
[BSP_LCD_FillRect\(\)](#), [BSP_LCD_GetXSize\(\)](#),
[BSP_LCD_SetTextColor\(\)](#), and
[LCD_DrawPropTypeDef::TextColor](#).

```
void BSP_LCD_ClockConfig ( LTDC_HandleTypeDef * hltc,  
                           void * Params  
                         )
```

Clock Config.

Parameters:

hltc,: LTDC handle

Params

Note:

This API is called by **BSP_LCD_Init()** Being __weak it can be overwritten by the application

Return values:

None

Definition at line **1402** of file **stm32746g_discovery_lcd.c**.

Referenced by **BSP_LCD_Init()**.

```
uint8_t BSP_LCD_DelInit ( void )
```

DeInitializes the LCD.

Return values:

LCD state

Definition at line **210** of file **stm32746g_discovery_lcd.c**.

References **BSP_LCD_MspDelInit()**, **hLtdcHandler**, and **LCD_OK**.

```
void BSP_LCD_DisplayChar ( uint16_t Xpos,  
                           uint16_t Ypos,  
                           uint8_t Ascii
```

)

Displays one character.

Parameters:

Xpos,: Start column address

Ypos,: Line where to display the character shape.

Ascii,: Character ascii code This parameter must be a number between Min_Data = 0x20 and Max_Data = 0x7E

Return values:

None

Definition at line [658](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [DrawChar\(\)](#), and
[LCD_DrawPropTypeDef::pFont](#).

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

void [BSP_LCD_DisplayOff](#)(void)

Disables the display.

Return values:

None

Definition at line [1267](#) of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#), [LCD_BL_CTRL_GPIO_PORT](#),
[LCD_BL_CTRL_PIN](#), [LCD_DISP_GPIO_PORT](#), and
[LCD_DISP_PIN](#).

void [BSP_LCD_DisplayOn](#)(void)

Enables the display.

Return values:

None

Definition at line [1255](#) of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#), [LCD_BL_CTRL_GPIO_PORT](#), [LCD_BL_CTRL_PIN](#), [LCD_DISP_GPIO_PORT](#), and [LCD_DISP_PIN](#).

```
void BSP_LCD_DisplayStringAt( uint16_t Xpos,  
                             uint16_t Ypos,  
                             uint8_t * Text,  
                             Text_AlignModeTypdef Mode  
 )
```

Displays characters on the LCD.

Parameters:

Xpos,: X position (in pixel)

Ypos,: Y position (in pixel)

Text,: Pointer to string to display on LCD

Mode,: Display mode This parameter can be one of the following values:

- [CENTER_MODE](#)
- [RIGHT_MODE](#)
- [LEFT_MODE](#)

Return values:

None

Definition at line [676](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_DisplayChar\(\)](#),

BSP_LCD_GetXSize(), **CENTER_MODE**, **LEFT_MODE**, **LCD_DrawPropTypeDef::pFont**, and **RIGHT_MODE**.

Referenced by **BSP_LCD_DisplayStringAtLine()**.

```
void BSP_LCD_DisplayStringAtLine( uint16_t Line,  
                                  uint8_t * ptr  
                                )
```

Displays a maximum of 60 characters on the LCD.

Parameters:

Line,: Line where to display the character shape
ptr,: Pointer to string to display on LCD

Return values:

None

Definition at line 737 of file **stm32746g_discovery_lcd.c**.

References **BSP_LCD_DisplayStringAt()**, and **LEFT_MODE**.

```
void BSP_LCD_DrawBitmap( uint32_t Xpos,  
                        uint32_t Ypos,  
                        uint8_t * pbmp  
                      )
```

Draws a bitmap picture loaded in the internal Flash in ARGB888 format (32 bits per pixel).

Parameters:

Xpos,: Bmp X position in the LCD
Ypos,: Bmp Y position in the LCD
pbmp,: Pointer to Bmp picture address in the internal Flash

Return values:

None

Definition at line 1021 of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_GetXSize\(\)](#), [hLtdcHandler](#), and [LL_ConvertLineToARGB8888\(\)](#).

```
void BSP_LCD_DrawCircle( uint16_t Xpos,  
                         uint16_t Ypos,  
                         uint16_t Radius  
)
```

Draws a circle.

Parameters:

Xpos,: X position
Ypos,: Y position
Radius,: Circle radius

Return values:

None

Definition at line 893 of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [BSP_LCD_DrawPixel\(\)](#).

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

```
void BSP_LCD_DrawEllipse( int Xpos,  
                          int Ypos,  
                          int XRadius,  
                          int YRadius  
)
```

Draws an ellipse on LCD.

Parameters:

Xpos,: X position
Ypos,: Y position
XRadius,: Ellipse X radius
YRadius,: Ellipse Y radius

Return values:

None

Definition at line [968](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [BSP_LCD_DrawPixel\(\)](#).

```
void BSP_LCD_DrawHLine ( uint16_t Xpos,  
                         uint16_t Ypos,  
                         uint16_t Length  
 )
```

Draws an horizontal line.

Parameters:

Xpos,: X position
Ypos,: Y position
Length,: Line length

Return values:

None

Definition at line [749](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_GetXSize\(\)](#), [hLtdcHandler](#), and [LL_FillBuffer\(\)](#).

Referenced by [BSP_LCD_DrawRect\(\)](#), [BSP_LCD_FillCircle\(\)](#), and [BSP_LCD_FillEllipse\(\)](#).

```
void BSP_LCD_DrawLine( uint16_t x1,  
                      uint16_t y1,  
                      uint16_t x2,  
                      uint16_t y2  
)
```

Draws an uni-line (between two points).

Parameters:

x1,: Point 1 X position
y1,: Point 1 Y position
x2,: Point 2 X position
y2,: Point 2 Y position

Return values:

None

Definition at line 800 of file [stm32746g_discovery_lcd.c](#).

References [ABS](#), [ActiveLayer](#), and [BSP_LCD_DrawPixel\(\)](#).

Referenced by [BSP_LCD_DrawPolygon\(\)](#), and [FillTriangle\(\)](#).

```
void BSP_LCD_DrawPixel( uint16_t Xpos,  
                       uint16_t Ypos,  
                       uint32_t RGB_Code  
)
```

Draws a pixel on LCD.

Parameters:

Xpos,: X position
Ypos,: Y position
RGB_Code,: Pixel color in ARGB mode (8-8-8-8)

Return values:

None

Definition at line [1001](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_GetXSize\(\)](#), and [hLtdcHandler](#).

Referenced by [BSP_LCD_DrawCircle\(\)](#), [BSP_LCD_DrawEllipse\(\)](#), [BSP_LCD_DrawLine\(\)](#), and [DrawChar\(\)](#).

```
void BSP_LCD_DrawPolygon( pPoint Points,
                           uint16_t PointCount
                         )
```

Draws an poly-line (between many points).

Parameters:

Points,: Pointer to the points array
PointCount,: Number of points

Return values:

None

Definition at line [940](#) of file [stm32746g_discovery_lcd.c](#).

References [BSP_LCD_DrawLine\(\)](#), [Point::X](#), and [Point::Y](#).

```
void BSP_LCD_DrawRect( uint16_t Xpos,
                        uint16_t Ypos,
                        uint16_t Width,
```

```
    uint16_t Height  
)  
}
```

Draws a rectangle.

Parameters:

Xpos,: X position
Ypos,: Y position
Width,: Rectangle width
Height,: Rectangle height

Return values:

None

Definition at line 875 of file [stm32746g_discovery_lcd.c](#).

References [BSP_LCD_DrawHLine\(\)](#), and [BSP_LCD_DrawVLine\(\)](#).

```
void BSP_LCD_DrawVLine ( uint16_t Xpos,  
                         uint16_t Ypos,  
                         uint16_t Length  
)
```

Draws a vertical line.

Parameters:

Xpos,: X position
Ypos,: Y position
Length,: Line length

Return values:

None

Definition at line 774 of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_GetXSize\(\)](#), [hLtdcHandler](#), and [LL_FillBuffer\(\)](#).

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

```
void BSP_LCD_FillCircle ( uint16_t Xpos,  
                          uint16_t Ypos,  
                          uint16_t Radius  
)
```

Draws a full circle.

Parameters:

Xpos,: X position
Ypos,: Y position
Radius,: Circle radius

Return values:

None

Definition at line [1109](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_DrawCircle\(\)](#), [BSP_LCD_DrawHLine\(\)](#), and [BSP_LCD_SetTextColor\(\)](#).

```
void BSP_LCD_FillEllipse ( int Xpos,  
                           int Ypos,  
                           int XRadius,  
                           int YRadius  
)
```

Draws a full ellipse.

Parameters:

Xpos,: X position
Ypos,: Y position
XRadius,: Ellipse X radius
YRadius,: Ellipse Y radius

Return values:

None

Definition at line [1225](#) of file [stm32746g_discovery_lcd.c](#).

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

```
void BSP_LCD_FillPolygon ( pPoint Points,
                           uint16_t PointCount
                         )
```

Draws a full poly-line (between many points).

Parameters:

Points,: Pointer to the points array
PointCount,: Number of points

Return values:

None

Definition at line [1157](#) of file [stm32746g_discovery_lcd.c](#).

References [FillTriangle\(\)](#), [POLY_X](#), [POLY_Y](#), [Point::X](#), and [Point::Y](#).

```
void BSP_LCD_FillRect ( uint16_t Xpos,
                        uint16_t Ypos,
                        uint16_t Width,
                        uint16_t Height
```

)

Draws a full rectangle.

Parameters:

- Xpos,: X position**
- Ypos,: Y position**
- Width,: Rectangle width**
- Height,: Rectangle height**

Return values:

None

Definition at line [1082](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_GetXSize\(\)](#),
[BSP_LCD_SetTextColor\(\)](#), [hLtdcHandler](#), and [LL_FillBuffer\(\)](#).

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

uint32_t BSP_LCD_GetBackColor(void)

Gets the LCD background color.

Return values:

Used background colour

Definition at line [562](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [LCD_DrawPropTypeDef::BackColor](#).

sFONT * BSP_LCD_GetFont(void)

Gets the LCD text font.

Return values:

Used layer font

Definition at line [581](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [LCD_DrawPropTypeDef::pFont](#).

uint32_t BSP_LCD_GetTextColor(void)

Gets the LCD text color.

Return values:

Used text color.

Definition at line [543](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [LCD_DrawPropTypeDef::TextColor](#).

uint32_t BSP_LCD_GetXSize(void)

Gets the LCD X size.

Return values:

Used LCD X size

Definition at line [231](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [hLtdcHandler](#).

Referenced by [BSP_LCD_Clear\(\)](#), [BSP_LCD_ClearStringLine\(\)](#),
[BSP_LCD_DisplayStringAt\(\)](#), [BSP_LCD_DrawBitmap\(\)](#),
[BSP_LCD_DrawHLine\(\)](#), [BSP_LCD_DrawPixel\(\)](#),
[BSP_LCD_DrawVLine\(\)](#), [BSP_LCD_FillRect\(\)](#),
[BSP_LCD_LayerDefaultInit\(\)](#), [BSP_LCD_LayerRgb565Init\(\)](#), and
[BSP_LCD_ReadPixel\(\)](#).

`uint32_t BSP_LCD_GetYSize(void)`

Gets the LCD Y size.

Return values:

Used LCD Y size

Definition at line [240](#) of file `stm32746g_discovery_lcd.c`.

References `ActiveLayer`, and `hLtdcHandler`.

Referenced by `BSP_LCD_Clear()`, `BSP_LCD_LayerDefaultInit()`, and `BSP_LCD_LayerRgb565Init()`.

`uint8_t BSP_LCD_Init(void)`

Initializes the LCD.

Return values:

LCD state

Definition at line [148](#) of file `stm32746g_discovery_lcd.c`.

References `BSP_LCD_ClockConfig()`, `BSP_LCD_MspInit()`, `BSP_LCD_SetFont()`, `BSP_SDRAM_Init()`, `hLtdcHandler`, `LCD_BL_CTRL_GPIO_PORT`, `LCD_BL_CTRL_PIN`, `LCD_DEFAULT_FONT`, `LCD_DISP_GPIO_PORT`, `LCD_DISP_PIN`, and `LCD_OK`.

`void BSP_LCD_LayerDefaultInit(uint16_t LayerIndex, uint32_t FB_Address)`

Initializes the LCD layer in ARGB8888 format (32 bits per pixel).

Parameters:

LayerIndex,: Layer foreground or background

FB_Address,: Layer frame buffer

Return values:

None

Definition at line 271 of file [stm32746g_discovery_lcd.c](#).

References [LCD_DrawPropTypeDef::BackColor](#),
[BSP_LCDGetXSize\(\)](#), [BSP_LCDGetYSize\(\)](#), [hLtdcHandler](#),
[LCD_COLOR_BLACK](#), [LCD_COLOR_WHITE](#),
[LCD_LayerCfgTypeDef](#), [LCD_DrawPropTypeDef::pFont](#), and
[LCD_DrawPropTypeDef::TextColor](#).

```
void BSP_LCD_LayerRgb565Init( uint16_t LayerIndex,  
                                uint32_t FB_Address  
                            )
```

Initializes the LCD layer in RGB565 format (16 bits per pixel).

Parameters:

LayerIndex,: Layer foreground or background

FB_Address,: Layer frame buffer

Return values:

None

Definition at line 305 of file [stm32746g_discovery_lcd.c](#).

References [LCD_DrawPropTypeDef::BackColor](#),
[BSP_LCDGetXSize\(\)](#), [BSP_LCDGetYSize\(\)](#), [hLtdcHandler](#),
[LCD_COLOR_BLACK](#), [LCD_COLOR_WHITE](#),
[LCD_LayerCfgTypeDef](#), [LCD_DrawPropTypeDef::pFont](#), and
[LCD_DrawPropTypeDef::TextColor](#).

```
void BSP_LCD_MspDeInit( LTDC_HandleTypeDef * hLcdc,  
                        void *                               Params  
                      )
```

DeInitializes BSP_LCD MSP.

Parameters:

hLcdc,: LTDC handle
Params

Return values:

None

Definition at line [1353](#) of file [stm32746g_discovery_lcd.c](#).

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

```
void BSP_LCD_MspInit( LTDC_HandleTypeDef * hLcdc,  
                      void *                               Params  
                    )
```

Initializes the LTDC MSP.

Parameters:

hLcdc,: LTDC handle
Params

Return values:

None

Definition at line [1281](#) of file [stm32746g_discovery_lcd.c](#).

References [LCD_BL_CTRL_GPIO_CLK_ENABLE](#),
[LCD_BL_CTRL_GPIO_PORT](#), [LCD_BL_CTRL_PIN](#),

`LCD_DISP_GPIO_CLK_ENABLE`, `LCD_DISP_GPIO_PORT`, and `LCD_DISP_PIN`.

Referenced by `BSP_LCD_Init()`.

```
uint32_t BSP_LCD_ReadPixel( uint16_t Xpos,  
                           uint16_t Ypos  
                         )
```

Reads an LCD pixel.

Parameters:

`Xpos`,: X position
`Ypos`,: Y position

Return values:

`RGB` pixel color

Definition at line 592 of file `stm32746g_discovery_lcd.c`.

References `ActiveLayer`, `BSP_LCD_GetXSize()`, and `hLtdcHandler`.

```
void BSP_LCD_Reload( uint32_t ReloadType )
```

Disables the color keying without reloading.

Parameters:

`ReloadType`,: can be one of the following values

- `LCD_RELOAD_IMMEDIATE`
- `LCD_RELOAD_VERTICAL_BLANKING`

Return values:

`None`

Definition at line 524 of file `stm32746g_discovery_lcd.c`.

References [hLtdcHandler](#).

`void BSP_LCD_ResetColorKeying (uint32_t LayerIndex)`

Disables the color keying.

Parameters:

`LayerIndex`,: Layer foreground or background

Return values:

`None`

Definition at line [500](#) of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

`void BSP_LCD_ResetColorKeying_NoReload (uint32_t LayerIndex)`

Disables the color keying without reloading.

Parameters:

`LayerIndex`,: Layer foreground or background

Return values:

`None`

Definition at line [511](#) of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

`void BSP_LCD_SelectLayer (uint32_t LayerIndex)`

Selects the LCD Layer.

Parameters:

LayerIndex,: Layer foreground or background

Return values:

None

Definition at line 338 of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#).

void [BSP_LCD_SetBackColor](#)(uint32_t **Color)**

Sets the LCD background color.

Parameters:

Color,: Layer background color code ARGB(8-8-8-8)

Return values:

None

Definition at line 553 of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [LCD_DrawPropTypeDef::BackColor](#).

void [BSP_LCD_SetColorKeying](#)(uint32_t **LayerIndex,**
uint32_t **RGBValue**
)

Configures and sets the color keying.

Parameters:

LayerIndex,: Layer foreground or background

RGBValue,: Color reference

Return values:

None

Definition at line [475](#) of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

```
void BSP_LCD_SetColorKeying_NoReload ( uint32_t LayerIndex,  
                                      uint32_t RGBValue  
                                    )
```

Configures and sets the color keying without reloading.

Parameters:

LayerIndex,: Layer foreground or background
RGBValue,: Color reference

Return values:

None

Definition at line [488](#) of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

```
void BSP_LCD_SetFont ( sFONT * fonts )
```

Sets the LCD text font.

Parameters:

fonts,: Layer font to be used

Return values:

None

Definition at line [572](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [LCD_DrawPropTypeDef::pFont](#).

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

```
void BSP_LCD_SetLayerAddress ( uint32_t LayerIndex,  
                               uint32_t Address  
                           )
```

Sets an LCD layer frame buffer address.

Parameters:

LayerIndex,: Layer foreground or background

Address,: New LCD frame buffer value

Return values:

None

Definition at line [417](#) of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

```
void BSP_LCD_SetLayerAddress_NoReload ( uint32_t LayerIndex,  
                                       uint32_t Address  
                                     )
```

Sets an LCD layer frame buffer address without reloading.

Parameters:

LayerIndex,: Layer foreground or background

Address,: New LCD frame buffer value

Return values:

None

Definition at line [428](#) of file `stm32746g_discovery_lcd.c`.

References `hLtdcHandler`.

```
void BSP_LCD_SetLayerVisible ( uint32_t LayerIndex,  
                               FunctionalState State  
                           )
```

Sets an LCD Layer visible.

Parameters:

LayerIndex,: Visible Layer

State,: New state of the specified layer This parameter can be one of the following values:

- ENABLE
- DISABLE

Return values:

None

Definition at line [352](#) of file `stm32746g_discovery_lcd.c`.

References `hLtdcHandler`.

```
void BSP_LCD_SetLayerVisible_NoReload ( uint32_t Layer  
                                       FunctionalState State  
                                     )
```

Sets an LCD Layer visible without reloading.

Parameters:

LayerIndex,: Visible Layer

State,: New state of the specified layer This parameter can be one of the following values:

- ENABLE
- DISABLE

Return values:

None

Definition at line 374 of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

```
void BSP\_LCD\_SetLayerWindow( uint16_t LayerIndex,  
                           uint16_t Xpos,  
                           uint16_t Ypos,  
                           uint16_t Width,  
                           uint16_t Height  
)
```

Sets display window.

Parameters:

LayerIndex,: Layer index
Xpos,: LCD X position
Ypos,: LCD Y position
Width,: LCD window width
Height,: LCD window height

Return values:

None

Definition at line 442 of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

```
void BSP\_LCD\_SetLayerWindow\_NoReload( uint16_t LayerIndex,
```

```
        uint16_t Xpos,  
        uint16_t Ypos,  
        uint16_t Width,  
        uint16_t Height  
    )
```

Sets display window without reloading.

Parameters:

LayerIndex ,	Layer index
Xpos ,	LCD X position
Ypos ,	LCD Y position
Width ,	LCD window width
Height ,	LCD window height

Return values:

None

Definition at line 460 of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

void [BSP_LCD_SetTextColor](#)([uint32_t](#) Color)

Sets the LCD text color.

Parameters:

Color,

Text color code ARGB(8-8-8-8)

Return values:

None

Definition at line 534 of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [LCD_DrawPropTypeDef::TextColor](#).

Referenced by [BSP_LCD_ClearStringLine\(\)](#), [BSP_LCD_FillCircle\(\)](#), and [BSP_LCD_FillRect\(\)](#).

```
void BSP_LCD_SetTransparency( uint32_t LayerIndex,  
                             uint8_t Transparency  
                           )
```

Configures the transparency.

Parameters:

LayerIndex,: Layer foreground or background.

Transparency,: Transparency This parameter must be a number between Min_Data = 0x00 and Max_Data = 0xFF

Return values:

None

Definition at line 394 of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

```
void BSP_LCD_SetTransparency_NoReload( uint32_t LayerIndex,  
                                      uint8_t Transparency  
                                    )
```

Configures the transparency without reloading.

Parameters:

LayerIndex,: Layer foreground or background.

Transparency,: Transparency This parameter must be a number between Min_Data = 0x00 and Max_Data = 0xFF

Return values:

None

Definition at line [406](#) of file [stm32746g_discovery_lcd.c](#).

References [hLtdcHandler](#).

void [BSP_LCD_SetXSize](#) ([uint32_t](#) **imageWidthPixels)**

Set the LCD X size.

Parameters:

imageWidthPixels : image width in pixels unit

Return values:

None

Definition at line [250](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [hLtdcHandler](#).

void [BSP_LCD_SetYSize](#) ([uint32_t](#) **imageHeightPixels)**

Set the LCD Y size.

Parameters:

imageHeightPixels : image height in lines unit

Return values:

None

Definition at line [260](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), and [hLtdcHandler](#).

```
static void DrawChar ( uint16_t Xpos,  
                      uint16_t Ypos,  
                      const uint8_t * c  
) [static]
```

Draws a character on LCD.

Parameters:

Xpos,: Line where to display the character shape
Ypos,: Start column address
c,: Pointer to the character data

Return values:

None

Definition at line [1430](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [BSP_LCD_DrawPixel\(\)](#), and [LCD_DrawPropTypeDef::pFont](#).

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

```
static void FillTriangle ( uint16_t x1,  
                          uint16_t x2,  
                          uint16_t x3,  
                          uint16_t y1,  
                          uint16_t y2,  
                          uint16_t y3  
) [static]
```

Fills a triangle (between 3 points).

Parameters:

x1,: Point 1 X position

y1,: Point 1 Y position
x2,: Point 2 X position
y2,: Point 2 Y position
x3,: Point 3 X position
y3,: Point 3 Y position

Return values:

None

Definition at line [1489](#) of file [stm32746g_discovery_lcd.c](#).

References [ABS](#), and [BSP_LCD_DrawLine\(\)](#).

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

```
static void LL_ConvertLineToARGB8888 ( void * pSrc,
                                         void * pDst,
                                         uint32_t xSize,
                                         uint32_t ColorMode
                                       ) [static]
```

Converts a line to an ARGB8888 pixel format.

Parameters:

pSrc,: Pointer to source buffer
pDst,: Output color
xSize,: Buffer width
ColorMode,: Input color mode

Return values:

None

Definition at line [1605](#) of file [stm32746g_discovery_lcd.c](#).

References [hDma2dHandler](#).

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

```
static void LL_FillBuffer ( uint32_t LayerIndex,  
                           void *    pDst,  
                           uint32_t  xSize,  
                           uint32_t  ySize,  
                           uint32_t  OffLine,  
                           uint32_t  ColorIndex  
                         )      [static]
```

Fills a buffer.

Parameters:

LayerIndex,: Layer index
pDst,: Pointer to destination buffer
xSize,: Buffer width
ySize,: Buffer height
OffLine,: Offset
ColorIndex,: Color index

Return values:

None

Definition at line [1567](#) of file [stm32746g_discovery_lcd.c](#).

References [ActiveLayer](#), [hDma2dHandler](#), and [hLtdcHandler](#).

Referenced by [BSP_LCD_Clear\(\)](#), [BSP_LCD_DrawHLine\(\)](#),
[BSP_LCD_DrawVLine\(\)](#), and [BSP_LCD_FillRect\(\)](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Functions		
STM32746G_DISCOVERY QSPI Exported Functions			
STM32746G-Discovery QSPI			

Functions

uint8_t **BSP_QSPI_Init** (void)

Initializes the QSPI interface.

uint8_t **BSP_QSPI_DelInit** (void)

De-Initializes the QSPI interface.

uint8_t **BSP_QSPI_Read** (uint8_t *pData, uint32_t ReadAddr,
uint32_t Size)

Reads an amount of data from the QSPI memory.

uint8_t **BSP_QSPI_Write** (uint8_t *pData, uint32_t WriteAddr,
uint32_t Size)

Writes an amount of data to the QSPI memory.

uint8_t **BSP_QSPI_Erase_Block** (uint32_t BlockAddress)

Erases the specified block of the QSPI memory.

uint8_t **BSP_QSPI_Erase_Chip** (void)

Erases the entire QSPI memory.

uint8_t **BSP_QSPI_GetStatus** (void)

Reads current status of the QSPI memory.

uint8_t **BSP_QSPI_GetInfo** (QSPI_Info *pInfo)

Return the configuration of the QSPI memory.

uint8_t **BSP_QSPI_EnableMemoryMappedMode** (void)

Configure the QSPI in memory-mapped mode.

void **BSP_QSPI_MspInit** (QSPI_HandleTypeDef *hqspi, void
*Params)

QSPI MSP Initialization This function configures the hardware resources used in this example:

void **BSP_QSPI_MspDelInit** (QSPI_HandleTypeDef *hqspi, void
*Params)

QSPI MSP De-Initialization This function frees the hardware resources used in this example:

Function Documentation

`uint8_t BSP_QSPI_DelInit(void)`

De-Initializes the QSPI interface.

Return values:

QSPI memory status

Definition at line [165](#) of file `stm32746g_discovery_qspi.c`.

References `BSP_QSPI_MspDelInit()`, `QSPI_ERROR`, `QSPI_OK`, and `QSPIHandle`.

`uint8_t BSP_QSPI_EnableMemoryMappedMode(void)`

Configure the QSPI in memory-mapped mode.

Return values:

QSPI memory status

Definition at line [459](#) of file `stm32746g_discovery_qspi.c`.

References `QSPI_ERROR`, `QSPI_OK`, and `QSPIHandle`.

`uint8_t BSP_QSPI_Erase_Block(uint32_t BlockAddress)`

Erases the specified block of the QSPI memory.

Parameters:

BlockAddress,: Block address to erase

Return values:

QSPI memory status

Definition at line 308 of file `stm32746g_discovery_qspi.c`.

References `QSPI_AutoPollingMemReady()`, `QSPI_ERROR`, `QSPI_OK`, `QSPI_WriteEnable()`, and `QSPIHandle`.

`uint8_t BSP_QSPI_Erase_Chip (void)`

Erases the entire QSPI memory.

Return values:

`QSPI` memory status

Definition at line 350 of file `stm32746g_discovery_qspi.c`.

References `QSPI_AutoPollingMemReady()`, `QSPI_ERROR`, `QSPI_OK`, `QSPI_WriteEnable()`, and `QSPIHandle`.

`uint8_t BSP_QSPI_GetInfo (QSPI_Info * pInfo)`

Return the configuration of the QSPI memory.

Parameters:

`pInfo`,: pointer on the configuration structure

Return values:

`QSPI` memory status

Definition at line 443 of file `stm32746g_discovery_qspi.c`.

References `QSPI_Info::EraseSectorSize`, `QSPI_Info::EraseSectorsNumber`, `QSPI_Info::FlashSize`, `QSPI_Info::ProgPageSize`, `QSPI_Info::ProgPagesNumber`, and `QSPI_OK`.

`uint8_t BSP_QSPI_GetStatus (void)`

Reads current status of the QSPI memory.

Return values:

QSPI memory status

Definition at line [390](#) of file `stm32746g_discovery_qspi.c`.

References `QSPI_BUSY`, `QSPI_ERROR`, `QSPI_OK`, `QSPI_SUSPENDED`, and `QSPIHandle`.

uint8_t BSP_QSPI_Init (void)

Initializes the QSPI interface.

Return values:

QSPI memory status

Definition at line [118](#) of file `stm32746g_discovery_qspi.c`.

References `BSP_QSPI_MspInit()`, `QSPI_DummyCyclesCfg()`, `QSPI_ERROR`, `QSPI_NOT_SUPPORTED`, `QSPI_OK`, `QSPI_ResetMemory()`, and `QSPIHandle`.

QSPI MSP De-Initialization This function frees the hardware resources used in this example:

- Disable the Peripheral's clock
 - Revert GPIO and NVIC configuration to their default state

Return values:

None

Definition at line 570 of file `stm32746g_discovery_qspi.c`.

References `QSPI_CLK_DISABLE`, `QSPI_CLK_GPIO_PORT`, `QSPI_CLK_PIN`, `QSPI_CS_GPIO_PORT`, `QSPI_CS_PIN`, `QSPI_D0_GPIO_PORT`, `QSPI_D0_PIN`, `QSPI_D1_GPIO_PORT`, `QSPI_D1_PIN`, `QSPI_D2_GPIO_PORT`, `QSPI_D2_PIN`, `QSPI_D3_GPIO_PORT`, `QSPI_D3_PIN`, `QSPI_FORCE_RESET`, and `QSPI_RELEASE_RESET`.

Referenced by `BSP_QSPI_DeInit()`.

```
void BSP_QSPI_MspInit ( QSPI_HandleTypeDef * hqspi,  
                        void *                                     Params  
                      )
```

QSPI MSP Initialization This function configures the hardware resources used in this example:

- Peripheral's clock enable
- Peripheral's GPIO Configuration
- NVIC configuration for QSPI interrupt

Return values:

None

Definition at line 504 of file `stm32746g_discovery_qspi.c`.

References `QSPI_CLK_ENABLE`, `QSPI_CLK_GPIO_CLK_ENABLE`, `QSPI_CLK_GPIO_PORT`, `QSPI_CLK_PIN`, `QSPI_CS_GPIO_CLK_ENABLE`, `QSPI_CS_GPIO_PORT`, `QSPI_CS_PIN`, `QSPI_D0_GPIO_CLK_ENABLE`, `QSPI_D0_GPIO_PORT`, `QSPI_D0_PIN`, `QSPI_D1_GPIO_CLK_ENABLE`, `QSPI_D1_GPIO_PORT`, `QSPI_D1_PIN`, `QSPI_D2_GPIO_CLK_ENABLE`, `QSPI_D2_GPIO_PORT`, `QSPI_D2_PIN`, `QSPI_D3_GPIO_CLK_ENABLE`, `QSPI_D3_GPIO_PORT`, `QSPI_D3_PIN`, `QSPI_FORCE_RESET`, and

QSPI_RELEASE_RESET.

Referenced by **BSP_QSPI_Init()**.

```
uint8_t BSP_QSPI_Read ( uint8_t * pData,  
                        uint32_t ReadAddr,  
                        uint32_t Size  
                      )
```

Reads an amount of data from the QSPI memory.

Parameters:

pData,: Pointer to data to be read
ReadAddr,: Read start address
Size,: Size of data to read

Return values:

QSPI memory status

Definition at line 188 of file **stm32746g_discovery_qspi.c**.

References **QSPI_ERROR**, **QSPI_OK**, and **QSPIHandle**.

```
uint8_t BSP_QSPI_Write ( uint8_t * pData,  
                         uint32_t WriteAddr,  
                         uint32_t Size  
                       )
```

Writes an amount of data to the QSPI memory.

Parameters:

pData,: Pointer to data to be written
WriteAddr,: Write start address
Size,: Size of data to write

Return values:

QSPI memory status

Definition at line [234](#) of file [stm32746g_discovery_qspi.c](#).

References [QSPI_AutoPollingMemReady\(\)](#), [QSPI_ERROR](#),
[QSPI_OK](#), [QSPI_WriteEnable\(\)](#), and [QSPIHandle](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Defines

STM32746G_DISCOVERY_QSPI Exported Constants

[STM32746G-Discovery QSPI](#)

Defines

```
#define QSPI_OK ((uint8_t)0x00)
#define QSPI_ERROR ((uint8_t)0x01)
#define QSPI_BUSY ((uint8_t)0x02)
#define QSPI_NOT_SUPPORTED ((uint8_t)0x04)
#define QSPI_SUSPENDED ((uint8_t)0x08)
#define QSPI_CLK_ENABLE() __HAL_RCC_QSPI_CLK_ENABLE()
#define QSPI_CLK_DISABLE() __HAL_RCC_QSPI_CLK_DISABLE()
#define QSPI_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define QSPI_CLK_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
#define QSPI_D0_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define QSPI_D1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define QSPI_D2_GPIO_CLK_ENABLE() __HAL_RCC_GPIOE_CLK_ENABLE()
#define QSPI_D3_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_CLK_ENABLE()
#define QSPI_FORCE_RESET() __HAL_RCC_QSPI_FORCE_RESET()
#define QSPI_RELEASE_RESET() __HAL_RCC_QSPI_RELEASE_RESET()
#define QSPI_CS_PIN GPIO_PIN_6
#define QSPI_CS_GPIO_PORT GPIOB
#define QSPI_CLK_PIN GPIO_PIN_2
#define QSPI_CLK_GPIO_PORT GPIOB
#define QSPI_D0_PIN GPIO_PIN_11
#define QSPI_D0_GPIO_PORT GPIOD
#define QSPI_D1_PIN GPIO_PIN_12
#define QSPI_D1_GPIO_PORT GPIOD
#define QSPI_D2_PIN GPIO_PIN_2
#define QSPI_D2_GPIO_PORT GPIOE
#define QSPI_D3_PIN GPIO_PIN_13
#define QSPI_D3_GPIO_PORT GPIOD
#define QSPI_FLASH_SIZE 23 /* Address bus width to access whole
#define QSPI_PAGE_SIZE 256
#define BSP_QSPI_MemoryMappedMode BSP_QSPI_EnableMem
```

Define Documentation

```
#define BSP_QSPI_MemoryMappedMode  BSP_QSPI_EnableMemoryMap
```

Definition at line [109](#) of file [stm32746g_discovery_qspi.h](#).

```
#define QSPI_BUSY ((uint8_t)0x02)
```

Definition at line [71](#) of file [stm32746g_discovery_qspi.h](#).

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

```
#define QSPI_CLK_DISABLE() __HAL_RCC_QSPI_CLK_DISABLE
```

Definition at line [78](#) of file [stm32746g_discovery_qspi.h](#).

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

```
#define QSPI_CLK_ENABLE() __HAL_RCC_QSPI_CLK_ENABLE
```

Definition at line [77](#) of file [stm32746g_discovery_qspi.h](#).

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

```
#define QSPI_CLK_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB
```

Definition at line [80](#) of file [stm32746g_discovery_qspi.h](#).

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

```
#define QSPI_CLK_GPIO_PORT GPIOB
```

Definition at line [93](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspDeInit()`, and `BSP_QSPI_MspInit()`.

```
#define QSPI_CLK_PIN GPIO_PIN_2
```

Definition at line [92](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspDeInit()`, and `BSP_QSPI_MspInit()`.

```
#define QSPI_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_C
```

Definition at line [79](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspInit()`.

```
#define QSPI_CS_GPIO_PORT GPIOB
```

Definition at line [91](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspDeInit()`, and `BSP_QSPI_MspInit()`.

```
#define QSPI_CS_PIN GPIO_PIN_6
```

Definition at line [90](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspDeInit()`, and `BSP_QSPI_MspInit()`.

```
#define QSPI_D0_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_C
```

Definition at line [81](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspInit()`.

```
#define QSPI_D0_GPIO_PORT GPIOD
```

Definition at line [95](#) of file [stm32746g_discovery_qspi.h](#).

Referenced by [BSP_QSPI_MspDeInit\(\)](#), and [BSP_QSPI_MspInit\(\)](#).

```
#define QSPI_D0_PIN GPIO_PIN_11
```

Definition at line [94](#) of file [stm32746g_discovery_qspi.h](#).

Referenced by [BSP_QSPI_MspDeInit\(\)](#), and [BSP_QSPI_MspInit\(\)](#).

```
#define QSPI_D1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC
```

Definition at line [82](#) of file [stm32746g_discovery_qspi.h](#).

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

```
#define QSPI_D1_GPIO_PORT GPIOD
```

Definition at line [97](#) of file [stm32746g_discovery_qspi.h](#).

Referenced by [BSP_QSPI_MspDeInit\(\)](#), and [BSP_QSPI_MspInit\(\)](#).

```
#define QSPI_D1_PIN GPIO_PIN_12
```

Definition at line [96](#) of file [stm32746g_discovery_qspi.h](#).

Referenced by [BSP_QSPI_MspDeInit\(\)](#), and [BSP_QSPI_MspInit\(\)](#).

```
#define QSPI_D2_GPIO_CLK_ENABLE() __HAL_RCC_GPIOE
```

Definition at line [83](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspInit()`.

```
#define QSPI_D2_GPIO_PORT GPIOE
```

Definition at line [99](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspDeInit()`, and `BSP_QSPI_MspInit()`.

```
#define QSPI_D2_PIN GPIO_PIN_2
```

Definition at line [98](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspDeInit()`, and `BSP_QSPI_MspInit()`.

```
#define QSPI_D3_GPIO_CLK_ENABLE() __HAL_RCC_GPIOD_C
```

Definition at line [84](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspInit()`.

```
#define QSPI_D3_GPIO_PORT GPIOD
```

Definition at line [101](#) of file `stm32746g_discovery_qspi.h`.

Referenced by `BSP_QSPI_MspDeInit()`, and `BSP_QSPI_MspInit()`.

```
#define QSPI_D3_PIN GPIO_PIN_13
```

Definition at line [100](#) of file `stm32746g_discovery_qspi.h`.

Referenced by [BSP_QSPI_MspDeInit\(\)](#), and [BSP_QSPI_MspInit\(\)](#).

#define QSPI_ERROR ((uint8_t)0x01)

Definition at line [70](#) of file [stm32746g_discovery_qspi.h](#).

Referenced by [BSP_QSPI_DeInit\(\)](#),
[BSP_QSPI_EnableMemoryMappedMode\(\)](#),
[BSP_QSPI_Erase_Block\(\)](#), [BSP_QSPI_Erase_Chip\(\)](#),
[BSP_QSPI_GetStatus\(\)](#), [BSP_QSPI_Init\(\)](#), [BSP_QSPI_Read\(\)](#),
[BSP_QSPI_Write\(\)](#), [QSPI_AutoPollingMemReady\(\)](#),
[QSPI_DummyCyclesCfg\(\)](#), [QSPI_ResetMemory\(\)](#), and
[QSPI_WriteEnable\(\)](#).

#define QSPI_FLASH_SIZE 23 /* Address bus width to access whole flash */

Definition at line [105](#) of file [stm32746g_discovery_qspi.h](#).

#define QSPI_FORCE_RESET() __HAL_RCC_QSPI_FORCE_RESET()

Definition at line [86](#) of file [stm32746g_discovery_qspi.h](#).

Referenced by [BSP_QSPI_MspDeInit\(\)](#), and [BSP_QSPI_MspInit\(\)](#).

#define QSPI_NOT_SUPPORTED ((uint8_t)0x04)

Definition at line [72](#) of file [stm32746g_discovery_qspi.h](#).

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

#define QSPI_OK ((uint8_t)0x00)

Definition at line [69](#) of file [stm32746g_discovery_qspi.h](#).

Referenced by [BSP_QSPI_DelInit\(\)](#),
[BSP_QSPI_EnableMemoryMappedMode\(\)](#),
[BSP_QSPI_Erase_Block\(\)](#), [BSP_QSPI_Erase_Chip\(\)](#),
[BSP_QSPI_GetInfo\(\)](#), [BSP_QSPI_GetStatus\(\)](#), [BSP_QSPI_Init\(\)](#),
[BSP_QSPI_Read\(\)](#), [BSP_QSPI_Write\(\)](#),
[QSPI_AutoPollingMemReady\(\)](#), [QSPI_DummyCyclesCfg\(\)](#),
[QSPI_ResetMemory\(\)](#), and [QSPI_WriteEnable\(\)](#).

#define QSPI_PAGE_SIZE 256

Definition at line [106](#) of file [stm32746g_discovery_qspi.h](#).

#define QSPI_RELEASE_RESET() __HAL_RCC_QSPI_RELEASE

Definition at line [87](#) of file [stm32746g_discovery_qspi.h](#).

Referenced by [BSP_QSPI_MspDelInit\(\)](#), and [BSP_QSPI_MspInit\(\)](#).

#define QSPI_SUSPENDED ((uint8_t)0x08)

Definition at line [73](#) of file [stm32746g_discovery_qspi.h](#).

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

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Functions		
STM32746G_DISCOVERY QSPI Private Functions			
STM32746G-Discovery QSPI			

Functions

static uint8_t **QSPI_ResetMemory** (QSPI_HandleTypeDef *hqspi)

This function reset the QSPI memory.

static uint8_t **QSPI_DummyCyclesCfg** (QSPI_HandleTypeDef *hqspi)

This function configure the dummy cycles on memory side.

static uint8_t **QSPI_WriteEnable** (QSPI_HandleTypeDef *hqspi)

This function send a Write Enable and wait it is effective.

static uint8_t **QSPI_AutoPollingMemReady** (QSPI_HandleTypeDef *hqspi, uint32_t Timeout)

This function read the SR of the memory and wait the EOP.

__weak void **BSP_QSPI_MspInit** (QSPI_HandleTypeDef *hqspi, void *Params)

QSPI MSP Initialization This function configures the hardware resources used in this example:

__weak void **BSP_QSPI_MspDeInit** (QSPI_HandleTypeDef *hqspi, void *Params)

QSPI MSP De-Initialization This function frees the hardware resources used in this example:

Function Documentation

```
__weak void BSP_QSPI_MspDeInit( QSPI_HandleTypeDef * hqspi,  
                                void *                               Param  
)
```

QSPI MSP De-Initialization This function frees the hardware resources used in this example:

- Disable the Peripheral's clock
- Revert GPIO and NVIC configuration to their default state

Return values:

None

Definition at line 570 of file [stm32746g_discovery_qspi.c](#).

References [QSPI_CLK_DISABLE](#), [QSPI_CLK_GPIO_PORT](#), [QSPI_CLK_PIN](#), [QSPI_CS_GPIO_PORT](#), [QSPI_CS_PIN](#), [QSPI_D0_GPIO_PORT](#), [QSPI_D0_PIN](#), [QSPI_D1_GPIO_PORT](#), [QSPI_D1_PIN](#), [QSPI_D2_GPIO_PORT](#), [QSPI_D2_PIN](#), [QSPI_D3_GPIO_PORT](#), [QSPI_D3_PIN](#), [QSPI_FORCE_RESET](#), and [QSPI_RELEASE_RESET](#).

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

```
__weak void BSP_QSPI_MspInit( QSPI_HandleTypeDef * hqspi,  
                             void *                               Params  
)
```

QSPI MSP Initialization This function configures the hardware resources used in this example:

- Peripheral's clock enable
- Peripheral's GPIO Configuration

- NVIC configuration for QSPI interrupt

Return values:

None

Definition at line 504 of file [stm32746g_discovery_qspi.c](#).

References [QSPI_CLK_ENABLE](#),
[QSPI_CLK_GPIO_CLK_ENABLE](#), [QSPI_CLK_GPIO_PORT](#),
[QSPI_CLK_PIN](#), [QSPI_CS_GPIO_CLK_ENABLE](#),
[QSPI_CS_GPIO_PORT](#), [QSPI_CS_PIN](#),
[QSPI_D0_GPIO_CLK_ENABLE](#), [QSPI_D0_GPIO_PORT](#),
[QSPI_D0_PIN](#), [QSPI_D1_GPIO_CLK_ENABLE](#),
[QSPI_D1_GPIO_PORT](#), [QSPI_D1_PIN](#),
[QSPI_D2_GPIO_CLK_ENABLE](#), [QSPI_D2_GPIO_PORT](#),
[QSPI_D2_PIN](#), [QSPI_D3_GPIO_CLK_ENABLE](#),
[QSPI_D3_GPIO_PORT](#), [QSPI_D3_PIN](#), [QSPI_FORCE_RESET](#), and
[QSPI_RELEASE_RESET](#).

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

```
static uint8_t QSPI_AutoPollingMemReady ( QSPI_HandleTypeDef :  
                                         uint32_t  
                                         )
```

This function read the SR of the memory and wait the EOP.

Parameters:

hqspi: QSPI handle

Timeout

Return values:

None

Definition at line 745 of file [stm32746g_discovery_qspi.c](#).

References [QSPI_ERROR](#), and [QSPI_OK](#).

Referenced by [BSP_QSPI_Erase_Block\(\)](#), [BSP_QSPI_Erase_Chip\(\)](#), [BSP_QSPI_Write\(\)](#), and [QSPI_ResetMemory\(\)](#).

static uint8_t QSPI_DummyCyclesCfg (QSPI_HandleTypeDef * hqs

This function configure the dummy cycles on memory side.

Parameters:

hqspi,: QSPI handle

Return values:

None

Definition at line [640](#) of file [stm32746g_discovery_qspi.c](#).

References [QSPI_ERROR](#), [QSPI_OK](#), and [QSPI_WriteEnable\(\)](#).

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

static uint8_t QSPI_ResetMemory (QSPI_HandleTypeDef * hqspi)

This function reset the QSPI memory.

Parameters:

hqspi,: QSPI handle

Return values:

None

Definition at line [598](#) of file [stm32746g_discovery_qspi.c](#).

References [QSPI_AutoPollingMemReady\(\)](#), [QSPI_ERROR](#), and [QSPI_OK](#).

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

static uint8_t QSPI_WriteEnable (QSPI_HandleTypeDef * hspi) [s]

This function send a Write Enable and wait it is effective.

Parameters:

hspi: QSPI handle

Return values:

None

Definition at line **699** of file **stm32746g_discovery_qspi.c**.

References **QSPI_ERROR**, and **QSPI_OK**.

Referenced by **BSP_QSPI_Erase_Block()**,
BSP_QSPI_Erase_Chip(), **BSP_QSPI_Write()**, and
QSPI_DummyCyclesCfg().

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Functions		
STM32746G_DISCOVERY_SD Exported Functions			
STM32746G_DISCOVERY_SD			

Functions

uint8_t **BSP_SD_Init** (void)

Initializes the SD card device.

uint8_t **BSP_SD_DelInit** (void)

Deinitializes the SD card device.

uint8_t **BSP_SD_ITConfig** (void)

Configures Interrupt mode for SD detection pin.

uint8_t **BSP_SD_IsDetected** (void)

Detects if SD card is correctly plugged in the memory slot or not.

uint8_t **BSP_SD_ReadBlocks** (uint32_t *pData, uint32_t

ReadAddr, uint32_t NumOfBlocks, uint32_t Timeout)

Reads block(s) from a specified address in an SD card, in polling mode.

uint8_t **BSP_SD_WriteBlocks** (uint32_t *pData, uint32_t

WriteAddr, uint32_t NumOfBlocks, uint32_t Timeout)

Writes block(s) to a specified address in an SD card, in polling mode.

uint8_t **BSP_SD_ReadBlocks_DMA** (uint32_t *pData, uint32_t

ReadAddr, uint32_t NumOfBlocks)

Reads block(s) from a specified address in an SD card, in DMA mode.

uint8_t **BSP_SD_WriteBlocks_DMA** (uint32_t *pData, uint32_t

WriteAddr, uint32_t NumOfBlocks)

Writes block(s) to a specified address in an SD card, in DMA mode.

uint8_t **BSP_SD_Erase** (uint32_t StartAddr, uint32_t EndAddr)

Erases the specified memory area of the given SD card.

__weak void **BSP_SD_MspInit** (SD_HandleTypeDef *hsd, void *Params)

Initializes the SD MSP.

<code>__weak void</code>	BSP_SD_Detect_MspInit (SD_HandleTypeDef *hsd, void *Params)	Initializes the SD Detect pin MSP.
<code>__weak void</code>	BSP_SD_MspDeInit (SD_HandleTypeDef *hsd, void *Params)	DeInitializes the SD MSP.
<code>uint8_t</code>	BSP_SD_GetCardState (void)	Gets the current SD card data status.
<code>void</code>	BSP_SD_GetCardInfo (HAL_SD_CardInfoTypeDef *CardInfo)	Get SD information about specific SD card.
<code>void</code>	HAL_SD_AbortCallback (SD_HandleTypeDef *hsd)	SD Abort callbacks.
<code>void</code>	HAL_SD_TxCpltCallback (SD_HandleTypeDef *hsd)	Tx Transfer completed callbacks.
<code>void</code>	HAL_SD_RxCpltCallback (SD_HandleTypeDef *hsd)	Rx Transfer completed callbacks.
<code>__weak void</code>	BSP_SD_AbortCallback (void)	BSP SD Abort callbacks.
<code>__weak void</code>	BSP_SD_WriteCpltCallback (void)	BSP Tx Transfer completed callbacks.
<code>__weak void</code>	BSP_SD_ReadCpltCallback (void)	BSP Rx Transfer completed callbacks.

Function Documentation

void [BSP_SD_AbortCallback](#)(void)

BSP SD Abort callbacks.

Return values:

None

Definition at line [559](#) of file [stm32746g_discovery_sd.c](#).

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

uint8_t [BSP_SD_DelInit](#)(void)

DeInitializes the SD card device.

Return values:

SD status

Definition at line [190](#) of file [stm32746g_discovery_sd.c](#).

References [BSP_SD_MspDelInit\(\)](#), [MSD_ERROR](#), [MSD_OK](#), and [uSdHandle](#).

**void [BSP_SD_Detect_MspInit](#)(SD_HandleTypeDef * hsd,
 Params
)
)**

Initializes the SD Detect pin MSP.

Parameters:

hsd,: SD handle

Params

Return values:

None

Definition at line 451 of file [stm32746g_discovery_sd.c](#).

References [SD_DETECT_GPIO_CLK_ENABLE](#),
[SD_DETECT_GPIO_PORT](#), and [SD_DETECT_PIN](#).

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

```
uint8_t BSP_SD_Erase ( uint32_t StartAddr,  
                      uint32_t EndAddr  
)
```

Erases the specified memory area of the given SD card.

Parameters:

StartAddr,: Start byte address
EndAddr,: End byte address

Return values:

SD status

Definition at line 334 of file [stm32746g_discovery_sd.c](#).

References [MSD_ERROR](#), [MSD_OK](#), and [uSdHandle](#).

```
void BSP_SD_GetCardInfo ( HAL_SD_CardInfoTypeDef * CardInfo )
```

Get SD information about specific SD card.

Parameters:

CardInfo,: Pointer to [HAL_SD_CardInfoTypedef](#) structure

Return values:

None

Definition at line 519 of file [stm32746g_discovery_sd.c](#).

References [uSdHandle](#).

`uint8_t BSP_SD_GetCardState (void)`

Gets the current SD card data status.

Return values:

Data transfer state. This value can be one of the following values:

- SD_TRANSFER_OK: No data transfer is acting
- SD_TRANSFER_BUSY: Data transfer is acting

Definition at line 508 of file [stm32746g_discovery_sd.c](#).

References [SD_TRANSFER_BUSY](#), [SD_TRANSFER_OK](#), and [uSdHandle](#).

`uint8_t BSP_SD_Init (void)`

Initializes the SD card device.

Return values:

SD status

Definition at line 139 of file [stm32746g_discovery_sd.c](#).

References [BSP_SD_Detect_MspInit\(\)](#), [BSP_SD_IsDetected\(\)](#), [BSP_SD_MspInit\(\)](#), [MSD_ERROR](#), [MSD_ERROR_SD_NOT_PRESENT](#), [MSD_OK](#), [SD_PRESENT](#), and [uSdHandle](#).

`uint8_t BSP_SD_IsDetected (void)`

Detects if SD card is correctly plugged in the memory slot or not.

Return values:

Returns if SD is detected or not

Definition at line [235](#) of file `stm32746g_discovery_sd.c`.

References `SD_DETECT_GPIO_PORT`, `SD_DETECT_PIN`, `SD_NOT_PRESENT`, and `SD_PRESENT`.

Referenced by `BSP_SD_Init()`.

`uint8_t BSP_SD_ITConfig (void)`

Configures Interrupt mode for SD detection pin.

Return values:

Returns `MSD_OK`

Definition at line [213](#) of file `stm32746g_discovery_sd.c`.

References `MSD_OK`, `SD_DETECT_EXTI_IRQn`, `SD_DETECT_GPIO_PORT`, and `SD_DETECT_PIN`.

```
void BSP_SD_MspDeInit ( SD_HandleTypeDef * hsd,  
                        void *                                     Params  
                      )
```

DeInitializes the SD MSP.

Parameters:

hsd,: SD handle

Params

Return values:

None

Definition at line 471 of file [stm32746g_discovery_sd.c](#).

References [SD_DMAx_Rx_IRQHandler](#), [SD_DMAx_Rx_STREAM](#), [SD_DMAx_Tx_IRQHandler](#), and [SD_DMAx_Tx_STREAM](#).

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

```
void BSP_SD_MspInit ( SD_HandleTypeDef * hsd,  
                      void *                  Params  
                    )
```

Initializes the SD MSP.

Parameters:

hsd,: SD handle
Params

Return values:

None

Definition at line 352 of file [stm32746g_discovery_sd.c](#).

References [__DMAx_TxRx_CLK_ENABLE](#), [SD_DMAx_Rx_CHANNEL](#), [SD_DMAx_Rx_IRQHandler](#), [SD_DMAx_Rx_STREAM](#), [SD_DMAx_Tx_CHANNEL](#), [SD_DMAx_Tx_IRQHandler](#), and [SD_DMAx_Tx_STREAM](#).

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

```
uint8_t BSP_SD_ReadBlocks ( uint32_t * pData,
```

```
        uint32_t ReadAddr,  
        uint32_t NumOfBlocks,  
        uint32_t Timeout  
    )
```

Reads block(s) from a specified address in an SD card, in polling mode.

Parameters:

pData,: Pointer to the buffer that will contain the data to transmit
ReadAddr,: Address from where data is to be read
NumOfBlocks,: Number of SD blocks to read
Timeout,: Timeout for read operation

Return values:

SD status

Definition at line 256 of file [stm32746g_discovery_sd.c](#).

References [MSD_ERROR](#), [MSD_OK](#), and [uSdHandle](#).

```
uint8_t BSP_SD_ReadBlocks_DMA ( uint32_t * pData,  
                                uint32_t ReadAddr,  
                                uint32_t NumOfBlocks  
    )
```

Reads block(s) from a specified address in an SD card, in DMA mode.

Parameters:

pData,: Pointer to the buffer that will contain the data to transmit
ReadAddr,: Address from where data is to be read
NumOfBlocks,: Number of SD blocks to read

Return values:

SD status

Definition at line 295 of file [stm32746g_discovery_sd.c](#).

References [MSD_ERROR](#), [MSD_OK](#), and [uSdHandle](#).

void BSP_SD_ReadCpltCallback (void)

BSP Rx Transfer completed callbacks.

Return values:

None

Definition at line 577 of file [stm32746g_discovery_sd.c](#).

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

```
uint8_t BSP_SD_WriteBlocks ( uint32_t * pData,  
                            uint32_t WriteAddr,  
                            uint32_t NumOfBlocks,  
                            uint32_t Timeout  
)
```

Writes block(s) to a specified address in an SD card, in polling mode.

Parameters:

pData,: Pointer to the buffer that will contain the data to transmit
WriteAddr,: Address from where data is to be written
NumOfBlocks,: Number of SD blocks to write
Timeout,: Timeout for write operation

Return values:

SD status

Definition at line [276](#) of file `stm32746g_discovery_sd.c`.

References `MSD_ERROR`, `MSD_OK`, and `uSdHandle`.

```
uint8_t BSP_SD_WriteBlocks_DMA ( uint32_t * pData,  
                                uint32_t WriteAddr,  
                                uint32_t NumOfBlocks  
)
```

Writes block(s) to a specified address in an SD card, in DMA mode.

Parameters:

pData,: Pointer to the buffer that will contain the data to transmit
WriteAddr,: Address from where data is to be written
NumOfBlocks,: Number of SD blocks to write

Return values:

SD status

Definition at line [315](#) of file `stm32746g_discovery_sd.c`.

References `MSD_ERROR`, `MSD_OK`, and `uSdHandle`.

```
void BSP_SD_WriteCpltCallback ( void )
```

BSP Tx Transfer completed callbacks.

Return values:

None

Definition at line [568](#) of file `stm32746g_discovery_sd.c`.

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

void HAL_SD_AbortCallback (SD_HandleTypeDef * hsd)

SD Abort callbacks.

Parameters:

hsd,: SD handle

Return values:

None

Definition at line [530](#) of file [stm32746g_discovery_sd.c](#).

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

void HAL_SD_RxCpltCallback (SD_HandleTypeDef * hsd)

Rx Transfer completed callbacks.

Parameters:

hsd,: SD handle

Return values:

None

Definition at line [550](#) of file [stm32746g_discovery_sd.c](#).

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

void HAL_SD_TxCpltCallback (SD_HandleTypeDef * hsd)

Tx Transfer completed callbacks.

Parameters:

hsd,: SD handle

Return values:

None

Definition at line [540](#) of file [stm32746g_discovery_sd.c](#).

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

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Defines

STM32746G_DISCOVERY_SD Exported Types

[STM32746G_DISCOVERY_SD](#)

Defines

```
#define BSP_SD_CardInfo HAL_SD_CardInfoTypeDef  
SD Card information structure.
```

Define Documentation

#define BSP_SD_CardInfo HAL_SD_CardInfoTypeDef

SD Card information structure.

Definition at line **69** of file [stm32746g_discovery_sd.h](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Functions
STM32746G_DISCOVERY_SDRAM Exported Functions				
STM32746G_DISCOVERY_SDRAM				

Functions

uint8_t	BSP_SDRAM_Init (void)	Initializes the SDRAM device.
uint8_t	BSP_SDRAM_DeInit (void)	DeInitializes the SDRAM device.
void	BSP_SDRAM_Initialization_sequence (uint32_t RefreshCount)	Programs the SDRAM device.
uint8_t	BSP_SDRAM_ReadData (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Reads an amount of data from the SDRAM memory in polling mode.
uint8_t	BSP_SDRAM_ReadData_DMA (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Reads an amount of data from the SDRAM memory in DMA mode.
uint8_t	BSP_SDRAM_WriteData (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Writes an amount of data to the SDRAM memory in polling mode.
uint8_t	BSP_SDRAM_WriteData_DMA (uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)	Writes an amount of data to the SDRAM memory in DMA mode.
uint8_t	BSP_SDRAM_Sendcmd (FMC_SDRAM_CommandTypeDef *SdramCmd)	Sends command to the SDRAM bank.
__weak void	BSP_SDRAM_MspInit (SDRAM_HandleTypeDef *hsdram, void *Params)	Initializes SDRAM MSP.
__weak void	BSP_SDRAM_MspDeInit (SDRAM_HandleTypeDef *hsdram, void *Params)	DeInitializes SDRAM MSP.

Function Documentation

`uint8_t BSP_SDRAM_DeInit(void)`

DeInitializes the SDRAM device.

Return values:

SDRAM status

Definition at line [188](#) of file `stm32746g_discovery_sdram.c`.

References [BSP_SDRAM_MspDeInit\(\)](#), [SDRAM_ERROR](#), [SDRAM_OK](#), and [sdramHandle](#).

`uint8_t BSP_SDRAM_Init(void)`

Initializes the SDRAM device.

Return values:

SDRAM status

Definition at line [139](#) of file `stm32746g_discovery_sdram.c`.

References [BSP_SDRAM_Initialization_sequence\(\)](#), [BSP_SDRAM_MspInit\(\)](#), [REFRESH_COUNT](#), [SDCLOCK_PERIOD](#), [SDRAM_ERROR](#), [SDRAM_MEMORY_WIDTH](#), [SDRAM_OK](#), [sdramHandle](#), and [Timing](#).

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

`void BSP_SDRAM_Initialization_sequence(uint32_t RefreshCount)`

Programs the SDRAM device.

Parameters:

RefreshCount,: SDRAM refresh counter value

Return values:

None

Definition at line 214 of file [stm32746g_discovery_sdram.c](#).

References [Command](#), [SDRAM_MODEREG_BURST_LENGTH_1](#), [SDRAM_MODEREG_BURST_TYPE_SEQUENTIAL](#), [SDRAM_MODEREG_CAS_LATENCY_2](#), [SDRAM_MODEREG_OPERATING_MODE_STANDARD](#), [SDRAM_MODEREG_WRITEBURST_MODE_SINGLE](#), [SDRAM_TIMEOUT](#), and [sdramHandle](#).

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

```
void BSP_SDRAM_MspDeInit( SDRAM_HandleTypeDef * hsdram,  
                           void *                               Params  
                         )
```

DeInitializes SDRAM MSP.

Parameters:

hsdram,: SDRAM handle
Params

Return values:

None

Definition at line 459 of file [stm32746g_discovery_sdram.c](#).

References [SDRAM_DMAx_IRQn](#), and [SDRAM_DMAx_STREAM](#).

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

```
void BSP_SDRAM_MspInit( SDRAM_HandleTypeDef * hsdrdram,  
                        void * Params  
)
```

Initializes SDRAM MSP.

Parameters:

hsdrdram,: SDRAM handle

Params

Return values:

None

Definition at line 368 of file [stm32746g_discovery_sdram.c](#).

References [__DMAx_CLK_ENABLE](#), [SDRAM_DMAx_CHANNEL](#), [SDRAM_DMAx IRQn](#), and [SDRAM_DMAx_STREAM](#).

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

```
uint8_t BSP_SDRAM_ReadData( uint32_t uwStartAddress,  
                            uint32_t * pData,  
                            uint32_t uwDataSize  
)
```

Reads an amount of data from the SDRAM memory in polling mode.

Parameters:

uwStartAddress,: Read start address

pData,: Pointer to data to be read

uwDataSize,: Size of read data from the memory

Return values:

SDRAM status

Definition at line 276 of file `stm32746g_discovery_sdram.c`.

References `SDRAM_ERROR`, `SDRAM_OK`, and `sdramHandle`.

```
uint8_t BSP_SDRAM_ReadData_DMA ( uint32_t uwStartAddress,  
                                uint32_t * pData,  
                                uint32_t uwDataSize  
                                )
```

Reads an amount of data from the SDRAM memory in DMA mode.

Parameters:

uwStartAddress,: Read start address
pData,: Pointer to data to be read
uwDataSize,: Size of read data from the memory

Return values:

SDRAM status

Definition at line 295 of file `stm32746g_discovery_sdram.c`.

References `SDRAM_ERROR`, `SDRAM_OK`, and `sdramHandle`.

```
uint8_t BSP_SDRAM_Sendcmd ( FMC_SDRAM_CommandTypeDef * SdramCmd )
```

Sends command to the SDRAM bank.

Parameters:

SdramCmd,: Pointer to SDRAM command structure

Return values:

SDRAM status

Definition at line 350 of file `stm32746g_discovery_sdram.c`.

References **SDRAM_ERROR**, **SDRAM_OK**, **SDRAM_TIMEOUT**, and **sdramHandle**.

```
uint8_t BSP_SDRAM_WriteData( uint32_t uwStartAddress,  
                            uint32_t * pData,  
                            uint32_t uwDataSize  
 )
```

Writes an amount of data to the SDRAM memory in polling mode.

Parameters:

uwStartAddress,: Write start address
pData,: Pointer to data to be written
uwDataSize,: Size of written data from the memory

Return values:

SDRAM status

Definition at line 314 of file **stm32746g_discovery_sdram.c**.

References **SDRAM_ERROR**, **SDRAM_OK**, and **sdramHandle**.

```
uint8_t BSP_SDRAM_WriteData_DMA( uint32_t uwStartAddress,  
                                uint32_t * pData,  
                                uint32_t uwDataSize  
 )
```

Writes an amount of data to the SDRAM memory in DMA mode.

Parameters:

uwStartAddress,: Write start address
pData,: Pointer to data to be written
uwDataSize,: Size of written data from the memory

Return values:

SDRAM status

Definition at line 333 of file `stm32746g_discovery_sdram.c`.

References **SDRAM_ERROR**, **SDRAM_OK**, and **sdramHandle**.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
	Modules Functions		

STM32746G_DISCOVERY_TS Exported Functions

[STM32746G_DISCOVERY_TS](#)

Modules

TS Private Functions

Functions

uint8_t **BSP_TS_Init** (uint16_t ts_SizeX, uint16_t ts_SizeY)

Initializes and configures the touch screen functionalities and configures all necessary hardware resources (GPIOs, I2C, clocks..).

uint8_t **BSP_TS_DelInit** (void)

DeInitializes the TouchScreen.

uint8_t **BSP_TS_ITConfig** (void)

Configures and enables the touch screen interrupts.

uint8_t **BSP_TS_ITGetStatus** (void)

Gets the touch screen interrupt status.

uint8_t **BSP_TS_GetState** (**TS_StateTypeDef** *TS_State)

Returns status and positions of the touch screen.

uint8_t **BSP_TS_Get_GestureId** (**TS_StateTypeDef** *TS_State)

Update gesture Id following a touch detected.

void **BSP_TS_ITClear** (void)

Clears all touch screen interrupts.

uint8_t **BSP_TS_ResetTouchData** (**TS_StateTypeDef** *TS_State)

Function used to reset all touch data before a new acquisition of touch information.

Function Documentation

`uint8_t BSP_TS_DelInit(void)`

DeInitializes the TouchScreen.

Return values:

TS state

Definition at line 173 of file `stm32746g_discovery_ts.c`.

References [TS_OK](#).

`uint8_t BSP_TS_Get_GestureId(TS_StateTypeDef * TS_State)`

Update gesture Id following a touch detected.

Parameters:

TS_State,: Pointer to touch screen current state structure

Return values:

TS_OK if all initializations are OK. Other value if error.

Definition at line 339 of file `stm32746g_discovery_ts.c`.

References `GEST_ID_MOVE_DOWN`, `GEST_ID_MOVE_LEFT`,
`GEST_ID_MOVE_RIGHT`, `GEST_ID_MOVE_UP`,
`GEST_ID_NO_GESTURE`, `GEST_ID_ZOOM_IN`,
`GEST_ID_ZOOM_OUT`, `TS_StateTypeDef::gestureId`, `I2cAddress`,
`TS_ERROR`, and [TS_OK](#).

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

`uint8_t BSP_TS_GetState(TS_StateTypeDef * TS_State)`

Returns status and positions of the touch screen.

Parameters:

TS_State,: Pointer to touch screen current state structure

Return values:

TS_OK if all initializations are OK. Other value if error.

Definition at line 219 of file [stm32746g_discovery_ts.c](#).

References [BSP_TS_Get_GestureId\(\)](#), [I2cAddress](#),
[TOUCH_EVENT_CONTACT](#), [TOUCH_EVENT_LIFT_UP](#),
[TOUCH_EVENT_NO_EVT](#), [TOUCH_EVENT_PRESS_DOWN](#),
[TS_StateTypeDef::touchArea](#), [TS_StateTypeDef::touchDetected](#),
[TS_StateTypeDef::touchEventId](#), [TS_StateTypeDef::touchWeight](#),
[TS_StateTypeDef::touchX](#), [TS_StateTypeDef::touchY](#),
[TS_ERROR](#), [TS_MAX_NB_TOUCH](#), [TS_OK](#), [TS_SWAP_NONE](#),
[TS_SWAP_X](#), [TS_SWAP_XY](#), [TS_SWAP_Y](#), [tsDriver](#),
[tsOrientation](#), [tsXBoundary](#), and [tsYBoundary](#).

```
uint8_t BSP_TS_Init( uint16_t ts_SizeX,  
                      uint16_t ts_SizeY  
                    )
```

Initializes and configures the touch screen functionalities and configures all necessary hardware resources (GPIOs, I2C, clocks..).

Parameters:

ts_SizeX,: Maximum X size of the TS area on LCD

ts_SizeY,: Maximum Y size of the TS area on LCD

Return values:

TS_OK if all initializations are OK. Other value if error.

Definition at line 143 of file [stm32746g_discovery_ts.c](#).

References [I2cAddress](#), [TS_DEVICE_NOT_FOUND](#), [TS_I2C_ADDRESS](#), [TS_OK](#), [TS_SWAP_XY](#), [tsDriver](#), [tsOrientation](#), [tsXBoundary](#), and [tsYBoundary](#).

`void BSP_TS_ITClear (void)`

Clears all touch screen interrupts.

Definition at line [383](#) of file [stm32746g_discovery_ts.c](#).

References [I2cAddress](#), and [tsDriver](#).

`uint8_t BSP_TS_ITConfig (void)`

Configures and enables the touch screen interrupts.

Return values:

[TS_OK](#) if all initializations are OK. Other value if error.

Definition at line [183](#) of file [stm32746g_discovery_ts.c](#).

References [I2cAddress](#), [TS_INT_EXTI_IRQn](#), [TS_INT_GPIO_PORT](#), [TS_INT_PIN](#), [TS_OK](#), and [tsDriver](#).

`uint8_t BSP_TS_ITGetStatus (void)`

Gets the touch screen interrupt status.

Return values:

[TS_OK](#) if all initializations are OK. Other value if error.

Definition at line [208](#) of file [stm32746g_discovery_ts.c](#).

References [I2cAddress](#), and [tsDriver](#).

uint8_t BSP_TS_ResetTouchData (TS_StateTypeDef * TS_State)

Function used to reset all touch data before a new acquisition of touch information.

Parameters:

TS_State,: Pointer to touch screen current state structure

Return values:

TS_OK if OK, **TE_ERROR** if problem found.

Definition at line **401** of file **stm32746g_discovery_ts.c**.

References **GEST_ID_NO_GESTURE**,
TS_StateTypeDef::gestureId, **TOUCH_EVENT_NO_EVT**,
TS_StateTypeDef::touchArea, **TS_StateTypeDef::touchDetected**,
TS_StateTypeDef::touchEventId, **TS_StateTypeDef::touchWeight**,
TS_StateTypeDef::touchX, **TS_StateTypeDef::touchY**,
TS_ERROR, **TS_MAX_NB_TOUCH**, and **TS_OK**.

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Functions

TS Private Functions

[STM32746G_DISCOVERY_TS Exported Functions](#)

Functions

uint8_t **BSP_TS_ResetTouchData (TS_StateTypeDef *TS_State)**

Function used to reset all touch data before a new acquisition of touch information.

Function Documentation

`uint8_t BSP_TS_ResetTouchData (TS_StateTypeDef * TS_State)`

Function used to reset all touch data before a new acquisition of touch information.

Parameters:

`TS_State`,: Pointer to touch screen current state structure

Return values:

`TS_OK` if OK, `TE_ERROR` if problem found.

Definition at line 401 of file `stm32746g_discovery_ts.c`.

References `GEST_ID_NO_GESTURE`,
`TS_StateTypeDef::gestureId`, `TOUCH_EVENT_NO_EVT`,
`TS_StateTypeDef::touchArea`, `TS_StateTypeDef::touchDetected`,
`TS_StateTypeDef::touchEventId`, `TS_StateTypeDef::touchWeight`,
`TS_StateTypeDef::touchX`, `TS_StateTypeDef::touchY`,
`TS_ERROR`, `TS_MAX_NB_TOUCH`, and `TS_OK`.

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Variables
STM32746G_DISCOVERY_LOW_LEVEL Private Variables				
STM32746G_DISCOVERY_LOW_LEVEL				

Variables

```
const uint32_t GPIO_PIN [LEDn] = {LED1_PIN}
GPIO_TypeDef * BUTTON_PORT [BUTTONn]
const uint16_t BUTTON_PIN [BUTTONn]
const uint16_t BUTTON IRQn [BUTTONn]
USART_TypeDef * COM_USART [COMn] =
{DISCOVERY_COM1}
GPIO_TypeDef * COM_TX_PORT [COMn] =
{DISCOVERY_COM1_TX_GPIO_PORT}
GPIO_TypeDef * COM_RX_PORT [COMn] =
{DISCOVERY_COM1_RX_GPIO_PORT}
const uint16_t COM_TX_PIN [COMn] =
{DISCOVERY_COM1_TX_PIN}
const uint16_t COM_RX_PIN [COMn] =
{DISCOVERY_COM1_RX_PIN}
const uint16_t COM_TX_AF [COMn] =
{DISCOVERY_COM1_TX_AF}
const uint16_t COM_RX_AF [COMn] =
{DISCOVERY_COM1_RX_AF}
static I2C_HandleTypeDef hI2cAudioHandler = {0}
static I2C_HandleTypeDef hI2cExtHandler = {0}
```

Variable Documentation

const uint16_t BUTTON_IRQn[BUTTONn]

Initial value:

Definition at line [101](#) of file `stm32746g_discovery.c`.

Referenced by [BSP_PB_Delnit\(\)](#), and [BSP_PB_Init\(\)](#).

const uint16_t BUTTON_PIN[BUTTONn]

Initial value:

```
{WAKEUP_BUTTON_PIN,  
    TAMPER_BUTTON  
ON_PIN,  
    KEY_BUTTON_  
PIN}
```

Definition at line 97 of file [stm32746g_discovery.c](#).

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

GPIO_TypeDef* BUTTON_PORT[BUTTONN]

Initial value:

```
{WAKEUP_BUTTON_GPIO_PORT,  
ON_GPIO_PORT,  
TAMPER_BUTTON_GPIO_PORT}
```

KEY_BUTTON_

GPIO_PORT}

Definition at line [93](#) of file `stm32746g_discovery.c`.

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

`const uint16_t COM_RX_AF[COMn] = {DISCOVERY_COM1_RX_AF}`

Definition at line [117](#) of file `stm32746g_discovery.c`.

Referenced by `BSP_COM_Init()`.

`const uint16_t COM_RX_PIN[COMn] = {DISCOVERY_COM1_RX_PII}`

Definition at line [113](#) of file `stm32746g_discovery.c`.

Referenced by `BSP_COM_Init()`.

`GPIO_TypeDef* COM_RX_PORT[COMn] = {DISCOVERY_COM1_RX}`

Definition at line [109](#) of file `stm32746g_discovery.c`.

Referenced by `BSP_COM_Init()`.

`const uint16_t COM_TX_AF[COMn] = {DISCOVERY_COM1_TX_AF}`

Definition at line [115](#) of file `stm32746g_discovery.c`.

Referenced by `BSP_COM_Init()`.

`const uint16_t COM_TX_PIN[COMn] = {DISCOVERY_COM1_TX_PIN}`

Definition at line [111](#) of file `stm32746g_discovery.c`.

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

GPIO_TypeDef* COM_TX_PORT[COMn] = {DISCOVERY_COM1_TX}

Definition at line [107](#) of file `stm32746g_discovery.c`.

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

USART_TypeDef* COM_USART[COMn] = {DISCOVERY_COM1}

Definition at line [105](#) of file `stm32746g_discovery.c`.

Referenced by [BSP_COM_DelInit\(\)](#), and [BSP_COM_Init\(\)](#).

const uint32_t GPIO_PIN[LEDn] = {LED1_PIN}

Definition at line [91](#) of file `stm32746g_discovery.c`.

Referenced by [BSP_LED_DelInit\(\)](#), [BSP_LED_Init\(\)](#),
[BSP_LED_Off\(\)](#), [BSP_LED_On\(\)](#), and [BSP_LED_Toggle\(\)](#).

I2C_HandleTypeDef hI2cAudioHandler = {0} [static]

Definition at line [119](#) of file `stm32746g_discovery.c`.

Referenced by [AUDIO_IO_Init\(\)](#), [AUDIO_IO_Read\(\)](#),
[AUDIO_IO_Write\(\)](#), [I2Cx_Init\(\)](#), [I2Cx_MspInit\(\)](#), [TS_IO_Init\(\)](#),
[TS_IO_Read\(\)](#), and [TS_IO_Write\(\)](#).

I2C_HandleTypeDefDef hI2cExtHandler = {0} [static]

Definition at line [120](#) of file [stm32746g_discovery.c](#).

Referenced by [CAMERA_IO_Init\(\)](#), [CAMERA_IO_Read\(\)](#),
[CAMERA_IO_Write\(\)](#), [EEPROM_IO_Init\(\)](#),
[EEPROM_IO_IsDeviceReady\(\)](#), [EEPROM_IO_ReadData\(\)](#), and
[EEPROM_IO_WriteData\(\)](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Enumerations
STM32746G_DISCOVERY_LOW_LEVEL Exported Types				
<u>STM32746G_DISCOVERY_LOW_LEVEL</u>				

Enumerations

```
enum Led_TypeDef { LED1 = 0, LED_GREEN = LED1 }
```

```
enum Button_TypeDef { BUTTON_WAKEUP = 0,  
                     BUTTON_TAMPER = 1, BUTTON_KEY = 2 }
```

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

```
enum COM_TypeDef { COM1 = 0, COM2 = 1 }
```

Enumeration Type Documentation

enum Button_TypeDef

Enumerator:

BUTTON_WAKEUP
BUTTON_TAMPER
BUTTON_KEY

Definition at line [71](#) of file [stm32746g_discovery.h](#).

enum ButtonMode_TypeDef

Enumerator:

BUTTON_MODE_GPIO
BUTTON_MODE_EXTI

Definition at line [78](#) of file [stm32746g_discovery.h](#).

enum COM_TypeDef

Enumerator:

COM1
COM2

Definition at line [84](#) of file [stm32746g_discovery.h](#).

enum Led_TypeDef

Enumerator:

LED1

LED_GREEN

Definition at line **65** of file [stm32746g_discovery.h](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Defines		
STM32746G_DISCOVERY_LOW_LEVEL_BUTTON			
STM32746G_DISCOVERY_LOW_LEVEL Exported Constants			

Defines

```
#define BUTTONn ((uint8_t)3)
#define WAKEUP_BUTTON_PIN GPIO_PIN_11
Wakeup push-button.

#define WAKEUP_BUTTON_GPIO_PORT GPIOI
#define WAKEUP_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_G
#define WAKEUP_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_G
#define WAKEUP_BUTTON_EXTI_IRQn EXTI15_10_IRQn
#define TAMPER_BUTTON_PIN GPIO_PIN_11
Tamper push-button.

#define TAMPER_BUTTON_GPIO_PORT GPIOI
#define TAMPER_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_G
#define TAMPER_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_G
#define TAMPER_BUTTON_EXTI_IRQn EXTI15_10_IRQn
#define KEY_BUTTON_PIN GPIO_PIN_11
Key push-button.

#define KEY_BUTTON_GPIO_PORT GPIOI
#define KEY_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI
#define KEY_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI
#define KEY_BUTTON_EXTI_IRQn EXTI15_10_IRQn
#define BUTTONx_GPIO_CLK_ENABLE(__INDEX__)
#define BUTTONx_GPIO_CLK_DISABLE(__INDEX__)
```

Define Documentation

```
#define BUTTONn ((uint8_t)3)
```

Definition at line 122 of file [stm32746g_discovery.h](#).

```
#define BUTTONx_GPIO_CLK_DISABLE( __INDEX__ )
```

Value:

```
(((__INDEX__) == 0) ? WAKEUP_BUTTON_GPIO_CLK_DISABLE() :\\
(\\
(__INDEX__) == 1) ? TAMPER_BUTTON_GPIO_CLK_DISABLE() : KEY_BUTTON_GPIO_CLK_DISABLE())
```

Definition at line 155 of file [stm32746g_discovery.h](#).

```
#define BUTTONx_GPIO_CLK_ENABLE( __INDEX__ )
```

Value:

```
do { if((__INDEX__) == 0) WAKEUP_BUTTON_GPIO_CLK_ENABLE(); else\\
     if((__INDEX__) == 1) TAMPER_BUTTON_GPIO_CLK_ENABLE(); else\\
                           KEY_BUTTON_GPIO_CLK_ENABLE(); } while(0)
```

Definition at line 151 of file [stm32746g_discovery.h](#).

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

```
#define KEY_BUTTON_EXTI_IRQn  EXTI15_10_IRQn
```

Definition at line [149](#) of file `stm32746g_discovery.h`.

```
#define KEY_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_GPIO
```

Definition at line [148](#) of file `stm32746g_discovery.h`.

```
#define KEY_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_GPIO
```

Definition at line [147](#) of file `stm32746g_discovery.h`.

```
#define KEY_BUTTON_GPIO_PORT GPIOI
```

Definition at line [146](#) of file `stm32746g_discovery.h`.

```
#define KEY_BUTTON_PIN GPIO_PIN_11
```

Key push-button.

Definition at line [145](#) of file `stm32746g_discovery.h`.

```
#define TAMPER_BUTTON_EXTI_IRQn EXTI15_10_IRQn
```

Definition at line [140](#) of file `stm32746g_discovery.h`.

```
#define TAMPER_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC
```

Definition at line [139](#) of file `stm32746g_discovery.h`.

```
#define TAMPER_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_
```

Definition at line [138](#) of file `stm32746g_discovery.h`.

```
#define TAMPER_BUTTON_GPIO_PORT GPIOI
```

Definition at line [137](#) of file `stm32746g_discovery.h`.

```
#define TAMPER_BUTTON_PIN GPIO_PIN_11
```

Tamper push-button.

Definition at line [136](#) of file `stm32746g_discovery.h`.

```
#define WAKEUP_BUTTON_EXTI_IRQn EXTI15_10_IRQn
```

Definition at line [131](#) of file `stm32746g_discovery.h`.

```
#define WAKEUP_BUTTON_GPIO_CLK_DISABLE() __HAL_RCC_
```

Definition at line [130](#) of file `stm32746g_discovery.h`.

```
#define WAKEUP_BUTTON_GPIO_CLK_ENABLE() __HAL_RCC_
```

Definition at line [129](#) of file `stm32746g_discovery.h`.

```
#define WAKEUP_BUTTON_GPIO_PORT GPIOI
```

Definition at line [128](#) of file `stm32746g_discovery.h`.

```
#define WAKEUP_BUTTON_PIN GPIO_PIN_11
```

Wakeup push-button.

Definition at line [127](#) of file [stm32746g_discovery.h](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Defines				STM32746G_DISCOVERY_CAMERA Private Defines
STM32746G_DISCOVERY_CAMERA				

Defines

```
#define CAMERA_VGA_RES_X 640
#define CAMERA_VGA_RES_Y 480
#define CAMERA_480x272_RES_X 480
#define CAMERA_480x272_RES_Y 272
#define CAMERA_QVGA_RES_X 320
#define CAMERA_QVGA_RES_Y 240
#define CAMERA_QQVGA_RES_X 160
#define CAMERA_QQVGA_RES_Y 120
```

Define Documentation

#define CAMERA_480x272_RES_X 480

Definition at line [91](#) of file [stm32746g_discovery_camera.c](#).

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

#define CAMERA_480x272_RES_Y 272

Definition at line [92](#) of file [stm32746g_discovery_camera.c](#).

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

#define CAMERA_QQVGA_RES_X 160

Definition at line [95](#) of file [stm32746g_discovery_camera.c](#).

#define CAMERA_QQVGA_RES_Y 120

Definition at line [96](#) of file [stm32746g_discovery_camera.c](#).

#define CAMERA_QVGA_RES_X 320

Definition at line [93](#) of file [stm32746g_discovery_camera.c](#).

#define CAMERA_QVGA_RES_Y 240

Definition at line [94](#) of file [stm32746g_discovery_camera.c](#).

#define CAMERA_VGA_RES_X 640

Definition at line **89** of file [stm32746g_discovery_camera.c](#).

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

#define CAMERA_VGA_RES_Y 480

Definition at line **90** of file [stm32746g_discovery_camera.c](#).

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

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories	Variables			
STM32746G_DISCOVERY_CAMERA Private Variables				STM32746G_DISCOVERY_CAMERA

Variables

DCMI_HandleTypeDef	hDcmiHandler
CAMERA_DrvTypeDef *	camera_drv
static uint32_t	CameraCurrentResolution
static uint32_t	CameraHwAddress

Variable Documentation

CAMERA_DrvTypeDef* `camera_drv`

Definition at line [112](#) of file `stm32746g_discovery_camera.c`.

Referenced by `BSP_CAMERA_BlackWhiteConfig()`,
`BSP_CAMERA_ColorEffectConfig()`,
`BSP_CAMERA_ContrastBrightnessConfig()`, and
`BSP_CAMERA_Init()`.

uint32_t `CameraCurrentResolution` **[static]**

Definition at line [114](#) of file `stm32746g_discovery_camera.c`.

Referenced by `BSP_CAMERA_ContinuousStart()`,
`BSP_CAMERA_Init()`, and `BSP_CAMERA_SnapshotStart()`.

uint32_t `CameraHwAddress` **[static]**

Definition at line [117](#) of file `stm32746g_discovery_camera.c`.

Referenced by `BSP_CAMERA_BlackWhiteConfig()`,
`BSP_CAMERA_ColorEffectConfig()`,
`BSP_CAMERA_ContrastBrightnessConfig()`, and
`BSP_CAMERA_Init()`.

DCMI_HandleTypeDef `hDcmiHandler`

Definition at line [111](#) of file `stm32746g_discovery_camera.c`.

Referenced by `BSP_CAMERA_ContinuousStart()`,
`BSP_CAMERA_DelInit()`, `BSP_CAMERA_Init()`,

BSP_CAMERA_Resume(), **BSP_CAMERA_SnapshotStart()**,
BSP_CAMERA_Stop(), and **BSP_CAMERA_Suspend()**.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Defines Enumerations

STM32746G_DISCOVERY_CAMERA Exported Types

[STM32746G_DISCOVERY_CAMERA](#)

Defines

```
#define RESOLUTION_R160x120 CAMERA_R160x120 /* QQVGA  
Resolution */  
#define RESOLUTION_R320x240 CAMERA_R320x240 /* QVGA  
Resolution */  
#define RESOLUTION_R480x272 CAMERA_R480x272 /* 480x272  
Resolution */  
#define RESOLUTION_R640x480 CAMERA_R640x480 /* VGA  
Resolution */
```

Enumerations

```
enum Camera_StatusTypeDef {
    CAMERA_OK = 0x00, CAMERA_ERROR = 0x01,
    CAMERA_TIMEOUT = 0x02, CAMERA_NOT_DETECTED =
    0x03,
    CAMERA_NOT_SUPPORTED = 0x04
}
```

Camera State structures definition. [More...](#)

Define Documentation

```
#define RESOLUTION_R160x120 CAMERA_R160x120 /* QQVGA Resolution
```

Definition at line [81](#) of file [stm32746g_discovery_camera.h](#).

```
#define RESOLUTION_R320x240 CAMERA_R320x240 /* QVGA Resolution
```

Definition at line [82](#) of file [stm32746g_discovery_camera.h](#).

```
#define RESOLUTION_R480x272 CAMERA_R480x272 /* 480x272 FHD Resolution
```

Definition at line [83](#) of file [stm32746g_discovery_camera.h](#).

```
#define RESOLUTION_R640x480 CAMERA_R640x480 /* VGA Resolution
```

Definition at line [84](#) of file [stm32746g_discovery_camera.h](#).

Enumeration Type Documentation

enum Camera_StatusTypeDef

Camera State structures definition.

Enumerator:

- CAMERA_OK*
- CAMERA_ERROR*
- CAMERA_TIMEOUT*
- CAMERA_NOT_DETECTED*
- CAMERA_NOT_SUPPORTED*

Definition at line [71](#) of file [stm32746g_discovery_camera.h](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Variables
STM32746G_DISCOVERY_SDRAM Private Variables				
STM32746G_DISCOVERY_SDRAM				

Variables

SDRAM_HandleTypeDef	sdramHandle
static FMC_SDRAM_TimingTypeDef	Timing
static FMC_SDRAM_CommandTypeDef	Command

Variable Documentation

FMC_SDRAM_CommandTypeDef Command [static]

Definition at line [119](#) of file [stm32746g_discovery_sdram.c](#).

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

SDRAM_HandleTypeDef sdramHandle

Definition at line [117](#) of file [stm32746g_discovery_sdram.c](#).

Referenced by [BSP_SDRAM_DeInit\(\)](#), [BSP_SDRAM_Init\(\)](#),
[BSP_SDRAM_Initialization_sequence\(\)](#),
[BSP_SDRAM_ReadData\(\)](#), [BSP_SDRAM_ReadData_DMA\(\)](#),
[BSP_SDRAM_Sendcmd\(\)](#), [BSP_SDRAM_WriteData\(\)](#), and
[BSP_SDRAM_WriteData_DMA\(\)](#).

FMC_SDRAM_TimingTypeDef Timing [static]

Definition at line [118](#) of file [stm32746g_discovery_sdram.c](#).

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

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Defines				STM32746G_DISCOVERY_AUDIO Exported Macros
STM32746G_DISCOVERY_AUDIO				

Defines

```
#define DMA_MAX(x) (((x) <= DMA_MAX_SZE)?  
                   (x):DMA_MAX_SZE)
```

Define Documentation

`#define DMA_MAX(x) (((x) <= DMA_MAX_SZE)? (x):DMA_MAX_`

Definition at line [203](#) of file `stm32746g_discovery_audio.h`.

Referenced by [`BSP_AUDIO_OUT_Play\(\)`](#).

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
Defines				STM32746G_DISCOVERY_EEPROM Exported Constants
STM32746G_DISCOVERY_EEPROM				

Defines

```
#define EEPROM_PAGESIZE ((uint8_t)4)
#define EEPROM_MAX_SIZE ((uint16_t)0x2000) /* 64Kbit */
#define EEPROM_MAX_TRIALS ((uint32_t)3000)
#define EEPROM_OK ((uint32_t)0)
#define EEPROM_FAIL ((uint32_t)1)
#define EEPROM_TIMEOUT ((uint32_t)2)
```

Define Documentation

```
#define EEPROM_FAIL ((uint32_t)1)
```

Definition at line [82](#) of file [stm32746g_discovery_eeprom.h](#).

Referenced by [BSP_EEPROM_Init\(\)](#), [BSP_EEPROM_ReadBuffer\(\)](#),
and [BSP_EEPROM_WritePage\(\)](#).

```
#define EEPROM_MAX_SIZE ((uint16_t)0x2000) /* 64Kbit */
```

Definition at line [75](#) of file [stm32746g_discovery_eeprom.h](#).

```
#define EEPROM_MAX_TRIALS ((uint32_t)3000)
```

Definition at line [79](#) of file [stm32746g_discovery_eeprom.h](#).

Referenced by [BSP_EEPROM_Init\(\)](#), and
[BSP_EEPROM_WaitEepromStandbyState\(\)](#).

```
#define EEPROM_OK ((uint32_t)0)
```

Definition at line [81](#) of file [stm32746g_discovery_eeprom.h](#).

Referenced by [BSP_EEPROM_DeInit\(\)](#), [BSP_EEPROM_Init\(\)](#),
[BSP_EEPROM_ReadBuffer\(\)](#),
[BSP_EEPROM_WaitEepromStandbyState\(\)](#),
[BSP_EEPROM_WriteBuffer\(\)](#), and [BSP_EEPROM_WritePage\(\)](#).

```
#define EEPROM_PAGESIZE ((uint8_t)4)
```

Definition at line [74](#) of file [stm32746g_discovery_eeprom.h](#).

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

#define EEPROM_TIMEOUT ((uint32_t)2)

Definition at line [83](#) of file [stm32746g_discovery_eeprom.h](#).

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

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories	Variables			
STM32746G_DISCOVERY_EEPROM Private Variables				STM32746G_DISCOVERY_EEPROM

Variables

```
__IO uint16_t EEPROMAddress = 0  
__IO uint16_t EEPROMDataRead  
__IO uint8_t EEPROMDataWrite
```

Variable Documentation

`__IO uint16_t EEPROMAddress = 0`

Definition at line [128](#) of file `stm32746g_discovery_eeprom.c`.

Referenced by `BSP_EEPROM_Init()`, `BSP_EEPROM_ReadBuffer()`,
`BSP_EEPROM_WaitEepromStandbyState()`, and
`BSP_EEPROM_WritePage()`.

`__IO uint16_t EEPROMDataRead`

Definition at line [129](#) of file `stm32746g_discovery_eeprom.c`.

Referenced by `BSP_EEPROM_ReadBuffer()`.

`__IO uint8_t EEPROMDataWrite`

Definition at line [130](#) of file `stm32746g_discovery_eeprom.c`.

Referenced by `BSP_EEPROM_WritePage()`.

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Variables

STM32746G_DISCOVERY_TS Private Variables

[STM32746G_DISCOVERY_TS](#)

Variables

```
static TS_DrvTypeDef * tsDriver
    static uint16_t tsXBoundary
    static uint16_t tsYBoundary
    static uint8_t tsOrientation
    static uint8_t I2cAddress
```

Variable Documentation

uint8_t I2cAddress [static]

Definition at line [120](#) of file [stm32746g_discovery_ts.c](#).

Referenced by [BSP_TS_GetGestureId\(\)](#), [BSP_TS_GetState\(\)](#), [BSP_TS_Init\(\)](#), [BSP_TS_ITClear\(\)](#), [BSP_TS_ITConfig\(\)](#), and [BSP_TS_ITGetStatus\(\)](#).

TS_DrvTypeDef* tsDriver [static]

Definition at line [117](#) of file [stm32746g_discovery_ts.c](#).

Referenced by [BSP_TS_GetState\(\)](#), [BSP_TS_Init\(\)](#), [BSP_TS_ITClear\(\)](#), [BSP_TS_ITConfig\(\)](#), and [BSP_TS_ITGetStatus\(\)](#).

uint8_t tsOrientation [static]

Definition at line [119](#) of file [stm32746g_discovery_ts.c](#).

Referenced by [BSP_TS_GetState\(\)](#), and [BSP_TS_Init\(\)](#).

uint16_t tsXBoundary [static]

Definition at line [118](#) of file [stm32746g_discovery_ts.c](#).

Referenced by [BSP_TS_GetState\(\)](#), and [BSP_TS_Init\(\)](#).

uint16_t tsYBoundary [static]

Definition at line [118](#) of file `stm32746g_discovery_ts.c`.

Referenced by `BSP_TS_GetState()`, and `BSP_TS_Init()`.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Defines		
STM32746G_DISCOVERY_LCD Exported Constants			
STM32746G_DISCOVERY_LCD			

Defines

```
#define MAX_LAYER_NUMBER ((uint32_t)2)
#define LCD_LayerCfgTypeDef LTDC_LayerCfgTypeDef
#define LTDC_ACTIVE_LAYER ((uint32_t)1) /* Layer 1 */
#define LCD_OK ((uint8_t)0x00)
LCD status structure definition.

#define LCD_ERROR ((uint8_t)0x01)
#define LCD_TIMEOUT ((uint8_t)0x02)
#define LCD_FB_START_ADDRESS ((uint32_t)0xC0000000)
LCD FB_StartAddress.

#define LCD_COLOR_BLUE ((uint32_t)0xFF0000FF)
LCD color.

#define LCD_COLOR_GREEN ((uint32_t)0xFF00FF00)
#define LCD_COLOR_RED ((uint32_t)0xFFFF0000)
#define LCD_COLOR_CYAN ((uint32_t)0xFF00FFFF)
#define LCD_COLOR_MAGENTA ((uint32_t)0xFFFF00FF)
#define LCD_COLOR_YELLOW ((uint32_t)0xFFFFFFFF00)
#define LCD_COLOR_LIGHTBLUE ((uint32_t)0xFF8080FF)
#define LCD_COLOR_LIGHTGREEN ((uint32_t)0xFF80FF80)
#define LCD_COLOR_LIGHTRED ((uint32_t)0xFFFF8080)
#define LCD_COLOR_LIGHTCYAN ((uint32_t)0xFF80FFFF)
#define LCD_COLOR_LIGHTMAGENTA ((uint32_t)0xFFFF80FF)
#define LCD_COLOR_LIGHTYELLOW ((uint32_t)0xFFFFFFFF80)
#define LCD_COLOR_DARKBLUE ((uint32_t)0xFF000080)
#define LCD_COLOR_DARKGREEN ((uint32_t)0xFF008000)
#define LCD_COLOR_DARKRED ((uint32_t)0xFF800000)
#define LCD_COLOR_DARKCYAN ((uint32_t)0xFF008080)
#define LCD_COLOR_DARKMAGENTA ((uint32_t)0xFF800080)
#define LCD_COLOR_DARKYELLOW ((uint32_t)0xFF808000)
#define LCD_COLOR_WHITE ((uint32_t)0xFFFFFFFFFF)
#define LCD_COLOR_LIGHTGRAY ((uint32_t)0xFFD3D3D3)
#define LCD_COLOR_GRAY ((uint32_t)0xFF808080)
```

```
#define LCD_COLOR_DARKGRAY ((uint32_t)0xFF404040)
#define LCD_COLOR_BLACK ((uint32_t)0xFF000000)
#define LCD_COLOR_BROWN ((uint32_t)0xFFA52A2A)
#define LCD_COLOR_ORANGE ((uint32_t)0xFFFFA500)
#define LCD_COLOR_TRANSPARENT ((uint32_t)0xFF000000)
#define LCD_DEFAULT_FONT Font24
LCD default font.

#define LCD_RELOAD_IMMEDIATE ((uint32_t)LTDC_SRCR_IMR)
LCD Reload Types.

#define LCD_RELOAD_VERTICAL_BLANKING ((uint32_t)LTDC_SF
#define LCD_DISP_PIN GPIO_PIN_12
LCD special pins.

#define LCD_DISP_GPIO_PORT GPIOI
#define LCD_DISP_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_ENABLE()
#define LCD_DISP_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DISABLE()
#define LCD_BL_CTRL_PIN GPIO_PIN_3
#define LCD_BL_CTRL_GPIO_PORT GPIOK
#define LCD_BL_CTRL_GPIO_CLK_ENABLE() __HAL_RCC_GPIOK_CLK_ENABLE()
#define LCD_BL_CTRL_GPIO_CLK_DISABLE() __HAL_RCC_GPIOK_CLK_DISABLE()
```

Define Documentation

```
#define LCD_BL_CTRL_GPIO_CLK_DISABLE() __HAL_RCC_GPI
```

Definition at line 175 of file `stm32746g_discovery_lcd.h`.

```
#define LCD_BL_CTRL_GPIO_CLK_ENABLE() __HAL_RCC_GPI
```

Definition at line 174 of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_MspInit()`.

```
#define LCD_BL_CTRL_GPIO_PORT GPIOK
```

Definition at line 173 of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_DisplayOff()`, `BSP_LCD_DisplayOn()`,
`BSP_LCD_Init()`, and `BSP_LCD_MspInit()`.

```
#define LCD_BL_CTRL_PIN GPIO_PIN_3
```

Definition at line 172 of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_DisplayOff()`, `BSP_LCD_DisplayOn()`,
`BSP_LCD_Init()`, and `BSP_LCD_MspInit()`.

```
#define LCD_COLOR_BLACK ((uint32_t)0xFF000000)
```

Definition at line 145 of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_LayerDefaultInit()`, and
`BSP_LCD_LayerRgb565Init()`.

```
#define LCD_COLOR_BLUE ((uint32_t)0xFF0000FF)
```

LCD color.

Definition at line [123](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_BROWN ((uint32_t)0xFFA52A2A)
```

Definition at line [146](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_CYAN ((uint32_t)0xFF00FFFF)
```

Definition at line [126](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_DARKBLUE ((uint32_t)0xFF000080)
```

Definition at line [135](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_DARKCYAN ((uint32_t)0xFF008080)
```

Definition at line [138](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_DARKGRAY ((uint32_t)0xFF404040)
```

Definition at line [144](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_DARKGREEN ((uint32_t)0xFF008000)
```

Definition at line [136](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_DARKMAGENTA ((uint32_t)0xFF800080)
```

Definition at line [139](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_DARKRED ((uint32_t)0xFF800000)
```

Definition at line [137](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_DARKYELLOW ((uint32_t)0xFF808000)
```

Definition at line [140](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_GRAY ((uint32_t)0xFF808080)
```

Definition at line [143](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_GREEN ((uint32_t)0xFF00FF00)
```

Definition at line [124](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_LIGHTBLUE ((uint32_t)0xFF8080FF)
```

Definition at line [129](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_LIGHTCYAN ((uint32_t)0xFF80FFFF)
```

Definition at line [132](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_LIGHTGRAY ((uint32_t)0xFFD3D3D3)
```

Definition at line [142](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_LIGHTGREEN ((uint32_t)0xFF80FF80)
```

Definition at line [130](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_LIGHTMAGENTA ((uint32_t)0xFFFF80FF)
```

Definition at line [133](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_LIGHTRED ((uint32_t)0xFFFF8080)
```

Definition at line [131](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_LIGHTYELLOW ((uint32_t)0xFFFFFFF80)
```

Definition at line [134](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_MAGENTA ((uint32_t)0xFFFF00FF)
```

Definition at line [127](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_ORANGE ((uint32_t)0xFFFFA500)
```

Definition at line [147](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_RED ((uint32_t)0xFFFF0000)
```

Definition at line [125](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_TRANSPARENT ((uint32_t)0xFF000000)
```

Definition at line [148](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_COLOR_WHITE ((uint32_t)0xFFFFFFFF)
```

Definition at line [141](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_LayerDefaultInit()`, and
`BSP_LCD_LayerRgb565Init()`.

```
#define LCD_COLOR_YELLOW ((uint32_t)0xFFFFF00)
```

Definition at line [128](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_DEFAULT_FONT Font24
```

LCD default font.

Definition at line [153](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_Init()`.

```
#define LCD_DISP_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_C
```

Definition at line [169](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_DISP_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_C
```

Definition at line [168](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_MspInit()`.

```
#define LCD_DISP_GPIO_PORT GPIOI
```

Definition at line [167](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_DisplayOff()`, `BSP_LCD_DisplayOn()`, `BSP_LCD_Init()`, and `BSP_LCD_MspInit()`.

```
#define LCD_DISP_PIN GPIO_PIN_12
```

LCD special pins.

Definition at line [166](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_DisplayOff()`, `BSP_LCD_DisplayOn()`, `BSP_LCD_Init()`, and `BSP_LCD_MspInit()`.

```
#define LCD_ERROR ((uint8_t)0x01)
```

Definition at line [112](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_FB_START_ADDRESS ((uint32_t)0xC0000000)
```

LCD FB_StartAddress.

Definition at line [118](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_LayerCfgTypeDef LTDC_LayerCfgTypeDef
```

Definition at line [105](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_LayerDefaultInit()`, and
`BSP_LCD_LayerRgb565Init()`.

```
#define LCD_OK ((uint8_t)0x00)
```

LCD status structure definition.

Definition at line [111](#) of file `stm32746g_discovery_lcd.h`.

Referenced by `BSP_LCD_DeInit()`, and `BSP_LCD_Init()`.

```
#define LCD_RELOAD_IMMEDIATE ((uint32_t)LTDC_SRCR_IMR)
```

LCD Reload Types.

Definition at line [158](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_RELOAD_VERTICAL_BLANKING ((uint32_t)LTDC_SI
```

Definition at line [159](#) of file `stm32746g_discovery_lcd.h`.

```
#define LCD_TIMEOUT ((uint8_t)0x02)
```

Definition at line [113](#) of file `stm32746g_discovery_lcd.h`.

```
#define LTDC_ACTIVE_LAYER ((uint32_t)1) /* Layer 1 */
```

Definition at line [107](#) of file `stm32746g_discovery_lcd.h`.

```
#define MAX_LAYER_NUMBER ((uint32_t)2)
```

Definition at line **103** of file **stm32746g_discovery_lcd.h**.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Defines

STM32746G_DISCOVERY_LOW_LEVEL_LED

[STM32746G_DISCOVERY_LOW_LEVEL Exported Constants](#)

Define for STM32746G_DISCOVERY board. [More...](#)

Defines

```
#define LEDn ((uint8_t)1)
#define LED1_GPIO_PORT GPIOI
#define LED1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_EN
#define LED1_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_DI
#define LED1_PIN GPIO_PIN_1
```

Detailed Description

Define for STM32746G_DISCOVERY board.

Define Documentation

```
#define LED1_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_
```

Definition at line [112](#) of file `stm32746g_discovery.h`.

```
#define LED1_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_I
```

Definition at line [111](#) of file `stm32746g_discovery.h`.

Referenced by `BSP_LED_Init()`.

```
#define LED1_GPIO_PORT GPIOI
```

Definition at line [110](#) of file `stm32746g_discovery.h`.

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

```
#define LED1_PIN GPIO_PIN_1
```

Definition at line [113](#) of file `stm32746g_discovery.h`.

```
#define LEDn ((uint8_t)1)
```

Definition at line [108](#) of file `stm32746g_discovery.h`.

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Defines		
STM32746G_DISCOVERY_LCD Private Defines			<u>STM32746G_DISCOVERY_LCD</u>

Defines

```
#define POLY_X(Z) ((int32_t)((Points + Z)->X))  
#define POLY_Y(Z) ((int32_t)((Points + Z)->Y))
```

Define Documentation

```
#define POLY_X( Z ) ((int32_t)((Points + Z)->X))
```

Definition at line [102](#) of file [stm32746g_discovery_lcd.c](#).

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

```
#define POLY_Y( Z ) ((int32_t)((Points + Z)->Y))
```

Definition at line [103](#) of file [stm32746g_discovery_lcd.c](#).

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

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Variables
STM32746G_DISCOVERY_QSPI_Private Variables				
STM32746G-Discovery QSPI				

Variables

QSPI_HandleTypeDef **QSPIHandle**

Variable Documentation

QSPI_HandleTypeDef QSPIHandle

Definition at line **88** of file [stm32746g_discovery_qspi.c](#).

Referenced by [BSP_QSPI_DeInit\(\)](#),
[BSP_QSPI_EnableMemoryMappedMode\(\)](#),
[BSP_QSPI_Erase_Block\(\)](#), [BSP_QSPI_Erase_Chip\(\)](#),
[BSP_QSPI_GetStatus\(\)](#), [BSP_QSPI_Init\(\)](#), [BSP_QSPI_Read\(\)](#), and
[BSP_QSPI_Write\(\)](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories	Defines		
STM32746G_DISCOVERY_LOW_LEVEL_SIGNAL			
STM32746G_DISCOVERY_LOW_LEVEL Exported Constants			

Defines

```
#define SIGNALn ((uint8_t)1)
#define SD_DETECT_PIN GPIO_PIN_13
SD-detect signal.

#define SD_DETECT_GPIO_PORT GPIOC
#define SD_DETECT_GPIO_CLK_ENABLE() __HAL_RCC_GPIOC_
#define SD_DETECT_GPIO_CLK_DISABLE() __HAL_RCC_GPIOC_
#define SD_DETECT_EXTI_IRQn EXTI15_10_IRQn
#define TS_INT_PIN GPIO_PIN_13
Touch screen interrupt signal.

#define TS_INT_GPIO_PORT GPIOI
#define TS_INT_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK_E_
#define TS_INT_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK_
#define TS_INT_EXTI_IRQn EXTI15_10_IRQn
```

Define Documentation

```
#define SD_DETECT EXTI_IRQn  EXTI15_10_IRQn
```

Definition at line [174](#) of file `stm32746g_discovery.h`.

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

```
#define SD_DETECT_GPIO_CLK_DISABLE() __HAL_RCC_GPIO
```

Definition at line [173](#) of file `stm32746g_discovery.h`.

```
#define SD_DETECT_GPIO_CLK_ENABLE() __HAL_RCC_GPIO
```

Definition at line [172](#) of file `stm32746g_discovery.h`.

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

```
#define SD_DETECT_GPIO_PORT GPIOC
```

Definition at line [171](#) of file `stm32746g_discovery.h`.

Referenced by [BSP_SD_Detect_MspInit\(\)](#), [BSP_SD_IsDetected\(\)](#), and [BSP_SD_ITConfig\(\)](#).

```
#define SD_DETECT_PIN GPIO_PIN_13
```

SD-detect signal.

Definition at line [170](#) of file `stm32746g_discovery.h`.

Referenced by [BSP_SD_Detect_MspInit\(\)](#), [BSP_SD_IsDetected\(\)](#),

and [BSP_SD_ITConfig\(\)](#).

```
#define SIGNALn ((uint8_t)1)
```

Definition at line [165](#) of file [stm32746g_discovery.h](#).

```
#define TS_INT EXTI IRQn EXTI15_10 IRQn
```

Definition at line [183](#) of file [stm32746g_discovery.h](#).

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

```
#define TS_INT_GPIO_CLK_DISABLE() __HAL_RCC_GPIOI_CLK
```

Definition at line [182](#) of file [stm32746g_discovery.h](#).

```
#define TS_INT_GPIO_CLK_ENABLE() __HAL_RCC_GPIOI_CLK
```

Definition at line [181](#) of file [stm32746g_discovery.h](#).

```
#define TS_INT_GPIO_PORT GPIOI
```

Definition at line [180](#) of file [stm32746g_discovery.h](#).

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

```
#define TS_INT_PIN GPIO_PIN_13
```

Touch screen interrupt signal.

Definition at line [179](#) of file [stm32746g_discovery.h](#).

Referenced by **BSP_TS_ITConfig()**.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Modules Defines

STM32746G_DISCOVERY_SDRAM Exported Types

[STM32746G_DISCOVERY_SDRAM](#)

Modules

[**STM32746G_DISCOVERY_SDRAM Exported Constants**](#)

Defines

```
#define SDRAM_OK ((uint8_t)0x00)
SDRAM status structure definition.

#define SDRAM_ERROR ((uint8_t)0x01)

#define SDRAM_MODEREG_BURST_LENGTH_1 ((uint16_t)0x0000
FMC SDRAM Mode definition register defines.

#define SDRAM_MODEREG_BURST_LENGTH_2 ((uint16_t)0x0001
#define SDRAM_MODEREG_BURST_LENGTH_4 ((uint16_t)0x0002
#define SDRAM_MODEREG_BURST_LENGTH_8 ((uint16_t)0x0004
#define SDRAM_MODEREG_BURST_TYPE_SEQUENTIAL ((uint16
#define SDRAM_MODEREG_BURST_TYPE_INTERLEAVED ((uint1
#define SDRAM_MODEREG_CAS_LATENCY_2 ((uint16_t)0x0020)
#define SDRAM_MODEREG_CAS_LATENCY_3 ((uint16_t)0x0030)
#define SDRAM_MODEREG_OPERATING_MODE_STANDARD ((ui
#define SDRAM_MODEREG_WRITEBURST_MODE_PROGRAMMABLE
#define SDRAM_MODEREG_WRITEBURST_MODE_SINGLE ((uint:
```

Define Documentation

```
#define SDRAM_ERROR ((uint8_t)0x01)
```

Definition at line [70](#) of file [stm32746g_discovery_sdram.h](#).

Referenced by [BSP_SDRAM_DelInit\(\)](#), [BSP_SDRAM_Init\(\)](#),
[BSP_SDRAM_ReadData\(\)](#), [BSP_SDRAM_ReadData_DMA\(\)](#),
[BSP_SDRAM_Sendcmd\(\)](#), [BSP_SDRAM_WriteData\(\)](#), and
[BSP_SDRAM_WriteData_DMA\(\)](#).

```
#define SDRAM_MODEREG_BURST_LENGTH_1 ((uint16_t)0x0000)
```

FMC SDRAM Mode definition register defines.

Definition at line [102](#) of file [stm32746g_discovery_sdram.h](#).

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

```
#define SDRAM_MODEREG_BURST_LENGTH_2 ((uint16_t)0x0001)
```

Definition at line [103](#) of file [stm32746g_discovery_sdram.h](#).

```
#define SDRAM_MODEREG_BURST_LENGTH_4 ((uint16_t)0x0002)
```

Definition at line [104](#) of file [stm32746g_discovery_sdram.h](#).

```
#define SDRAM_MODEREG_BURST_LENGTH_8 ((uint16_t)0x0004)
```

Definition at line [105](#) of file [stm32746g_discovery_sdram.h](#).

```
#define SDRAM_MODEREG_BURST_TYPE_INTERLEAVED ((uint1
```

Definition at line [107](#) of file `stm32746g_discovery_sdram.h`.

```
#define SDRAM_MODEREG_BURST_TYPE_SEQUENTIAL ((uint16
```

Definition at line [106](#) of file `stm32746g_discovery_sdram.h`.

Referenced by `BSP_SDRAM_Initialization_sequence()`.

```
#define SDRAM_MODEREG_CAS_LATENCY_2 ((uint16_t)0x0020)
```

Definition at line [108](#) of file `stm32746g_discovery_sdram.h`.

Referenced by `BSP_SDRAM_Initialization_sequence()`.

```
#define SDRAM_MODEREG_CAS_LATENCY_3 ((uint16_t)0x0030)
```

Definition at line [109](#) of file `stm32746g_discovery_sdram.h`.

```
#define SDRAM_MODEREG_OPERATING_MODE_STANDARD ((ui
```

Definition at line [110](#) of file `stm32746g_discovery_sdram.h`.

Referenced by `BSP_SDRAM_Initialization_sequence()`.

```
#define SDRAM_MODEREG_WRITEBURST_MODE_PROGRAMMED
```

Definition at line [111](#) of file `stm32746g_discovery_sdram.h`.

```
#define SDRAM_MODEREG_WRITEBURST_MODE_SINGLE ((uint
```

Definition at line [112](#) of file `stm32746g_discovery_sdram.h`.

Referenced by `BSP_SDRAM_Initialization_sequence()`.

```
#define SDRAM_OK ((uint8_t)0x00)
```

SDRAM status structure definition.

Definition at line [69](#) of file `stm32746g_discovery_sdram.h`.

Referenced by `BSP_SDRAM_DelInit()`, `BSP_SDRAM_Init()`,
`BSP_SDRAM_ReadData()`, `BSP_SDRAM_ReadData_DMA()`,
`BSP_SDRAM_Sendcmd()`, `BSP_SDRAM_WriteData()`, and
`BSP_SDRAM_WriteData_DMA()`.

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				Variables
STM32746G_DISCOVERY_TS Imported Variables				STM32746G_DISCOVERY_TS

Variables

char * **ts_event_string_tab [TOUCH_EVENT_NB_MAX]**

Table for touchscreen event information display on LCD : table indexed on enum **TS_TouchEventTypeDef** information.

char * **ts_gesture_id_string_tab [GEST_ID_NB_MAX]**

Table for touchscreen gesture Id information display on LCD : table indexed on enum **TS_GestureIdTypeDef** information.

Variable Documentation

`char* ts_event_string_tab[TOUCH_EVENT_NB_MAX]`

Table for touchscreen event information display on LCD : table indexed on enum `TS_TouchEventTypeDef` information.

`char* ts_gesture_id_string_tab[GEST_ID_NB_MAX]`

Table for touchscreen gesture Id information display on LCD : table indexed on enum `TS_GestureIdTypeDef` information.

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
				Variables
STM32746G_DISCOVERY_SD Private Variables				
<u>STM32746G_DISCOVERY_SD</u>				

Variables

SD_HandleTypeDef **uSdHandle**

Variable Documentation

SD_HandleTypeDef uSdHandle

Definition at line [118](#) of file [stm32746g_discovery_sd.c](#).

Referenced by [BSP_SD_DelInit\(\)](#), [BSP_SD_Erase\(\)](#),
[BSP_SD_GetCardInfo\(\)](#), [BSP_SD_GetCardState\(\)](#), [BSP_SD_Init\(\)](#),
[BSP_SD_ReadBlocks\(\)](#), [BSP_SD_ReadBlocks_DMA\(\)](#),
[BSP_SD_WriteBlocks\(\)](#), and [BSP_SD_WriteBlocks_DMA\(\)](#).

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Drivers			

Drivers Directory Reference

Directories

directory **BSP**

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Drivers > BSP >			
BSP Directory Reference			

Directories

directory **STM32746G-Discovery**

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
Drivers	BSP	STM32746G-Discovery	

STM32746G-Discovery Directory Reference

Files

file **stm32746g_discovery.c** [code]

This file provides a set of firmware functions to manage LEDs, push-buttons and COM ports available on STM32746G-Discovery board(MB1191) from STMicroelectronics.

file **stm32746g_discovery.h** [code]

This file contains definitions for STM32746G_DISCOVERY's LEDs, push-buttons and COM ports hardware resources.

file **stm32746g_discovery_audio.c** [code]

This file provides the Audio driver for the STM32746G-Discovery board.

file **stm32746g_discovery_audio.h** [code]

This file contains the common defines and functions prototypes for the **stm32746g_discovery_audio.c** driver.

file **stm32746g_discovery_camera.c** [code]

This file includes the driver for Camera modules mounted on STM32746G-Discovery board.

file **stm32746g_discovery_camera.h** [code]

This file contains the common defines and functions prototypes for the **stm32746g_discovery_camera.c** driver.

file **stm32746g_discovery_eeprom.c** [code]

This file provides a set of functions needed to manage an I2C M24LR64 EEPROM memory.

file **stm32746g_discovery_eeprom.h** [code]

This file contains all the functions prototypes for the **stm32746g_discovery_eeprom.c** firmware driver.

file **stm32746g_discovery_lcd.c** [code]

This file includes the driver for Liquid Crystal Display (LCD) module mounted on STM32746G-Discovery board.

file **stm32746g_discovery_lcd.h** [code]

This file contains the common defines and functions prototypes for the **stm32746g_discovery_lcd.c** driver.

file **stm32746g_discovery_qspi.c** [code]

This file includes a standard driver for the N25Q128A QSPI memory mounted on STM32746G-Discovery board.

file **stm32746g_discovery_qspi.h** [code]

This file contains the common defines and functions prototypes for the **stm32746g_discovery_qspi.c** driver.

file **stm32746g_discovery_sd.c** [code]

This file includes the uSD card driver mounted on STM32746G-Discovery board.

file **stm32746g_discovery_sd.h** [code]

This file contains the common defines and functions prototypes for the **stm32746g_discovery_sd.c** driver.

file **stm32746g_discovery_sdram.c** [code]

This file includes the SDRAM driver for the MT48LC4M32B2B5-7 memory device mounted on STM32746G-Discovery board.

file **stm32746g_discovery_sdram.h** [code]

This file contains the common defines and functions prototypes for the **stm32746g_discovery_sdram.c** driver.

file **stm32746g_discovery_ts.c** [code]

This file provides a set of functions needed to manage the Touch Screen on STM32746G-Discovery board.

file **stm32746g_discovery_ts.h** [code]

This file contains the common defines and functions prototypes for the **stm32746g_discovery_ts.c** driver.

BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains definitions
for STM32746G_DISCOVERY's LEDs,
00008 *           push-buttons and COM ports hard
ware resources.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and bin
ary forms, with or without modification,
00015 * are permitted provided that the followin
g conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038      */
00039 /* Define to prevent recursive inclusion ---
00040 -----
00041 #ifndef __STM32746G_DISCOVERY_H
00042 #define __STM32746G_DISCOVERY_H
00043
00044 #ifdef __cplusplus
00045   extern "C" {
00046 #endif
00047 /* Includes -----
00048 -----
00049
00050 /** @addtogroup BSP
00051   * @{
00052   */
00053
00054 /** @addtogroup STM32746G_DISCOVERY
00055   * @{
00056   */
00057
00058 /** @addtogroup STM32746G_DISCOVERY_LOW_LEVEL
00059   * @{
00060   */
00061
00062 /** @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
00063 Exported_Types STM32746G_DISCOVERY_LOW_LEVEL_ Exported_Types
00064   * @{
00065   */
00066 typedef enum
```

```
00066 {
00067 LED1 = 0,
00068 LED_GREEN = LED1,
00069 }Led_TypeDef;
00070
00071 typedef enum
00072 {
00073     BUTTON_WAKEUP = 0,
00074     BUTTON_TAMPER = 1,
00075     BUTTON_KEY = 2
00076 }Button_TypeDef;
00077
00078 typedef enum
00079 {
00080     BUTTON_MODE_GPIO = 0,
00081     BUTTON_MODE_EXTI = 1
00082 }ButtonMode_TypeDef;
00083
00084 typedef enum
00085 {
00086     COM1 = 0,
00087     COM2 = 1
00088 }COM_TypeDef;
00089 /**
00090 * @}
00091 */
00092
00093 /** @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
Exported_Constants STM32746G_DISCOVERY_LOW_LEVEL E
xported Constants
00094 * @{
00095 */
00096
00097 /**
00098 * @brief Define for STM32746G_DISCOVERY b
oard
00099 */
```

```
00100 #if !defined (USE_STM32746G_DISCO)
00101 #define USE_STM32746G_DISCO
00102 #endif
00103
00104 /** @addtogroup STM32746G_DISCOVERY_LOW_LEVEL_LED
00105 * @{
00106 */
00107
00108 #define LEDn ((uint8_t)1)
00109
00110 #define LED1_GPIO_PORT GPIO_OI
00111 #define LED1_GPIO_CLK_ENABLE() AL_RCC_GPIOI_CLK_ENABLE()
00112 #define LED1_GPIO_CLK_DISABLE() AL_RCC_GPIOI_CLK_DISABLE()
00113 #define LED1_PIN GPIO_PIN_1
00114
00115 /**
00116 * @{
00117 */
00118
00119 /** @addtogroup STM32746G_DISCOVERY_LOW_LEVEL_BUTTON
00120 * @{
00121 */
00122 #define BUTTONn ((uint8_t)3)
00123
00124 /**
00125 * @brief Wakeup push-button
00126 */
00127 #define WAKEUP_BUTTON_PIN GPIO_PIN_11
```

```
00128 #define WAKEUP_BUTTON_GPIO_PORT
GPIOI
00129 #define WAKEUP_BUTTON_GPIO_CLK_ENABLE()
__HAL_RCC_GPIOI_CLK_ENABLE()
00130 #define WAKEUP_BUTTON_GPIO_CLK_DISABLE()
__HAL_RCC_GPIOI_CLK_DISABLE()
00131 #define WAKEUP_BUTTON_EXTI_IRQn
EXTI15_10_IRQn
00132
00133 /**
00134  * @brief Tamper push-button
00135 */
00136 #define TAMPER_BUTTON_PIN
GPIO_PIN_11
00137 #define TAMPER_BUTTON_GPIO_PORT
GPIOI
00138 #define TAMPER_BUTTON_GPIO_CLK_ENABLE()
__HAL_RCC_GPIOI_CLK_ENABLE()
00139 #define TAMPER_BUTTON_GPIO_CLK_DISABLE()
__HAL_RCC_GPIOI_CLK_DISABLE()
00140 #define TAMPER_BUTTON_EXTI_IRQn
EXTI15_10_IRQn
00141
00142 /**
00143  * @brief Key push-button
00144 */
00145 #define KEY_BUTTON_PIN
GPIO_PIN_11
00146 #define KEY_BUTTON_GPIO_PORT
GPIOI
00147 #define KEY_BUTTON_GPIO_CLK_ENABLE()
__HAL_RCC_GPIOI_CLK_ENABLE()
00148 #define KEY_BUTTON_GPIO_CLK_DISABLE()
__HAL_RCC_GPIOI_CLK_DISABLE()
00149 #define KEY_BUTTON_EXTI_IRQn
EXTI15_10_IRQn
00150
```

```
00151 #define BUTTONx_GPIO_CLK_ENABLE(__INDEX__)
        do { if((__INDEX__) == 0) WAKEUP_BUTTON_GPIO_CLK
        _ENABLE(); else\
00152            if((__INDEX__) == 1) TAMPER_BUTTON_GPIO_CLK
        _ENABLE(); else\
00153
                                         KEY_BUTTON_GPIO_CLK_ENAB
LE(); } while(0)
00154
00155 #define BUTTONx_GPIO_CLK_DISABLE(__INDEX__)
        (((__INDEX__) == 0) ? WAKEUP_BUTTON_GPIO_CLK_DI
SABLE() :\
00156        ((__INDEX__) == 1) ? TAMPER_BUTTON_GPIO_CLK_DI
SABLE() : KEY_BUTTON_GPIO_CLK_DISABLE())
00157
00158 /**
00159 * @}
00160 */
00161
00162 /** @addtogroup STM32746G_DISCOVERY_LOW_LEVEL_SIGNAL
L_SIGNAL
00163 * @{
00164 */
00165 #define SIGNALn
((uint8_t)1)
00166
00167 /**
00168 * @brief SD-detect signal
00169 */
00170 #define SD_DETECT_PIN
        GPIO_PIN_13
00171 #define SD_DETECT_GPIO_PORT
```

```
GPIOC
00172 #define SD_DETECT_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOC_CLK_ENABLE()
00173 #define SD_DETECT_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOC_CLK_DISABLE()
00174 #define SD_DETECT EXTI_IRQn
        EXTI15_10_IRQn
00175
00176 /**
00177     * @brief Touch screen interrupt signal
00178 */
00179 #define TS_INT_PIN
        GPIO_PIN_13
00180 #define TS_INT_GPIO_PORT
        GPIOI
00181 #define TS_INT_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOI_CLK_ENABLE()
00182 #define TS_INT_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOI_CLK_DISABLE()
00183 #define TS_INT_EXTI_IRQn
        EXTI15_10_IRQn
00184
00185 /**
00186     * @}
00187 */
00188
00189 /** @addtogroup STM32746G_DISCOVERY_LOW_LEVEL_COM
00190     * @{
00191 */
00192 #define COMn                                ((u
int8_t)1)
00193
00194 /**
00195     * @brief Definition for COM port1, connected to USART1
00196 */

```

```
00197 #define DISCOVERY_COM1
        USART1
00198 #define DISCOVERY_COM1_CLK_ENABLE()
        __HAL_RCC_USART1_CLK_ENABLE()
00199 #define DISCOVERY_COM1_CLK_DISABLE()
        __HAL_RCC_USART1_CLK_DISABLE()
00200
00201 #define DISCOVERY_COM1_TX_PIN
        GPIO_PIN_9
00202 #define DISCOVERY_COM1_TX_GPIO_PORT
        GPIOA
00203 #define DISCOVERY_COM1_TX_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOA_CLK_ENABLE()
00204 #define DISCOVERY_COM1_TX_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOA_CLK_DISABLE()
00205 #define DISCOVERY_COM1_TX_AF
        GPIO_AF7_USART1
00206
00207 #define DISCOVERY_COM1_RX_PIN
        GPIO_PIN_7
00208 #define DISCOVERY_COM1_RX_GPIO_PORT
        GPIOB
00209 #define DISCOVERY_COM1_RX_GPIO_CLK_ENABLE()
        __HAL_RCC_GPIOB_CLK_ENABLE()
00210 #define DISCOVERY_COM1_RX_GPIO_CLK_DISABLE()
        __HAL_RCC_GPIOB_CLK_DISABLE()
00211 #define DISCOVERY_COM1_RX_AF
        GPIO_AF7_USART1
00212
00213 #define DISCOVERY_COM1 IRQn
        USART1_IRQn
00214
00215 #define DISCOVERY_COMx_CLK_ENABLE(__INDEX__)
            do { if((__INDEX__) == COM1) DISCOVERY
        _COM1_CLK_ENABLE(); } while(0)
00216 #define DISCOVERY_COMx_CLK_DISABLE(__INDEX__)
            (((__INDEX__) == 0) ? DISCOVERY_COM1_C
```

```

LK_DISABLE() : 0)
00217
00218 #define DISCOVERY_COMx_TX_GPIO_CLK_ENABLE(__
INDEX__) do { if((__INDEX__) == COM1) DISCOVERY
_COM1_TX_GPIO_CLK_ENABLE(); } while(0)
00219 #define DISCOVERY_COMx_TX_GPIO_CLK_DISABLE(_
INDEX_) (((__INDEX__) == 0) ? DISCOVERY_COM1_T
X_GPIO_CLK_DISABLE() : 0)
00220
00221 #define DISCOVERY_COMx_RX_GPIO_CLK_ENABLE(_
INDEX__) do { if((__INDEX__) == COM1) DISCOVERY
_COM1_RX_GPIO_CLK_ENABLE(); } while(0)
00222 #define DISCOVERY_COMx_RX_GPIO_CLK_DISABLE(_
INDEX_) (((__INDEX__) == 0) ? DISCOVERY_COM1_R
X_GPIO_CLK_DISABLE() : 0)
00223
00224 /* Exported constant IO -----
-----*/
00225
00226 #define LCD_I2C_ADDRESS ((u
int16_t)0x70)
00227 #define CAMERA_I2C_ADDRESS ((u
int16_t)0x60)
00228 #define AUDIO_I2C_ADDRESS ((u
int16_t)0x34)
00229 #define EEPROM_I2C_ADDRESS_A01 ((u
int16_t)0xA0)
00230 #define EEPROM_I2C_ADDRESS_A02 ((u
int16_t)0xA6)
00231 #define TS_I2C_ADDRESS ((u
int16_t)0x70)
00232
00233 /* I2C clock speed configuration (in Hz)
00234     WARNING:
00235     Make sure that this define is not already
declared in other files (ie.
00236     stm32746g_discovery.h file). It can be us

```

```
ed in parallel by other modules. */
00237 #ifndef I2C_SPEED
00238 #define I2C_SPEED ((u
int32_t)100000)
00239 #endif /* I2C_SPEED */
00240
00241 /* User can use this section to tailor I2Cx/
I2Cx instance used and associated
00242 resources */
00243 /* Definition for AUDIO and LCD I2Cx resourc
es */
00244 #define DISCOVERY_AUDIO_I2Cx
    I2C3
00245 #define DISCOVERY_AUDIO_I2Cx_CLK_ENABLE()
    __HAL_RCC_I2C3_CLK_ENABLE()
00246 #define DISCOVERY_AUDIO_DMAX_CLK_ENABLE()
    __HAL_RCC_DMA1_CLK_ENABLE()
00247 #define DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_CL
K_ENABLE() __HAL_RCC_GPIOH_CLK_ENABLE()
00248
00249 #define DISCOVERY_AUDIO_I2Cx_FORCE_RESET()
    __HAL_RCC_I2C3_FORCE_RESET()
00250 #define DISCOVERY_AUDIO_I2Cx_RELEASE_RESET()
    __HAL_RCC_I2C3_RELEASE_RESET()
00251
00252 /* Definition for I2Cx Pins */
00253 #define DISCOVERY_AUDIO_I2Cx_SCL_PIN
    GPIO_PIN_7
00254 #define DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PO
RT
    GPIOH
00255 #define DISCOVERY_AUDIO_I2Cx_SCL_SDA_AF
    GPIO_AF4_I2C3
00256 #define DISCOVERY_AUDIO_I2Cx_SDA_PIN
    GPIO_PIN_8
00257
00258 /* I2C interrupt requests */
00259 #define DISCOVERY_AUDIO_I2Cx_EV IRQn
```

```
          I2C3_EV IRQn
00260 #define DISCOVERY_AUDIO_I2Cx_ER_IRQn
          I2C3_ER IRQn
00261
00262 /* Definition for external, camera and Arduino connector I2Cx resources */
00263 #define DISCOVERY_EXT_I2Cx
          I2C1
00264 #define DISCOVERY_EXT_I2Cx_CLK_ENABLE()
          __HAL_RCC_I2C1_CLK_ENABLE()
00265 #define DISCOVERY_EXT_DMAx_CLK_ENABLE()
          __HAL_RCC_DMA1_CLK_ENABLE()
00266 #define DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_CLK_
ENABLE()      __HAL_RCC_GPIOB_CLK_ENABLE()
00267
00268 #define DISCOVERY_EXT_I2Cx_FORCE_RESET()
          __HAL_RCC_I2C1_FORCE_RESET()
00269 #define DISCOVERY_EXT_I2Cx_RELEASE_RESET()
          __HAL_RCC_I2C1_RELEASE_RESET()
00270
00271 /* Definition for I2Cx Pins */
00272 #define DISCOVERY_EXT_I2Cx_SCL_PIN
          GPIO_PIN_8
00273 #define DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_PORT
          GPIOB
00274 #define DISCOVERY_EXT_I2Cx_SCL_SDA_AF
          GPIO_AF4_I2C1
00275 #define DISCOVERY_EXT_I2Cx_SDA_PIN
          GPIO_PIN_9
00276
00277 /* I2C interrupt requests */
00278 #define DISCOVERY_EXT_I2Cx_EV_IRQn
          I2C1_EV IRQn
00279 #define DISCOVERY_EXT_I2Cx_ER_IRQn
          I2C1_ER IRQn
00280
00281 /* I2C TIMING Register define when I2C clock
```

```
    source is SYSCLK */
00282 /* I2C TIMING is calculated from APB1 source
   clock = 50 MHz */
00283 /* Due to the big MOFSET capacity for adapting
   the camera level the rising time is very large
   (>1us) */
00284 /* 0x40912732 takes in account the big rising
   and aims a clock of 100khz */
00285 #ifndef DISCOVERY_I2Cx_TIMING
00286 #define DISCOVERY_I2Cx_TIMING
00287     ((uint32_t)0x40912732)
00288#endif /* DISCOVERY_I2Cx_TIMING */
00289 /**
00290 * @}
00291 */
00292 /**
00293 * @}
00294 * @}
00295 */
00296 /**
00297 ** @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
   Exported_Macros STM32746G_DISCOVERY_LOW_LEVEL_Exported_Macros
00298 * @{
00299 */
00300 /**
00301 * @}
00302 */
00303 /**
00304 ** @addtogroup STM32746G_DISCOVERY_LOW_LEVEL_
   Exported_Functions
00305 * @{
00306 */
00307 uint32_t    BSP_GetVersion(void);
00308 void        BSP_LED_Init(Led_TypeDef Led);
00309 void        BSP_LED_DeInit(Led_TypeDef Led);
```

```
00310 void          BSP_LED_On(Led_TypeDef Led);
00311 void          BSP_LED_Off(Led_TypeDef Led);
00312 void          BSP_LED_Toggle(Led_TypeDef Led);
00313 void          BSP_PB_Init(Button_TypeDef Button,
ButtonMode_TypeDef ButtonMode);
00314 void          BSP_PB_DeInit(Button_TypeDef Button);
00315 uint32_t      BSP_PB_GetState(Button_TypeDef Button);
00316 void          BSP_COM_Init(COM_TypeDef COM, UART
_HandleTypeDef *husart);
00317 void          BSP_COM_DeInit(COM_TypeDef COM, UA
RT_HandleTypeDef *huart);
00318
00319 /**
00320  * @}
00321 */
00322
00323 /**
00324  * @}
00325 */
00326
00327 /**
00328  * @}
00329 */
00330
00331 /**
00332  * @}
00333 */
00334
00335 #ifdef __cplusplus
00336 }
00337 #endif
00338
00339 #endif /* __STM32746G_DISCOVERY_H */
00340
00341 /***** (C) COPYRIGHT STMi
```

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

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file provides a set of fir
mware functions to manage LEDs,
00008 *           push-buttons and COM ports avai
lable on STM32746G-Discovery
00009 *           board(MB1191) from STMicroelect
ronics.
00010 * ****
00011 * @attention
00012 *
00013 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00014 *
00015 * Redistribution and use in source and bin
ary forms, with or without modification,
```

00016 * are permitted provided that the following conditions are met:

00017 * 1. Redistributions of source code must retain the above copyright notice,

00018 * this list of conditions and the following disclaimer.

00019 * 2. Redistributions in binary form must reproduce the above copyright notice,

00020 * this list of conditions and the following disclaimer in the documentation

00021 * and/or other materials provided with the distribution.

00022 * 3. Neither the name of STMicroelectronics nor the names of its contributors

00023 * may be used to endorse or promote products derived from this software

00024 * without specific prior written permission.

00025 *

00026 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

00027 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

00028 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

00029 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

00030 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

00031 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

00032 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

00033 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

00034 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

```
00035     * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
00036     *
00037     ****
00038     */
00039
00040 /* Includes -----
00041 -----*/
00041 #include "stm32746g_discovery.h"
00042
00043 /** @addtogroup BSP
00044     * @{
00045     */
00046
00047 /** @addtogroup STM32746G_DISCOVERY
00048     * @{
00049     */
00050
00051 /** @defgroup STM32746G_DISCOVERY_LOW_LEVEL
00052 STM32746G_DISCOVERY_LOW_LEVEL
00053     * @{
00054     */
00055 /** @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
00056 Private_TypesDefinitions STM32746G_DISCOVERY_LOW_L
00057 EVEL Private Types Definitions
00058     * @{
00059     */
00060     */
00061
00062 /** @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
00063 Private_Defines STM32746G_DISCOVERY_LOW_LEVEL Priv
00064 ate Defines
00063     * @{
00064     */
```

```
00064      */
00065 /**
00066 * @brief STM32746G DISCOVERY BSP Driver ver
00067   sion number V2.0.0
00068 #define __STM32746G_DISCO_BSP_VERSION_MAIN
00069   (0x02) /*!< [31:24] main version */
00070 #define __STM32746G_DISCO_BSP_VERSION_SUB1
00071   (0x00) /*!< [23:16] sub1 version */
00072 #define __STM32746G_DISCO_BSP_VERSION_SUB2
00073   (0x00) /*!< [15:8]  sub2 version */
00074 #define __STM32746G_DISCO_BSP_VERSION_RC
00075   (0x00) /*!< [7:0]  release candidate */
00076 #define __STM32746G_DISCO_BSP_VERSION
00077   ((__STM32746G_DISCO_BSP_VERSION_MAIN << 24) \
00078   | (__STM32746G_DISCO_BSP_VERSION_SUB1 << 16) \
00079
00080 | (__STM32746G_DISCO_BSP_VERSION_SUB2 << 8 ) \
00081 | (__STM32746G_DISCO_BSP_VERSION_RC))
00082 /**
00083 * @}
00084 */
00085 */
00086
00087 /**
00088 * @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
00089 Private_Macros STM32746G_DISCOVERY_LOW_LEVEL Priva
00090 te Macros
00091 */
00092 */
00093 /**
00094 * @}
00095 */
00096
00097 /**
00098 * @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
00099 Private_Variables STM32746G_DISCOVERY_LOW_LEVEL Pr
00100 iate Variables
```

```
00088 * @{
00089 */
00090
00091 const uint32_t GPIO_PIN[LEDn] = {LED1_PIN};
00092
00093 GPIO_TypeDef* BUTTON_PORT[BUTTONn] = {WAKEUP
 _BUTTON_GPIO_PORT,
00094                                     TAMPER
 _BUTTON_GPIO_PORT,
00095                                     KEY_BU
TTON_GPIO_PORT};
00096
00097 const uint16_t BUTTON_PIN[BUTTONn] = {WAKEUP
 _BUTTON_PIN,
00098                                     TAMPER
 _BUTTON_PIN,
00099                                     KEY_BU
TTON_PIN};
00100
00101 const uint16_t BUTTON IRQn[BUTTONn] = {WAKEU
P_BUTTON_EXTI_IRQn,
00102                                     TAMPE
R_BUTTON_EXTI_IRQn,
00103                                     KEY_B
UTTON_EXTI_IRQn};
00104
00105 USART_TypeDef* COM_USART[COMn] = {DISCOVERY_
COM1};
00106
00107 GPIO_TypeDef* COM_TX_PORT[COMn] = {DISCOVERY_
 _COM1_TX_GPIO_PORT};
00108
00109 GPIO_TypeDef* COM_RX_PORT[COMn] = {DISCOVERY_
 _COM1_RX_GPIO_PORT};
00110
00111 const uint16_t COM_TX_PIN[COMn] = {DISCOVERY_
 _COM1_TX_PIN};
```

```
00112
00113 const uint16_t COM_RX_PIN[COMn] = {DISCOVERY_
00114 _COM1_RX_PIN};
00115 const uint16_t COM_TX_AF[COMn] = {DISCOVERY_-
00116 COM1_TX_AF};
00117 const uint16_t COM_RX_AF[COMn] = {DISCOVERY_-
00118 COM1_RX_AF};
00119 static I2C_HandleTypeDef hI2cAudioHandler =
00120 {0};
00121 static I2C_HandleTypeDef hI2cExtHandler = {0
00122 };
00123 /**
00124 * @}
00125 */
00126 /** @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
00127 Private_FunctionPrototypes STM32746G_DISCOVERY_LOW_
00128 _LEVEL Private Function Prototypes
00129 * @{
00130 */
00131
00132 static HAL_StatusTypeDef I2Cx_ReadMultiple(I
00133 2C_HandleTypeDef *i2c_handler, uint8_t Addr, uint1
00134 6_t Reg, uint16_t MemAddSize, uint8_t *Buffer, ui
00135 nt16_t Length);
00136 static HAL_StatusTypeDef I2Cx_WriteMultiple(
00137 I2C_HandleTypeDef *i2c_handler, uint8_t Addr, uint1
00138 6_t Reg, uint16_t MemAddSize, uint8_t *Buffer, ui
00139 nt16_t Length);
```

```
00134 static HAL_StatusTypeDef I2Cx_IsDeviceReady(  
I2C_HandleTypeDef *i2c_handler, uint16_t DevAddress,  
uint32_t Trials);  
00135 static void I2Cx_Error(I2C_HandleTypeDef *i2c_handler, uint8_t Addr);  
00136  
00137 /* AUDIO IO functions */  
00138 void AUDIO_IO_Init(void);  
00139 void AUDIO_IO_DeInit(void);  
00140 void AUDIO_IO_Write(uint8_t Addr,  
uint16_t Reg, uint16_t Value);  
00141 uint16_t AUDIO_IO_Read(uint8_t Addr,  
uint16_t Reg);  
00142 void AUDIO_IO_Delay(uint32_t Delay);  
00143  
00144 /* TOUCHSCREEN IO functions */  
00145 void TS_IO_Init(void);  
00146 void TS_IO_Write(uint8_t Addr, uint8_t Reg, uint8_t Value);  
00147 uint8_t TS_IO_Read(uint8_t Addr, uint8_t Reg);  
00148 void TS_IO_Delay(uint32_t Delay);  
00149  
00150 /* CAMERA IO functions */  
00151 void CAMERA_IO_Init(void);  
00152 void CAMERA_Delay(uint32_t Delay);  
;  
00153 void CAMERA_IO_Write(uint8_t Addr,  
uint8_t Reg, uint8_t Value);  
00154 uint8_t CAMERA_IO_Read(uint8_t Addr,  
uint8_t Reg);  
00155  
00156 /* I2C EEPROM IO function */  
00157 void EEPROM_IO_Init(void);  
00158 HAL_StatusTypeDef EEPROM_IO_WriteData(uint16_t DevAddress, uint16_t MemAddress, uint8_t* pBuffer);
```

```
fffer, uint32_t BufferSize);
00159 HAL_StatusTypeDef    EEPROM_IO_ReadData(uint1
6_t DevAddress, uint16_t MemAddress, uint8_t* pBuffer,
uint32_t BufferSize);
00160 HAL_StatusTypeDef    EEPROM_IO_IsDeviceReady(
uint16_t DevAddress, uint32_t Trials);
00161 /**
00162 * @}
00163 */
00164
00165 /** @defgroup STM32746G_DISCOVERY_LOW_LEVEL_
Exported_Functions STM32746G_DISCOVERY_LOW_LEVELST
M32746G_DISCOVERY_LOW_LEVEL Exported Functions
00166 * @{
00167 */
00168
00169 /**
00170 * @brief This method returns the STM32746
G DISCOVERY BSP Driver revision
00171 * @retval version: 0xXYZR (8bits for each
decimal, R for RC)
00172 */
00173 uint32_t BSP_GetVersion(void)
00174 {
00175     return __STM32746G_DISCO_BSP_VERSION;
00176 }
00177
00178 /**
00179 * @brief Configures LED on GPIO.
00180 * @param Led: LED to be configured.
00181 *           This parameter can be one of th
e following values:
00182     *          @arg LED1
00183     * @retval None
00184 */
00185 void BSP_LED_Init(Led_TypeDef Led)
00186 {
```

```
00187     GPIO_InitTypeDef    gpio_init_structure;
00188     GPIO_TypeDef*        gpio_led;
00189
00190     if (Led == LED1)
00191     {
00192         gpio_led = LED1_GPIO_PORT;
00193         /* Enable the GPIO_LED clock */
00194         LED1_GPIO_CLK_ENABLE();
00195
00196         /* Configure the GPIO_LED pin */
00197         gpio_init_structure.Pin = GPIO_PIN[Led];
00198         gpio_init_structure.Mode = GPIO_MODE_OUT
PUT_PP;
00199         gpio_init_structure.Pull = GPIO_PULLUP;
00200         gpio_init_structure.Speed = GPIO_SPEED_H
IGH;
00201
00202         HAL_GPIO_Init(gpio_led, &gpio_init_struct
ture);
00203
00204         /* By default, turn off LED */
00205         HAL_GPIO_WritePin(gpio_led, GPIO_PIN[Led]
], GPIO_PIN_RESET);
00206     }
00207 }
00208
00209 /**
00210 * @brief DeInit LEDs.
00211 * @param Led: LED to be configured.
00212 *           This parameter can be one of th
e following values:
00213 *           @arg LED1
00214 *           @note Led DeInit does not disable the GP
IO clock
00215 *           @retval None
00216 */
00217 void BSP_LED_DeInit(Led_TypeDef Led)
```



```
00250
00251 /**
00252     * @brief Turns selected LED Off.
00253     * @param Led: LED to be set off
00254     *           This parameter can be one of th
e following values:
00255     *           @arg LED1
00256     * @retval None
00257 */
00258 void BSP_LED_Off(Led_TypeDef Led)
00259 {
00260     GPIO_TypeDef*      gpio_led;
00261
00262     if (Led == LED1) /* Switch Off LED connect
ed to GPIO */
00263     {
00264         gpio_led = LED1_GPIO_PORT;
00265         HAL_GPIO_WritePin(gpio_led, GPIO_PIN[Led
], GPIO_PIN_RESET);
00266     }
00267 }
00268
00269 /**
00270     * @brief Toggles the selected LED.
00271     * @param Led: LED to be toggled
00272     *           This parameter can be one of th
e following values:
00273     *           @arg LED1
00274     * @retval None
00275 */
00276 void BSP_LED_Toggle(Led_TypeDef Led)
00277 {
00278     GPIO_TypeDef*      gpio_led;
00279
00280     if (Led == LED1)      /* Toggle LED connec
ted to GPIO */
00281     {
```

```
00282     gpio_led = LED1_GPIO_PORT;
00283     HAL_GPIO_TogglePin(gpio_led, GPIO_PIN[Le
d]);
00284 }
00285 }
00286
00287 /**
00288 * @brief Configures button GPIO and EXTI
Line.
00289 * @param Button: Button to be configured
00290 * This parameter can be one of th
e following values:
00291 * @arg BUTTON_WAKEUP: Wakeup P
ush Button
00292 * @arg BUTTON_TAMPER: Tamper P
ush Button
00293 * @arg BUTTON_KEY: Key Push Bu
tton
00294 * @param ButtonMode: Button mode
00295 * This parameter can be one of th
e following values:
00296 * @arg BUTTON_MODE_GPIO: Butto
n will be used as simple IO
00297 * @arg BUTTON_MODE_EXTI: Butto
n will be connected to EXTI line
00298 * with
interrupt generation capability
00299 * @note On STM32746G-Discovery board, the
three buttons (Wakeup, Tamper and key buttons)
00300 * are mapped on the same push button
named "User"
00301 * on the board serigraphy.
00302 * @retval None
00303 */
00304 void BSP_PB_Init(Button_TypeDef Button, Butt
onMode_TypeDef ButtonMode)
00305 {
```

```
00306     GPIO_InitTypeDef gpio_init_structure;
00307
00308     /* Enable the BUTTON clock */
00309     BUTTONx_GPIO_CLK_ENABLE(Button);
00310
00311     if(ButtonMode == BUTTON_MODE_GPIO)
00312     {
00313         /* Configure Button pin as input */
00314         gpio_init_structure.Pin = BUTTON_PIN[Button];
00315
00316         gpio_init_structure.Mode = GPIO_MODE_INP
00317 UT;
00318         gpio_init_structure.Pull = GPIO_NOPULL;
00319         gpio_init_structure.Speed = GPIO_SPEED_F
00320 AST;
00321         HAL_GPIO_Init(BUTTON_PORT[Button], &gpio
00322 _init_structure);
00323     }
00324
00325     if(ButtonMode == BUTTON_MODE_EXTI)
00326     {
00327         /* Configure Button pin as input with Ex
00328 ternal interrupt */
00329         gpio_init_structure.Pin = BUTTON_PIN[Button];
00330
00331         gpio_init_structure.Pull = GPIO_NOPULL;
00332         gpio_init_structure.Speed = GPIO_SPEED_F
00333 AST;
00334
00335         if(Button != BUTTON_WAKEUP)
00336         {
00337             gpio_init_structure.Mode = GPIO_MODE_I
00338 T_FALLING;
00339         }
00340         else
00341         {
00342             gpio_init_structure.Mode = GPIO_MODE_I
```

```
    T_RISING;
00335      }
00336
00337      HAL_GPIO_Init(BUTTON_PORT[Button], &gpio
 _init_structure);
00338
00339      /* Enable and set Button EXTI Interrupt
 to the lowest priority */
00340      HAL_NVIC_SetPriority((IRQn_Type)(BUTTON_
 IRQn[Button]), 0x0F, 0x00);
00341      HAL_NVIC_EnableIRQ((IRQn_Type)(BUTTON_IR
 Qn[Button]));
00342  }
00343 }
00344
00345 /**
00346  * @brief Push Button DeInit.
00347  * @param Button: Button to be configured
00348  *                 This parameter can be one of th
e following values:
00349  *                 @arg BUTTON_WAKEUP: Wakeup P
ush Button
00350  *                 @arg BUTTON_TAMPER: Tamper P
ush Button
00351  *                 @arg BUTTON_KEY: Key Push Bu
tton
00352  * @note On STM32746G-Discovery board, the
three buttons (Wakeups, Tamper and key buttons)
00353  *       are mapped on the same push button
named "User"
00354  *       on the board serigraphy.
00355  * @note PB DeInit does not disable the GPI
O clock
00356  * @retval None
00357 */
00358 void BSP_PB_DeInit(Button_TypeDef Button)
00359 {
```

```
00360     GPIO_InitTypeDef gpio_init_structure;
00361
00362     gpio_init_structure.Pin = BUTTON_PIN[Button];
00363     HAL_NVIC_DisableIRQ((IRQn_Type)(BUTTON IRQn[Button]));
00364     HAL_GPIO_DeInit(BUTTON_PORT[Button], gpio_init_structure.Pin);
00365 }
00366
00367
00368 /**
00369 * @brief Returns the selected button state.
00370 * @param Button: Button to be checked
00371 * This parameter can be one of the following values:
00372 * @arg BUTTON_WAKEUP: Wakeup Push Button
00373 * @arg BUTTON_TAMPER: Tamper Push Button
00374 * @arg BUTTON_KEY: Key Push Button
00375 * @note On STM32746G-Discovery board, the three buttons (Wakeup, Tamper and key buttons)
00376 * are mapped on the same push button named "User"
00377 * on the board serigraphy.
00378 * @retval The Button GPIO pin value
00379 */
00380 uint32_t BSP_PB_GetState(Button_TypeDef Button)
00381 {
00382     return HAL_GPIO_ReadPin(BUTTON_PORT[Button],
00383     BUTTON_PIN[Button]);
00384 }
```

```
00385 /**
00386     * @brief Configures COM port.
00387     * @param COM: COM port to be configured.
00388     *             This parameter can be one of the
00389     *             following values:
00390     *             @arg COM1
00391     *             @arg COM2
00392     * @param huart: Pointer to a UART_HandleTypeDef
00393     *             structure that contains the
00394     *             configuration information
00395     *             for the specified USART peripheral.
00396     * @retval None
00397     */
00398 void BSP_COM_Init(COM_TypeDef COM, UART_HandleTypeDef *huart)
00399 {
00400     GPIO_InitTypeDef gpio_init_structure;
00401
00402     /* Enable GPIO clock */
00403     DISCOVERY_COMx_TX_GPIO_CLK_ENABLE(COM);
00404     DISCOVERY_COMx_RX_GPIO_CLK_ENABLE(COM);
00405
00406     /* Enable USART clock */
00407     DISCOVERY_COMx_CLK_ENABLE(COM);
00408
00409     /* Configure USART Tx as alternate function */
00410     gpio_init_structure.Pin = COM_TX_PIN[COM];
00411     gpio_init_structure.Mode = GPIO_MODE_AF_PP;
00412     gpio_init_structure.Speed = GPIO_SPEED_FAST;
00413     gpio_init_structure.Pull = GPIO_PULLUP;
00414     gpio_init_structure.Alternate = COM_TX_AF[COM];
00415
00416     HAL_GPIO_Init(COM_TX_PORT[COM], &gpio_init_structure);
```

```
00413
00414     /* Configure USART Rx as alternate function */
00415     gpio_init_structure.Pin = COM_RX_PIN[COM];
00416     gpio_init_structure.Mode = GPIO_MODE_AF_PP
00417 ;
00418     gpio_init_structure.Alternate = COM_RX_AF[COM];
00419     HAL_GPIO_Init(COM_RX_PORT[COM], &gpio_init_
00420 _structure);
00421
00422     /* USART configuration */
00423     huart->Instance = COM_USART[COM];
00424     HAL_UART_Init(huart);
00425 }
00426 /**
00427 * @brief DeInit COM port.
00428 * @param COM: COM port to be configured.
00429 *           This parameter can be one of the
00430 * following values:
00431 *           @arg COM1
00432 *           @arg COM2
00433 * @param huart: Pointer to a UART_HandleTypeDef
00434 *               structure that contains the
00435 *               configuration information
00436 *               for the specified USART peripheral.
00437 * @retval None
00438 */
00439 void BSP_COM_DeInit(COM_HandleTypeDef COM, UART_Ha
ndleTypeDef *huart)
00440 {
00441     /* USART configuration */
00442     huart->Instance = COM_USART[COM];
00443     HAL_UART_DeInit(huart);
00444
00445     /* Enable USART clock */
```

```
00442     DISCOVERY_COMx_CLK_DISABLE(COM);
00443
00444     /* DeInit GPIO pins can be done in the app
00445      application
00446      (by surcharging this __weak function) */
00447
00448     /* GPIO pins clock, DMA clock can be shut
00449      down in the application
00450      by surcharging this __weak function */
00451 }
00452
00453 ****
00454 ****
00455 **** I2C Routine
00456 /**
00457 * @brief Initializes I2C MSP.
00458 * @param i2c_handler : I2C handler
00459 * @retval None
00460 */
00461 static void I2Cx_MspInit(I2C_HandleTypeDef *i2c_handler)
00462 {
00463     GPIO_InitTypeDef gpio_init_structure;
00464
00465     if (i2c_handler == (I2C_HandleTypeDef*)(&h
00466 I2cAudioHandler))
00467     {
00468         /* AUDIO and LCD I2C MSP init */
00469
00470         /*** Configure the GPIOs ***/
00471
00472         /* Enable GPIO clock */
```

```
00471     DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_CLK_EN
ABLE();
00472
00473     /* Configure I2C Tx as alternate function */
00474     gpio_init_structure.Pin = DISCOVERY_AUDIO_I2Cx_SCL_PIN;
00475     gpio_init_structure.Mode = GPIO_MODE_AF_OD;
00476     gpio_init_structure.Pull = GPIO_NOPULL;
00477     gpio_init_structure.Speed = GPIO_SPEED_FAST;
00478     gpio_init_structure.Alternate = DISCOVERY_AUDIO_I2Cx_SCL_SDA_AF;
00479     HAL_GPIO_Init(DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PORT, &gpio_init_structure);
00480
00481     /* Configure I2C Rx as alternate function */
00482     gpio_init_structure.Pin = DISCOVERY_AUDIO_I2Cx_SDA_PIN;
00483     HAL_GPIO_Init(DISCOVERY_AUDIO_I2Cx_SCL_SDA_GPIO_PORT, &gpio_init_structure);
00484
00485     /*** Configure the I2C peripheral ***/
00486     /* Enable I2C clock */
00487     DISCOVERY_AUDIO_I2Cx_CLK_ENABLE();
00488
00489     /* Force the I2C peripheral clock reset */
00490     DISCOVERY_AUDIO_I2Cx_FORCE_RESET();
00491
00492     /* Release the I2C peripheral clock reset */
00493     DISCOVERY_AUDIO_I2Cx_RELEASE_RESET();
00494
00495     /* Enable and set I2Cx Interrupt to a lo
```

```
wer priority */
00496     HAL_NVIC_SetPriority(DISCOVERY_AUDIO_I2C
x_EV IRQn, 0x0F, 0);
00497     HAL_NVIC_EnableIRQ(DISCOVERY_AUDIO_I2Cx_
EV IRQn);
00498
00499     /* Enable and set I2Cx Interrupt to a lo
wer priority */
00500     HAL_NVIC_SetPriority(DISCOVERY_AUDIO_I2C
x_ER IRQn, 0x0F, 0);
00501     HAL_NVIC_EnableIRQ(DISCOVERY_AUDIO_I2Cx_
ER IRQn);
00502 }
00503 else
00504 {
00505     /* External, camera and Arduino connecto
r I2C MSP init */
00506
00507     /*** Configure the GPIOs ***/
00508     /* Enable GPIO clock */
00509     DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_CLK_ENAB
LE();
00510
00511     /* Configure I2C Tx as alternate functio
n */
00512     gpio_init_structure.Pin = DISCOVERY_EXT_
I2Cx_SCL_PIN;
00513     gpio_init_structure.Mode = GPIO_MODE_AF_
OD;
00514     gpio_init_structure.Pull = GPIO_NOPULL;
00515     gpio_init_structure.Speed = GPIO_SPEED_F
AST;
00516     gpio_init_structure.Alternate = DISCOVER
Y_EXT_I2Cx_SCL_SDA_AF;
00517     HAL_GPIO_Init(DISCOVERY_EXT_I2Cx_SCL_SDA
_GPIO_PORT, &gpio_init_structure);
00518
```

```
00519     /* Configure I2C Rx as alternate function */
00520     gpio_init_structure.Pin = DISCOVERY_EXT_I2Cx_SDA_PIN;
00521     HAL_GPIO_Init(DISCOVERY_EXT_I2Cx_SCL_SDA_GPIO_PORT, &gpio_init_structure);
00522
00523     /*** Configure the I2C peripheral ***/
00524     /* Enable I2C clock */
00525     DISCOVERY_EXT_I2Cx_CLK_ENABLE();
00526
00527     /* Force the I2C peripheral clock reset */
00528     DISCOVERY_EXT_I2Cx_FORCE_RESET();
00529
00530     /* Release the I2C peripheral clock reset */
00531     DISCOVERY_EXT_I2Cx_RELEASE_RESET();
00532
00533     /* Enable and set I2Cx Interrupt to a lower priority */
00534     HAL_NVIC_SetPriority(DISCOVERY_EXT_I2Cx_EV IRQn, 0x0F, 0);
00535     HAL_NVIC_EnableIRQ(DISCOVERY_EXT_I2Cx_EV IRQn);
00536
00537     /* Enable and set I2Cx Interrupt to a lower priority */
00538     HAL_NVIC_SetPriority(DISCOVERY_EXT_I2Cx_ER IRQn, 0x0F, 0);
00539     HAL_NVIC_EnableIRQ(DISCOVERY_EXT_I2Cx_ER IRQn);
00540 }
00541 }
00542
00543 /**
00544 * @brief Initializes I2C HAL.
```

```
00545     * @param i2c_handler : I2C handler
00546     * @retval None
00547     */
00548 static void I2Cx_Init(I2C_HandleTypeDef *i2c
00549 _handler)
00550     if(HAL_I2C_GetState(i2c_handler) == HAL_I2
00551 C_STATE_RESET)
00551     {
00552         if (i2c_handler == (I2C_HandleTypeDef*)(&
00553 hI2cAudioHandler))
00553     {
00554         /* Audio and LCD I2C configuration */
00555         i2c_handler->Instance = DISCOVERY_AUDI
00556 O_I2Cx;
00556     }
00557     else
00558     {
00559         /* External, camera and Arduino connec
00560 tor I2C configuration */
00561         i2c_handler->Instance = DISCOVERY_EXT_
00562 I2Cx;
00563     }
00564     i2c_handler->Init.Timing          = DIS
00565 COVERY_I2Cx_TIMING;
00566     i2c_handler->Init.OwnAddress1    = 0;
00567     i2c_handler->Init.AddressingMode = I2C
00568 _ADDRESSINGMODE_7BIT;
00569     i2c_handler->Init.DualAddressMode = I2C
00570 _DUALADDRESS_DISABLE;
00571     i2c_handler->Init.OwnAddress2    = 0;
00572     i2c_handler->Init.GeneralCallMode = I2C
00573 _GENERALCALL_DISABLE;
00574     i2c_handler->Init.NoStretchMode  = I2C
00575 _NOSTRETCH_DISABLE;
00576
00577     /* Init the I2C */
```

```
00571     I2Cx_MspInit(i2c_handler);
00572     HAL_I2C_Init(i2c_handler);
00573 }
00574 }
00575
00576 /**
00577 * @brief Reads multiple data.
00578 * @param i2c_handler : I2C handler
00579 * @param Addr: I2C address
00580 * @param Reg: Reg address
00581 * @param MemAddress: Memory address
00582 * @param Buffer: Pointer to data buffer
00583 * @param Length: Length of the data
00584 * @retval Number of read data
00585 */
00586 static HAL_StatusTypeDef I2Cx_ReadMultiple(I
2C_HandleTypeDef *i2c_handler,
00587
00588     u
00589     int8_t Addr,
00590
00591     u
00592     int16_t Reg,
00593
00594     u
00595     int16_t MemAddress,
00596
00597     u
00598     int8_t *Buffer,
00599
00600     u
00601     int16_t Length)
00602 {
00603     HAL_StatusTypeDef status = HAL_OK;
00604
00605     status = HAL_I2C_Mem_Read(i2c_handler, Add
r, (uint16_t)Reg, MemAddress, Buffer, Length, 1000
);
00606
00607     /* Check the communication status */
00608     if(status != HAL_OK)
00609     {
```

```
00600     /* I2C error occurred */
00601     I2Cx_Error(i2c_handler, Addr);
00602 }
00603 return status;
00604 }
00605
00606 /**
00607  * @brief Writes a value in a register of
00608  * the device through BUS in using DMA mode.
00609  * @param i2c_handler : I2C handler
00610  * @param Addr: Device address on BUS Bus.
00611  * @param Reg: The target register address
00612  * to write
00613  * @param MemAddress: Memory address
00614  * @param Buffer: The target register valu
00615  * e to be written
00616  * @param Length: buffer size to be written
00617
00618  * @retval HAL status
00619 */
00620 static HAL_StatusTypeDef I2Cx_WriteMultiple(
00621 I2C_HandleTypeDef *i2c_handler,
00622
00623 uint8_t Addr,
00624
00625 uint16_t Reg,
00626
00627 uint16_t MemAddress,
00628
00629 uint8_t *Buffer,
00630
00631 uint16_t Length)
00632 {
00633     HAL_StatusTypeDef status = HAL_OK;
00634
00635     status = HAL_I2C_Mem_Write(i2c_handler, Ad
```

```
dr, (uint16_t)Reg, MemAddress, Buffer, Length, 100
0);
00626
00627     /* Check the communication status */
00628     if(status != HAL_OK)
00629     {
00630         /* Re-Initiaize the I2C Bus */
00631         I2Cx_Error(i2c_handler, Addr);
00632     }
00633     return status;
00634 }
00635
00636 /**
00637     * @brief Checks if target device is ready
00638     * for communication.
00639     * @param i2c_handler : I2C handler
00640     * @param DevAddress: Target device address
00641     * @param Trials: Number of trials
00642     * @retval HAL status
00643 */
00644 static HAL_StatusTypeDef I2Cx_IsDeviceReady(
I2C_HandleTypeDef *i2c_handler, uint16_t DevAddres
s, uint32_t Trials)
00645 {
00646     return (HAL_I2C_IsDeviceReady(i2c_handler,
DevAddress, Trials, 1000));
00647 }
00648
00649 /**
00650     * @brief Manages error callback by re-init
00651     * alizing I2C.
00652     * @param i2c_handler : I2C handler
00653     * @param Addr: I2C Address
00654     * @retval None
```

```
00654     */
00655 static void I2Cx_Error(I2C_HandleTypeDef *i2
c_handler, uint8_t Addr)
00656 {
00657     /* De-initialize the I2C communication bus
 */
00658     HAL_I2C_DeInit(i2c_handler);
00659
00660     /* Re-Initialize the I2C communication bus
 */
00661     I2Cx_Init(i2c_handler);
00662 }
00663
00664 /*****
00665 *****          LINK OPERATIONS
00666 *****/
00667
00668 /*****          LINK AUDI
0 *****/
00669
00670 /**
00671     * @brief  Initializes Audio low level.
00672     * @retval None
00673 */
00674 void AUDIO_IO_Init(void)
00675 {
00676     I2Cx_Init(&hI2cAudioHandler);
00677 }
00678
00679 /**
00680     * @brief  Deinitializes Audio low level.
00681     * @retval None
00682 */
00683 void AUDIO_IO_DeInit(void)
00684 {
```

```
00685 }
00686
00687 /**
00688     * @brief Writes a single data.
00689     * @param Addr: I2C address
00690     * @param Reg: Reg address
00691     * @param Value: Data to be written
00692     * @retval None
00693 */
00694 void AUDIO_IO_Write(uint8_t Addr, uint16_t Reg, uint16_t Value)
00695 {
00696     uint16_t tmp = Value;
00697
00698     Value = ((uint16_t)(tmp >> 8) & 0x00FF);
00699
00700     Value |= ((uint16_t)(tmp << 8)& 0xFF00);
00701
00702     I2Cx_WriteMultiple(&hI2cAudioHandler, Addr, Reg, I2C_MEMADD_SIZE_16BIT, (uint8_t*)&Value, 2);
00703 }
00704
00705 /**
00706     * @brief Reads a single data.
00707     * @param Addr: I2C address
00708     * @param Reg: Reg address
00709     * @retval Data to be read
00710 */
00711 uint16_t AUDIO_IO_Read(uint8_t Addr, uint16_t Reg)
00712 {
00713     uint16_t read_value = 0, tmp = 0;
00714
00715     I2Cx_ReadMultiple(&hI2cAudioHandler, Addr, Reg, I2C_MEMADD_SIZE_16BIT, (uint8_t*)&read_value, 2);
00716
```

```
00717     tmp = ((uint16_t)(read_value >> 8) & 0x00F
F);
00718
00719     tmp |= ((uint16_t)(read_value << 8)& 0xFF0
0);
00720
00721     read_value = tmp;
00722
00723     return read_value;
00724 }
00725
00726 /**
00727     * @brief  AUDIO Codec delay
00728     * @param  Delay: Delay in ms
00729     * @retval None
00730 */
00731 void AUDIO_IO_Delay(uint32_t Delay)
00732 {
00733     HAL_Delay(Delay);
00734 }
00735
00736 /***** LINK CAME
RA *****/
00737
00738 /**
00739     * @brief  Initializes Camera low level.
00740     * @retval None
00741 */
00742 void CAMERA_IO_Init(void)
00743 {
00744     I2Cx_Init(&hI2cExtHandler);
00745 }
00746
00747 /**
00748     * @brief  Camera writes single data.
00749     * @param  Addr: I2C address
00750     * @param  Reg: Register address
```

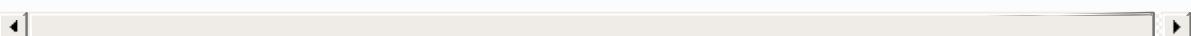
```
00751     * @param Value: Data to be written
00752     * @retval None
00753     */
00754 void CAMERA_IO_Write(uint8_t Addr, uint8_t Reg, uint8_t Value)
00755 {
00756     I2Cx_WriteMultiple(&hI2cExtHandler, Addr,
00757     (uint16_t)Reg, I2C_MEMADD_SIZE_8BIT, (uint8_t*)&Value, 1);
00758 }
00759 /**
00760     * @brief Camera reads single data.
00761     * @param Addr: I2C address
00762     * @param Reg: Register address
00763     * @retval Read data
00764 */
00765 uint8_t CAMERA_IO_Read(uint8_t Addr, uint8_t Reg)
00766 {
00767     uint8_t read_value = 0;
00768
00769     I2Cx_ReadMultiple(&hI2cExtHandler, Addr, Reg,
00770     I2C_MEMADD_SIZE_8BIT, (uint8_t*)&read_value, 1);
00771
00772     return read_value;
00773 }
00774 /**
00775     * @brief Camera delay
00776     * @param Delay: Delay in ms
00777     * @retval None
00778 */
00779 void CAMERA_Delay(uint32_t Delay)
00780 {
00781     HAL_Delay(Delay);
```

```
00782 }
00783
00784 /***** LINK I2C E
EEPROM *****/
00785
00786 /**
00787 * @brief Initializes peripherals used by
the I2C EEPROM driver.
00788 * @retval None
00789 */
00790 void EEPROM_IO_Init(void)
00791 {
00792     I2Cx_Init(&hI2cExtHandler);
00793 }
00794
00795 /**
00796 * @brief Write data to I2C EEPROM driver
in using DMA channel.
00797 * @param DevAddress: Target device address
00798 * @param MemAddress: Internal memory address
00799 * @param pBuffer: Pointer to data buffer
00800 * @param BufferSize: Amount of data to be
sent
00801 * @retval HAL status
00802 */
00803 HAL_StatusTypeDef EEPROM_IO_WriteData(uint16
_t DevAddress, uint16_t MemAddress, uint8_t* pBuffer,
uint32_t BufferSize)
00804 {
00805     return (I2Cx_WriteMultiple(&hI2cExtHandler
, DevAddress, MemAddress, I2C_MEMADD_SIZE_16BIT, pBuffer,
BufferSize));
00806 }
00807
00808 /**
```

```
00809     * @brief  Read data from I2C EEPROM driver  
00810     * @param DevAddress: Target device address  
  
00811     * @param MemAddress: Internal memory address  
00812     * @param pBuffer: Pointer to data buffer  
00813     * @param BufferSize: Amount of data to be  
00814     * @retval HAL status  
00815     */  
00816 HAL_StatusTypeDef EEPROM_IO_ReadData(uint16_t DevAddress, uint16_t MemAddress, uint8_t* pBuffer, uint32_t BufferSize)  
00817 {  
00818     return (I2Cx_ReadMultiple(&hI2cExtHandler,  
00819             DevAddress, MemAddress, I2C_MEMADD_SIZE_16BIT, pBuffer,  
00820             BufferSize));  
00821 }  
00822  
00823     * @brief Checks if target device is ready  
00824     * @note This function is used with Memory  
00825     * @param DevAddress: Target device address  
00826     * @param Trials: Number of trials  
00827     * @retval HAL status  
00828 HAL_StatusTypeDef EEPROM_IO_IsDeviceReady(uint16_t DevAddress, uint32_t Trials)  
00829 {  
00830     return (I2Cx_IsDeviceReady(&hI2cExtHandler  
00831             , DevAddress, Trials));  
00832 }
```

```
00833 /***** LINK TOUC
HSCREEN *****/
00834
00835 /**
00836     * @brief Initializes Touchscreen low level.
00837     * @retval None
00838 */
00839 void TS_IO_Init(void)
00840 {
00841     I2Cx_Init(&hI2cAudioHandler);
00842 }
00843
00844 /**
00845     * @brief Writes a single data.
00846     * @param Addr: I2C address
00847     * @param Reg: Reg address
00848     * @param Value: Data to be written
00849     * @retval None
00850 */
00851 void TS_IO_Write(uint8_t Addr, uint8_t Reg,
00852 uint8_t Value)
00853 {
00854     I2Cx_WriteMultiple(&hI2cAudioHandler, Addr,
00855     , (uint16_t)Reg, I2C_MEMADD_SIZE_8BIT,(uint8_t*)&Value, 1);
00856 }
00857
00858 /**
00859     * @brief Reads a single data.
00860     * @param Addr: I2C address
00861     * @param Reg: Reg address
00862     * @retval Data to be read
00863 */
00864 uint8_t TS_IO_Read(uint8_t Addr, uint8_t Reg
00865 )
00866 {
```

```
00864     uint8_t read_value = 0;
00865
00866     I2Cx_ReadMultiple(&hI2cAudioHandler, Addr,
00867     Reg, I2C_MEMADD_SIZE_8BIT, (uint8_t*)&read_value,
00868     1);
00869 }
00870
00871 /**
00872 * @brief TS delay
00873 * @param Delay: Delay in ms
00874 * @retval None
00875 */
00876 void TS_IO_Delay(uint32_t Delay)
00877 {
00878     HAL_Delay(Delay);
00879 }
00880
00881 /**
00882 * @}
00883 */
00884
00885 /**
00886 * @}
00887 */
00888
00889 /**
00890 * @}
00891 */
00892
00893 /**
00894 * @}
00895 */
00896
00897 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```



Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_audio.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_audio.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains the common defines and functions prototypes for
00008 *           the stm32746g_discovery_audio.c driver.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STMicroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and binary forms, with or without modification,
00015 * are permitted provided that the following conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038
00039 /* Define to prevent recursive inclusion ---
00040 -----*/
00040 #ifndef __STM32746G_DISCOVERY_AUDIO_H
00041 #define __STM32746G_DISCOVERY_AUDIO_H
00042
00043 #ifdef __cplusplus
00044     extern "C" {
00045 #endif
00046
00047 /* Includes -----
00048 -----*/
00048 /* Include audio component Driver */
00049 #include "../Components/wm8994/wm8994.h"
00050 #include "stm32746g_discovery.h"
00051
00052 /** @addtogroup BSP
00053     * @{
00054     */
00055
00056 /** @addtogroup STM32746G_DISCOVERY
00057     * @{
00058     */
00059
00060 /** @defgroup STM32746G_DISCOVERY_AUDIO STM3
00061     * @{
00062     */
00063
00064 /** @defgroup STM32746G_DISCOVERY_AUDIO_Expo
00065     * @{
00065     *
```

```

00066      */
00067  /**
00068  * @}
00069  */
00070
00071 /** @defgroup STM32746G_DISCOVERY_AUDIO_Expo
rted_Constants STM32746G_DISCOVERY_AUDIO Exported
Constants
00072  * @{
00073  */
00074
00075 /*-----
-----*
00076                               USER SAI defines p
arameters
00077 -----
-----*/
00078 /* CODEC_AudioFrame_SLOT_TDMMode
00079     In W8994 codec the Audio frame contains 4
slots : TDM Mode
00080     TDM format :
00081     +-----|-----|-----|-----+
-----|-----+-----+
00082     | CODEC_SLOT0 Left | CODEC_SLOT1 Left | C
ODEC_SLOT0 Right | CODEC_SLOT1 Right |
00083     +-----|-----|-----+
-----+
00084 */
00085 /* To have 2 separate audio stream in Both h
eadphone and speaker the 4 slot must be activated
*/
00086 #define CODEC_AUDIOFRAME_SLOT_0123
    SAI_SLOTACTIVE_0 | SAI_SLOTACTIVE_1 | SAI_SLOTACT
IVE_2 | SAI_SLOTACTIVE_3
00087 /* To have an audio stream in headphone only
SAI Slot 0 and Slot 2 must be activated */
00088 #define CODEC_AUDIOFRAME_SLOT_02

```

```
    SAI_SLOTACTIVE_0 | SAI_SLOTACTIVE_2
00089 /* To have an audio stream in speaker only S
AI Slot 1 and Slot 3 must be activated */
00090 #define CODEC_AUDIOFRAME_SLOT_13
    SAI_SLOTACTIVE_1 | SAI_SLOTACTIVE_3
00091
00092 /* SAI OUT peripheral configuration defines
*/
00093 #define AUDIO_OUT_SAIx
    SAI2_Block_A
00094 #define AUDIO_OUT_SAIx_CLK_ENABLE()
    __HAL_RCC_SAI2_CLK_ENABLE()
00095 #define AUDIO_OUT_SAIx_CLK_DISABLE()
    __HAL_RCC_SAI2_CLK_DISABLE()
00096 #define AUDIO_OUT_SAIx_SCK_AF
    GPIO_AF10_SAI2
00097 #define AUDIO_OUT_SAIx_FS_SD_MCLK_AF
    GPIO_AF10_SAI2
00098
00099 #define AUDIO_OUT_SAIx_MCLK_ENABLE()
    __HAL_RCC_GPIOI_CLK_ENABLE()
00100 #define AUDIO_OUT_SAIx_MCLK_GPIO_PORT
    GPIOI
00101 #define AUDIO_OUT_SAIx_MCLK_PIN
    GPIO_PIN_4
00102 #define AUDIO_OUT_SAIx_SCK_SD_ENABLE()
    __HAL_RCC_GPIOI_CLK_ENABLE()
00103 #define AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT
    GPIOI
00104 #define AUDIO_OUT_SAIx_SCK_PIN
    GPIO_PIN_5
00105 #define AUDIO_OUT_SAIx_SD_PIN
    GPIO_PIN_6
00106 #define AUDIO_OUT_SAIx_FS_ENABLE()
    __HAL_RCC_GPIOI_CLK_ENABLE()
00107 #define AUDIO_OUT_SAIx_FS_GPIO_PORT
    GPIOI
```

```
00108 #define AUDIO_OUT_SAIx_FS_PIN  
        GPIO_PIN_7  
00109  
00110 /* SAI DMA Stream definitions */  
00111 #define AUDIO_OUT_SAIx_DMAx_CLK_ENABLE()  
        __HAL_RCC_DMA2_CLK_ENABLE()  
00112 #define AUDIO_OUT_SAIx_DMAx_STREAM  
        DMA2_Stream4  
00113 #define AUDIO_OUT_SAIx_DMAx_CHANNEL  
        DMA_CHANNEL_3  
00114 #define AUDIO_OUT_SAIx_DMAx_IRQ  
        DMA2_Stream4_IRQn  
00115 #define AUDIO_OUT_SAIx_DMAx_PERIPH_DATA_SIZE  
        DMA_PDATAALIGN_HALFWORD  
00116 #define AUDIO_OUT_SAIx_DMAx_MEM_DATA_SIZE  
        DMA_MDATAALIGN_HALFWORD  
00117 #define DMA_MAX_SZE  
        ((uint16_t)0xFFFF)  
00118  
00119 #define AUDIO_OUT_SAIx_DMAx_IRQHandler  
        DMA2_Stream4_IRQHandler  
00120  
00121 /* Select the interrupt preemption priority  
for the DMA interrupt */  
00122 #define AUDIO_OUT_IRQ_PREPRI  
        ((uint32_t)0x0E) /* Select the preemption p  
riority level(0 is the highest) */  
00123  
00124 /*-----  
-----  
00125                                     AUDIO IN CONFIGURATI  
ON  
00126 -----*/  
00127 /* SAI IN peripheral configuration defines */  
00128 #define AUDIO_IN_SAIx
```

```
    SAI2_Block_B
00129 #define AUDIO_IN_SAIx_CLK_ENABLE()
        __HAL_RCC_SAI2_CLK_ENABLE()
00130 #define AUDIO_IN_SAIx_CLK_DISABLE()
        __HAL_RCC_SAI2_CLK_DISABLE()
00131 #define AUDIO_IN_SAIx_SD_AF
        GPIO_AF10_SAI2
00132
00133 #define AUDIO_IN_SAIx_SD_ENABLE()
        __HAL_RCC_GPIOG_CLK_ENABLE()
00134 #define AUDIO_IN_SAIx_SD_GPIO_PORT
        GPIOG
00135 #define AUDIO_IN_SAIx_SD_PIN
        GPIO_PIN_10
00136
00137 #define AUDIO_IN_INT_GPIO_ENABLE()
        __HAL_RCC_GPIOH_CLK_ENABLE()
00138 #define AUDIO_IN_INT_GPIO_PORT
        GPIOH
00139 #define AUDIO_IN_INT_GPIO_PIN
        GPIO_PIN_15
00140 #define AUDIO_IN_INT_IRQ
        EXTI15_10_IRQn
00141
00142 /* SAI DMA Stream definitions */
00143 #define AUDIO_IN_SAIx_DMAX_CLK_ENABLE()
        __HAL_RCC_DMA2_CLK_ENABLE()
00144 #define AUDIO_IN_SAIx_DMAX_STREAM
        DMA2_Stream7
00145 #define AUDIO_IN_SAIx_DMAX_CHANNEL
        DMA_CHANNEL_0
00146 #define AUDIO_IN_SAIx_DMAX_IRQ
        DMA2_Stream7_IRQn
00147 #define AUDIO_IN_SAIx_DMAX_PERIPH_DATA_SIZE
        DMA_PDATAALIGN_HALFWORD
00148 #define AUDIO_IN_SAIx_DMAX_MEM_DATA_SIZE
        DMA_MDATAALIGN_HALFWORD
```

```
00149
00150 #define AUDIO_IN_SAIx_DMAx_IRQHandler
        DMA2_Stream7_IRQHandler
00151 #define AUDIO_IN_INT_IRQHandler
        EXTI15_10_IRQHandler
00152
00153 /* Select the interrupt preemption priority
and subpriority for the IT/DMA interrupt */
00154 #define AUDIO_IN_IRQ_PREPRIO
        ((uint32_t)0x0F) /* Select the preemption p
riority level(0 is the highest) */
00155
00156 /*-----
-----*/
00157             CONFIGURATION: Audio Driver Con
figuration parameters
00158 -----
-----*/
00159
00160 #define AUDIODATA_SIZE
        ((uint16_t)2) /* 16-bits audio data size */
00161
00162 /* Audio status definition */
00163 #define AUDIO_OK
        ((uint8_t)0)
00164 #define AUDIO_ERROR
        ((uint8_t)1)
00165 #define AUDIO_TIMEOUT
        ((uint8_t)2)
00166
00167 /* AudioFreq * DataSize (2 bytes) * NumChann
els (Stereo: 2) */
00168 #define DEFAULT_AUDIO_IN_FREQ
I2S_AUDIOFREQ_16K
00169 #define DEFAULT_AUDIO_IN_BIT_RESOLUTION
        ((uint8_t)16)
00170 #define DEFAULT_AUDIO_IN_CHANNEL_NBR
```

```
((uint8_t)2) /* Mono = 1, Stereo = 2 */
00171 #define DEFAULT_AUDIO_IN_VOLUME
((uint16_t)64)
00172
00173 /*-----
-----*
00174                               OPTIONAL Configuration d
efines parameters
00175 -----
-----*/
00176
00177 /* Delay for the Codec to be correctly reset
 */
00178 #define CODEC_RESET_DELAY
((uint8_t)5)
00179
00180
00181 /*-----
-----*
00182                               OUTPUT DEVICES d
efinition
00183 -----
-----*/
00184 /* Alias on existing output devices to adapt
   for 2 headphones output */
00185 #define OUTPUT_DEVICE_HEADPHONE1 OUTPUT_DEVI
CE_HEADPHONE
00186 #define OUTPUT_DEVICE_HEADPHONE2 OUTPUT_DEVI
CE_SPEAKER /* Headphone2 is connected to Speaker o
utput of the wm8994 */
00187
00188 /**
00189  * @}
00190 */
00191
00192 /** @defgroup STM32746G_DISCOVERY_AUDIO_Expo
rted_Variables STM32746G_DISCOVERY_AUDIO Exported
```

```
Variables
00193     * @@
00194     */
00195 extern __IO uint16_t AudioInVolume;
00196 /**
00197     * @@
00198     */
00199
00200 /** @defgroup STM32746G_DISCOVERY_AUDIO_Exported_Macros STM32746G_DISCOVERY_AUDIO Exported Macros
00201     * @@
00202     */
00203 #define DMA_MAX(x) ((x) <= DMA_MAX_SZE)
X_SZE)? (x):DMA_MAX_SZE)
00204 /**
00205     * @@
00206     */
00207
00208 /** @addtogroup STM32746G_DISCOVERY_AUDIO_OUTPUT_Exported_Functions
00209     * @@
00210     */
00211 uint8_t BSP_AUDIO_OUT_Init(uint16_t OutputDevice, uint8_t Volume, uint32_t AudioFreq);
00212 uint8_t BSP_AUDIO_OUT_Play(uint16_t* pBuffer, uint32_t Size);
00213 void    BSP_AUDIO_OUT_ChangeBuffer(uint16_t *pData, uint16_t Size);
00214 uint8_t BSP_AUDIO_OUT_Pause(void);
00215 uint8_t BSP_AUDIO_OUT_Resume(void);
00216 uint8_t BSP_AUDIO_OUT_Stop(uint32_t Option);
00217 uint8_t BSP_AUDIO_OUT_SetVolume(uint8_t Volume);
00218 void    BSP_AUDIO_OUT_SetFrequency(uint32_t AudioFreq);
00219 void    BSP_AUDIO_OUT_SetAudioFrameSlot(uint
```

```
00213     uint32_t AudioFrameSlot);
00220     uint8_t    BSP_AUDIO_OUT_SetMute(uint32_t Cmd);
00221     uint8_t    BSP_AUDIO_OUT_SetOutputMode(uint8_t
00222     Output);
00223     void      BSP_AUDIO_OUT_DeInit(void);
00224
00224 /* User Callbacks: user has to implement the
00225   se functions in his code if they are needed. */
00225 /* This function is called when the requested
00226   data has been completely transferred.*/
00227     void      BSP_AUDIO_OUT_TransferComplete_CallB
00228 ack(void);
00229
00229 /* This function is called when half of the
00230   requested buffer has been transferred. */
00231     void      BSP_AUDIO_OUT_HalfTransfer_CallBack(
00232 void);
00233
00233 /* This function is called when an Interrupt
00234   due to transfer error on or peripheral
00235   error occurs. */
00236     void      BSP_AUDIO_OUT_Error_CallBack(void);
00237
00237 /* These function can be modified in case th
00238   e current settings (e.g. DMA stream)
00239   need to be changed for specific applicati
00240   on needs */
00241     void    BSP_AUDIO_OUT_ClockConfig(SAI_HandleTypeDef
00242 *hsai, uint32_t AudioFreq, void *Params);
00243     void    BSP_AUDIO_OUT_MspInit(SAI_HandleTypeDefDe
00244 f *hsai, void *Params);
00245     void    BSP_AUDIO_OUT_MspDeInit(SAI_HandleTypeDef
00246 Def *hsai, void *Params);
00247
00247 /**
00248  * @}
00249 */
```

```
00244
00245 /** @defgroup STM32746G_DISCOVERY_AUDIO_IN_Ex
00246   ported_Functions STM32746G_DISCOVERY_AUDIO_IN_Exp
00247   orted Functions
00248   * @{
00249   */
00248 uint8_t BSP_AUDIO_IN_Init(uint32_t AudioFreq
, uint32_t BitRes, uint32_t ChnlNbr);
00249 uint8_t BSP_AUDIO_IN_InitEx(uint16_t InputDe
vice, uint32_t AudioFreq, uint32_t BitRes, uint32_
t ChnlNbr);
00250 uint8_t BSP_AUDIO_IN_OUT_Init(uint16_t Input
Device, uint16_t OutputDevice, uint32_t AudioFreq,
uint32_t BitRes, uint32_t ChnlNbr);
00251 uint8_t BSP_AUDIO_IN_Record(uint16_t *pData,
uint32_t Size);
00252 uint8_t BSP_AUDIO_IN_Stop(uint32_t Option);
00253 uint8_t BSP_AUDIO_IN_Pause(void);
00254 uint8_t BSP_AUDIO_IN_Resume(void);
00255 uint8_t BSP_AUDIO_IN_SetVolume(uint8_t Volum
e);
00256 void     BSP_AUDIO_IN_DeInit(void);
00257 /* User Callbacks: user has to implement the
00258   se functions in his code if they are needed. */
00258 /* This function should be implemented by th
e user application.
00259   It is called into this driver when the cu
rrent buffer is filled to prepare the next
00260   buffer pointer and its size. */
00261 void     BSP_AUDIO_IN_TransferComplete_CallBa
ck(void);
00262 void     BSP_AUDIO_IN_HalfTransfer_CallBack(v
oid);
00263
00264 /* This function is called when an Interrupt
due to transfer error on or peripheral
00265   error occurs. */
```

```
00266 void      BSP_AUDIO_IN_Error_CallBack(void);
00267
00268 /* These function can be modified in case th
e current settings (e.g. DMA stream)
00269     need to be changed for specific applicati
on needs */
00270 void  BSP_AUDIO_IN_MspInit(SAI_HandleTypeDefDef
 *hsai, void *Params);
00271 void  BSP_AUDIO_IN_MspDeInit(SAI_HandleTypeDefDef
 *hsai, void *Params);
00272
00273 /**
00274  * @}
00275  */
00276
00277 /**
00278  * @}
00279  */
00280
00281 /**
00282  * @}
00283  */
00284
00285 /**
00286  * @}
00287  */
00288
00289 #ifdef __cplusplus
00290 }
00291 #endif
00292
00293 #endif /* __STM32746G_DISCOVERY_AUDIO_H */
00294
00295 ***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/
```

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_audio.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_audio.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file provides the Audio driver for the STM32746G-Discovery board.
00008 @verbatim
00009     How To use this driver:
00010     -----
00011         + This driver supports STM32F7xx devices on STM32746G-Discovery (MB1191) board.
00012         + Call the function BSP_AUDIO_OUT_Init()
00013             Outp
00014                 utDevice: physical output mode (OUTPUT_DEVICE_SPEAKER,
00015
00016                         OUTPUT_DEVICE_HEADPHONE or OUTPUT_DEVICE
00017                         _BOTH)
```



```
00029      + Call the function BSP_DISCOVERY_AUD
00030      IO_OUT_Play(
00030                           pBuffe
00031                           r: pointer to the audio data file address
00031                           Size
00031                           : size of the buffer to be sent in Bytes
00032                           )
00033                           to start playing (for the first ti
00033                           me) from the audio file/stream.
00034      + Call the function BSP_AUDIO_OUT_Pau
00034      se() to pause playing
00035      + Call the function BSP_AUDIO_OUT_Res
00035      ume() to resume playing.
00036      Note. After calling BSP_AUDIO_OUT
00036      _Pause() function for pause, only BSP_AUDIO_OUT_Re
00036      sume() should be called
00037      for resume (it is not allowed
00037      to call BSP_AUDIO_OUT_Play() in this case).
00038      Note. This function should be cal
00038      led only when the audio file is played or paused (
00038      not stopped).
00039      + For each mode, you may need to imple
00039      ment the relative callback functions into your co
00039      de.
00040      The Callback functions are named A
00040      UDIO_OUT_XXX_CallBack() and only their prototypes
00040      are declared in
00041      the stm32746g_discovery_audio.h fi
00041      le. (refer to the example for more details on the
00041      callbacks implementations)
00042      + To Stop playing, to modify the volu
00042      me level, the frequency, the audio frame slot,
00043      the device output mode the mute or
00043      the stop, use the functions: BSP_AUDIO_OUT_SetVol
00043      ume(),
00044      AUDIO_OUT_SetFrequency(), BSP_AUDI
00044      O_OUT_SetAudioFrameSlot(), BSP_AUDIO_OUT_SetOutput
```

```
Mode(),
00045           BSP_AUDIO_OUT_SetMute() and BSP_AU
DIO_OUT_Stop().
00046           + The driver API and the callback fun
ctions are at the end of the stm32746g_discovery_a
udio.h file.
00047
00048       Driver architecture:
00049       -----
00050           + This driver provides the High Audio
Layer: consists of the function API exported in t
he stm32746g_discovery_audio.h file
00051           (BSP_AUDIO_OUT_Init(), BSP_AUDIO_OU
T_Play() ... )
00052           + This driver provide also the Media
Access Layer (MAL): which consists of functions al
lowing to access the media containing/
00053           providing the audio file/stream. Th
ese functions are also included as local functions
into
00054           the stm32746g_discovery_audio_codec
.c file (SAIx_Out_Init() and SAIx_Out_DeInit(), SA
Ix_In_Init() and SAIx_In_DeInit())
00055
00056       Known Limitations:
00057       -----
00058           1- If the TDM Format used to play in
parallel 2 audio Stream (the first Stream is config
ured in codec SLOT0 and second
00059           Stream in SLOT1) the Pause/Resume,
volume and mute feature will control the both str
eams.
00060           2- Parsing of audio file is not imple
mented (in order to determine audio file propertie
s: Mono/Stereo, Data size,
00061           File size, Audio Frequency, Audio
Data header size ...). The configuration is fixed
```

for the given audio file.

00062 3- Supports only Stereo audio streaming.

00063 4- Supports only 16-bits audio data size.

00064 @endverbatim

00065 *****

00066 * @attention

00067 *

00068 * <h2><center>© COPYRIGHT(c) 2016 STMicroelectronics</center></h2>

00069 *

00070 * Redistribution and use in source and binary forms, with or without modification,

00071 * are permitted provided that the following conditions are met:

00072 * 1. Redistributions of source code must retain the above copyright notice,

00073 * this list of conditions and the following disclaimer.

00074 * 2. Redistributions in binary form must reproduce the above copyright notice,

00075 * this list of conditions and the following disclaimer in the documentation

00076 * and/or other materials provided with the distribution.

00077 * 3. Neither the name of STMicroelectronics nor the names of its contributors

00078 * may be used to endorse or promote products derived from this software

00079 * without specific prior written permission.

00080 *

00081 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

00082 * AND ANY EXPRESS OR IMPLIED WARRANTIES, I

NCLUDING, BUT NOT LIMITED TO, THE
00083 * IMPLIED WARRANTIES OF MERCHANTABILITY AN
D FITNESS FOR A PARTICULAR PURPOSE ARE
00084 * DISCLAIMED. IN NO EVENT SHALL THE COPYRI
GHT HOLDER OR CONTRIBUTORS BE LIABLE
00085 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP
ECIAL, EXEMPLARY, OR CONSEQUENTIAL
00086 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR
00087 * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER
00088 * CAUSED AND ON ANY THEORY OF LIABILITY, W
HETHER IN CONTRACT, STRICT LIABILITY,
00089 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI
SE) ARISING IN ANY WAY OUT OF THE USE
00090 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

00091 *

00092 ****

00093 */

00094

00095 /* Includes -----
----- */

00096 #include "stm32746g_discovery_audio.h"
00097

00098 /** @addtogroup BSP
00099 * @{
00100 */

00101

00102 /** @addtogroup STM32746G_DISCOVERY
00103 * @{
00104 */

00105

00106 /** @defgroup STM32746G_DISCOVERY_AUDIO STM3
2746G_DISCOVERY AUDIO
00107 * @brief This file includes the low layer

```
driver for wm8994 Audio Codec
00108      *           available on STM32746G-Discovery
board(MB1191).
00109      * @{
00110      */
00111
00112 /** @defgroup STM32746G_DISCOVERY_AUDIO_Private_Types STM32746G_DISCOVERY AUDIO Private Types
00113      * @{
00114      */
00115 /**
00116      * @}
00117      */
00118
00119 /** @defgroup STM32746G_DISCOVERY_AUDIO_Private_Defines STM32746G_DISCOVERY AUDIO Private Defines
00120      * @{
00121      */
00122 /**
00123      * @}
00124      */
00125
00126 /** @defgroup STM32746G_DISCOVERY_AUDIO_Private_Macros STM32746G_DISCOVERY AUDIO Private Macros
00127      * @{
00128      */
00129 /**
00130      * @}
00131      */
00132
00133 /** @defgroup STM32746G_DISCOVERY_AUDIO_Private_Variables STM32746G_DISCOVERY AUDIO Private Variables
00134      * @{
00135      */
```

```
00136 AUDIO_DrvTypeDef          *audio_drv;
00137 SAI_HandleTypeDef        haudio_out_sai={0}
00138 ;
00139 SAI_HandleTypeDef        haudio_in_sai={0};
00140 TIM_HandleTypeDef         haudio_tim;
00141
00142 uint16_t __IO AudioInVolume = DEFAULT_AUDIO_
IN_VOLUME;
00143 /**
00144  * @}
00145 */
00146
00147 /** @defgroup STM32746G_DISCOVERY_AUDIO_Priv
ate_Function_Protoypes STM32746G_DISCOVERY AUDIO
Private Function Prototypes
00148  * @{
00149  */
00150 static void SAIx_Out_Init(uint32_t AudioFreq
);
00151 static void SAIx_Out_DeInit(void);
00152 static void SAIx_In_Init(uint32_t SaiOutMode
, uint32_t SlotActive, uint32_t AudioFreq);
00153 static void SAIx_In_DeInit(void);
00154 /**
00155  * @}
00156 */
00157
00158 /** @defgroup STM32746G_DISCOVERY_AUDIO_OUT_
Exported_Functions STM32746G_DISCOVERY AUDIO Out E
xported Functions
00159  * @{
00160  */
00161
00162 /**
00163  * @brief Configures the audio peripherals.
```

```
00164     * @param OutputDevice: OUTPUT_DEVICE_SPEAKER, OUTPUT_DEVICE_HEADPHONE,  
00165     *                                     or OUTPUT_DEVICE_BOTH.  
00166     * @param Volume: Initial volume level (from 0 (Mute) to 100 (Max))  
00167     * @param AudioFreq: Audio frequency used to play the audio stream.  
00168     * @note The I2S PLL input clock must be done in the user application.  
00169     * @retval AUDIO_OK if correct communication, else wrong communication  
00170     */  
00171 uint8_t BSP_AUDIO_OUT_Init(uint16_t OutputDevice, uint8_t Volume, uint32_t AudioFreq)  
00172 {  
00173     uint8_t ret = AUDIO_ERROR;  
00174     uint32_t deviceid = 0x00;  
00175  
00176     /* Disable SAI */  
00177     SAIX_Out_DeInit();  
00178  
00179     /* PLL clock is set depending on the Audio Freq (44.1khz vs 48khz groups) */  
00180     BSP_AUDIO_OUT_ClockConfig(&haudio_out_sai, AudioFreq, NULL);  
00181  
00182     /* SAI data transfer preparation:  
00183     Prepare the Media to be used for the audio transfer from memory to SAI peripheral */  
00184     haudio_out_sai.Instance = AUDIO_OUT_SAIx;  
00185     if(HAL_SAI_GetState(&haudio_out_sai) == HAL_SAI_STATE_RESET)  
00186     {  
00187         /* Init the SAI MSP: this __weak function can be redefined by the application*/  
00188         BSP_AUDIO_OUT_MspInit(&haudio_out_sai, N
```

```
ULL);
00189 }
00190 SAIx_Out_Init(AudioFreq);
00191
00192 /* wm8994 codec initialization */
00193 deviceid = wm8994_drv.ReadID(AUDIO_I2C_ADD
RESS);
00194
00195 if((deviceid) == WM8994_ID)
00196 {
00197     /* Reset the Codec Registers */
00198     wm8994_drv.Reset(AUDIO_I2C_ADDRESS);
00199     /* Initialize the audio driver structure
*/
00200     audio_drv = &wm8994_drv;
00201     ret = AUDIO_OK;
00202 }
00203 else
00204 {
00205     ret = AUDIO_ERROR;
00206 }
00207
00208 if(ret == AUDIO_OK)
00209 {
00210     /* Initialize the codec internal registe
rs */
00211     audio_drv->Init(AUDIO_I2C_ADDRESS, Outpu
tDevice, Volume, AudioFreq);
00212 }
00213
00214 return ret;
00215 }
00216
00217 /**
00218 * @brief Starts playing audio stream from
a data buffer for a determined size.
00219 * @param pBuffer: Pointer to the buffer
```

```
00220     * @param  Size: Number of audio data in BY
TES unit.
00221     *           In memory, first element is for
left channel, second element is for right channel
00222     * @retval AUDIO_OK if correct communicatio
n, else wrong communication
00223     */
00224 uint8_t BSP_AUDIO_OUT_Play(uint16_t* pBuffer
, uint32_t Size)
00225 {
00226     /* Call the audio Codec Play function */
00227     if(audio_drv->Play(AUDIO_I2C_ADDRESS, pBuffer, Size) != 0)
00228     {
00229         return AUDIO_ERROR;
00230     }
00231     else
00232     {
00233         /* Update the Media layer and enable it
for play */
00234         HAL_SAI_Transmit_DMA(&haudio_out_sai, (u
int8_t*) pBuffer, DMA_MAX(Size / AUDIODATA_SIZE));
00235
00236         return AUDIO_OK;
00237     }
00238 }
00239
00240 /**
00241     * @brief Sends n-Bytes on the SAI interfa
ce.
00242     * @param pData: pointer on data address
00243     * @param Size: number of data to be writt
en
00244     * @retval None
00245     */
00246 void BSP_AUDIO_OUT_ChangeBuffer(uint16_t *pData, uint16_t Size)
```

```
00247 {
00248     HAL_SAI_Transmit_DMA(&haudio_out_sai, (ui
nt8_t*) pData, Size);
00249 }
00250
00251 /**
00252 * @brief This function Pauses the audio f
ile stream. In case
00253 *          of using DMA, the DMA Pause feat
ure is used.
00254 * @note When calling BSP_AUDIO_OUT_Pause()
function for pause, only
00255 *          BSP_AUDIO_OUT_Resume() function
should be called for resume (use of BSP_AUDIO_OUT
_Play())
00256 *          function for resume could lead
to unexpected behaviour).
00257 * @retval AUDIO_OK if correct communicatio
n, else wrong communication
00258 */
00259 uint8_t BSP_AUDIO_OUT_Pause(void)
00260 {
00261     /* Call the Audio Codec Pause/Resume funct
ion */
00262     if(audio_drv->Pause(AUDIO_I2C_ADDRESS) !=
0)
00263     {
00264         return AUDIO_ERROR;
00265     }
00266     else
00267     {
00268         /* Call the Media layer pause function */

00269         HAL_SAI_DMAPause(&haudio_out_sai);
00270
00271         /* Return AUDIO_OK when all operations a
re correctly done */
```

```
00272     return AUDIO_OK;
00273 }
00274 }
00275
00276 /**
00277 * @brief This function Resumes the audio
00278 * file stream.
00279 * @note When calling BSP_AUDIO_OUT_Pause()
00280 *       function for pause, only
00281 *       BSP_AUDIO_OUT_Resume() function
00282 *       should be called for resume (use of BSP_AUDIO_OUT
00283 *       _Play())
00284 *       function for resume could lead
00285 *       to unexpected behaviour).
00286 * @retval AUDIO_OK if correct communication,
00287 *       else wrong communication
00288 */
00289 uint8_t BSP_AUDIO_OUT_Resume(void)
00290 {
00291     /* Call the Audio Codec Pause/Resume function */
00292     if(audio_drv->Resume(AUDIO_I2C_ADDRESS) != 0)
00293     {
00294         return AUDIO_ERROR;
00295     }
00296     else
00297     {
00298         /* Call the Media layer pause/resume function */
00299         HAL_SAI_DMAResume(&haudio_out_sai);
00300
00301         /* Return AUDIO_OK when all operations are correctly done */
00302         return AUDIO_OK;
00303     }
00304 }
```

```
00299
00300 /**
00301   * @brief Stops audio playing and Power down the Audio Codec.
00302   * @param Option: could be one of the following parameters
00303   *           - CODEC_PDWN_SW: for software power off (by writing registers).
00304   *           Then no need to reconfigure the Codec after power on.
00305   *           - CODEC_PDWN_HW: completely shut down the codec (physically).
00306   *           Then need to reconfigure the Codec after power on.
00307   * @retval AUDIO_OK if correct communication, else wrong communication
00308 */
00309 uint8_t BSP_AUDIO_OUT_Stop(uint32_t Option)
00310 {
00311   /* Call the Media layer stop function */
00312   HAL_SAI_DMAShort(&haudio_out_sai);
00313
00314   /* Call Audio Codec Stop function */
00315   if(audio_drv->Stop(AUDIO_I2C_ADDRESS, Option) != 0)
00316   {
00317     return AUDIO_ERROR;
00318   }
00319   else
00320   {
00321     if(Option == CODEC_PDWN_HW)
00322     {
00323       /* Wait at least 100us */
00324       HAL_Delay(1);
00325     }
00326     /* Return AUDIO_OK when all operations are correctly done */

```

```
00327     return AUDIO_OK;
00328 }
00329 }
00330
00331 /**
00332  * @brief Controls the current audio volume level.
00333  * @param Volume: Volume level to be set in percentage from 0% to 100% (0 for
00334  * Mute and 100 for Max volume level).
00335  * @retval AUDIO_OK if correct communication, else wrong communication
00336 */
00337 uint8_t BSP_AUDIO_OUT_SetVolume(uint8_t Volume)
00338 {
00339     /* Call the codec volume control function with converted volume value */
00340     if(audio_drv->SetVolume(AUDIO_I2C_ADDRESS,
00341     Volume) != 0)
00341     {
00342         return AUDIO_ERROR;
00343     }
00344     else
00345     {
00346         /* Return AUDIO_OK when all operations are correctly done */
00347         return AUDIO_OK;
00348     }
00349 }
00350
00351 /**
00352  * @brief Enables or disables the MUTE mode by software
00353  * @param Cmd: Could be AUDIO_MUTE_ON to mute sound or AUDIO_MUTE_OFF to
```

```
00354     *          unmute the codec and restore pre  
00355     * @retval AUDIO_OK if correct communicatio  
n, else wrong communication  
00356     */  
00357 uint8_t BSP_AUDIO_OUT_SetMute(uint32_t Cmd)  
00358 {  
00359     /* Call the Codec Mute function */  
00360     if(audio_drv->SetMute(AUDIO_I2C_ADDRESS, C  
md) != 0)  
00361     {  
00362         return AUDIO_ERROR;  
00363     }  
00364     else  
00365     {  
00366         /* Return AUDIO_OK when all operations a  
re correctly done */  
00367         return AUDIO_OK;  
00368     }  
00369 }  
00370 /**  
00371     * @brief Switch dynamically (while audio  
file is played) the output target  
00372     *          (speaker or headphone).  
00373     * @param Output: The audio output target:  
00374     *          OUTPUT_DEVICE_SPEAKER,  
00375     *          OUTPUT_DEVICE_HEADPHONE or OUTPU  
T_DEVICE_BOTH  
00376     * @retval AUDIO_OK if correct communicatio  
n, else wrong communication  
00377     */  
00378 uint8_t BSP_AUDIO_OUT_SetOutputMode(uint8_t  
Output)  
00379 {  
00380     /* Call the Codec output device function */
```

```
00381     if(audio_drv->SetOutputMode(AUDIO_I2C_ADDR
00382         ESS, Output) != 0)
00383     {
00384         return AUDIO_ERROR;
00385     }
00386     {
00387         /* Return AUDIO_OK when all operations a
00388         re correctly done */
00389         return AUDIO_OK;
00390     }
00391 }
00392 /**
00393     * @brief Updates the audio frequency.
00394     * @param AudioFreq: Audio frequency used
00395     * to play the audio stream.
00396     * @note This API should be called after
00397     * the BSP_AUDIO_OUT_Init() to adjust the
00398     * audio frequency.
00399     * @retval None
00400 */
00401 void BSP_AUDIO_OUT_SetFrequency(uint32_t AudioFreq)
00402 {
00403     /* PLL clock is set depending by the Audio
00404     Freq (44.1khz vs 48khz groups) */
00405     BSP_AUDIO_OUT_ClockConfig(&haudio_out_sai,
00406         AudioFreq, NULL);
00407
00408     /* Disable SAI peripheral to allow access
00409     to SAI internal registers */
00410     __HAL_SAI_DISABLE(&haudio_out_sai);
00411
00412     /* Update the SAI audio frequency configur
00413     ation */
00414     haudio_out_sai.Init.AudioFrequency = Audio
```

```
Freq;
00409     HAL_SAI_Init(&haudio_out_sai);
00410
00411     /* Enable SAI peripheral to generate MCLK
 */
00412     __HAL_SAI_ENABLE(&haudio_out_sai);
00413 }
00414
00415 /**
00416     * @brief Updates the Audio frame slot configuration.
00417     * @param AudioFrameSlot: specifies the audio Frame slot
00418     *           This parameter can be one of the
00419     *           following values
00420     *           @arg CODEC_AUDIOFRAME_SLOT_01
00421     *           @arg CODEC_AUDIOFRAME_SLOT_02
00422     *           @arg CODEC_AUDIOFRAME_SLOT_13
00423     *           @note This API should be called after
00424     *           the BSP_AUDIO_OUT_Init() to adjust the
00425     *           audio frame slot.
00426     *           @retval None
00427 */
00428 void BSP_AUDIO_OUT_SetAudioFrameSlot(uint32_t AudioFrameSlot)
00429 {
00430     /* Disable SAI peripheral to allow access
00431     * to SAI internal registers */
00432     __HAL_SAI_DISABLE(&haudio_out_sai);
00433
00434     /* Update the SAI audio frame slot configuration */
00435     haudio_out_sai.SlotInit.SlotActive = Audio
00436     FrameSlot;
00437     HAL_SAI_Init(&haudio_out_sai);
00438 }
```

```
00435     /* Enable SAI peripheral to generate MCLK
00436     */
00437     __HAL_SAI_ENABLE(&haudio_out_sai);
00438 }
00439 /**
00440     * @brief Deinit the audio peripherals.
00441     * @retval None
00442     */
00443 void BSP_AUDIO_OUT_DeInit(void)
00444 {
00445     SAIx_Out_DeInit();
00446     /* DeInit the SAI MSP : this __weak function can be rewritten by the application */
00447     BSP_AUDIO_OUT_MspDeInit(&haudio_out_sai, N
00448     );
00449
00450 /**
00451     * @brief Tx Transfer completed callbacks.
00452     * @param hsai: SAI handle
00453     * @retval None
00454     */
00455 void HAL_SAI_TxCpltCallback(SAI_HandleTypeDef*hsai)
00456 {
00457     /* Manage the remaining file size and new
00458 address offset: This function
00459     should be coded by user (its prototype
00460     is already declared in stm32746g_discovery_audio.h
00461     */
00462     BSP_AUDIO_OUT_TransferComplete_CallBack();
00463 }
00464 /**
00465     * @brief Tx Half Transfer completed callbacks.
```

```
00464     * @param hsai: SAI handle
00465     * @retval None
00466     */
00467 void HAL_SAI_TxHalfCpltCallback(SAI_HandleTypeDef *hsai)
00468 {
00469     /* Manage the remaining file size and new
00470      address offset: This function
00471      should be coded by user (its prototype
00472      is already declared in stm32746g_discovery_audio.h
00473      ) */
00474     BSP_AUDIO_OUT_HalfTransfer_CallBack();
00475 }
00476 /**
00477     * @brief SAI error callbacks.
00478     * @param hsai: SAI handle
00479     * @retval None
00480     */
00481 void HAL_SAI_ErrorCallback(SAI_HandleTypeDef *hsai)
00482 {
00483     HAL_SAI_StateTypeDef audio_out_state;
00484     HAL_SAI_StateTypeDef audio_in_state;
00485
00486     audio_out_state = HAL_SAI_GetState(&haudio_out_sai);
00487     audio_in_state = HAL_SAI_GetState(&haudio_in_sai);
00488
00489     /* Determines if it is an audio out or aud
00490      io in error */
00491     if ((audio_out_state == HAL_SAI_STATE_BUSY)
00492         || (audio_out_state == HAL_SAI_STATE_BUSY_TX))
00493     {
00494         BSP_AUDIO_OUT_Error_CallBack();
00495     }
```

```
00492
00493     if ((audio_in_state == HAL_SAI_STATE_BUSY)
00494         ||
00495             (audio_in_state == HAL_SAI_STATE_BUSY_RX))
00496     {
00497         BSP_AUDIO_IN_Error_CallBack();
00498     }
00499 /**
00500     * @brief Manages the DMA full Transfer complete event.
00501     * @retval None
00502     */
00503 __weak void BSP_AUDIO_OUT_TransferComplete_C
allBack(void)
00504 {
00505 }
00506
00507 /**
00508     * @brief Manages the DMA Half Transfer complete event.
00509     * @retval None
00510     */
00511 __weak void BSP_AUDIO_OUT_HalfTransfer_CallB
ack(void)
00512 {
00513 }
00514
00515 /**
00516     * @brief Manages the DMA FIFO error event.
00517     * @retval None
00518     */
00519 __weak void BSP_AUDIO_OUT_Error_CallBack(void
)
00520 {
00521 }
```

```
00522
00523 /**
00524     * @brief Initializes BSP_AUDIO_OUT MSP.
00525     * @param hsai: SAI handle
00526     * @param Params
00527     * @retval None
00528 */
00529 __weak void BSP_AUDIO_OUT_MspInit(SAI_HandleTypeDef *hsai, void *Params)
00530 {
00531     static DMA_HandleTypeDef hdma_sai_tx;
00532     GPIO_InitTypeDef gpio_init_structure;
00533
00534     /* Enable SAI clock */
00535     AUDIO_OUT_SAIx_CLK_ENABLE();
00536
00537     /* Enable GPIO clock */
00538     AUDIO_OUT_SAIx_MCLK_ENABLE();
00539     AUDIO_OUT_SAIx_SCK_SD_ENABLE();
00540     AUDIO_OUT_SAIx_FS_ENABLE();
00541     /* CODEC_SAI pins configuration: FS, SCK,
MCK and SD pins -----*/
00542     gpio_init_structure.Pin = AUDIO_OUT_SAIx_FS_PIN;
00543     gpio_init_structure.Mode = GPIO_MODE_AF_PP;
00544     gpio_init_structure.Pull = GPIO_NOPULL;
00545     gpio_init_structure.Speed = GPIO_SPEED_HIGH;
00546     gpio_init_structure.Alternate = AUDIO_OUT_SAIx_FS_SD_MCLK_AF;
00547     HAL_GPIO_Init(AUDIO_OUT_SAIx_FS_GPIO_PORT,
&gpio_init_structure);
00548
00549     gpio_init_structure.Pin = AUDIO_OUT_SAIx_SCK_PIN;
00550     gpio_init_structure.Mode = GPIO_MODE_AF_PP
```

```
;  
00551     gpio_init_structure.Pull = GPIO_NOPULL;  
00552     gpio_init_structure.Speed = GPIO_SPEED_HIGH;  
00553     gpio_init_structure.Alternate = AUDIO_OUT_SAIx_SCK_AF;  
00554     HAL_GPIO_Init(AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT, &gpio_init_structure);  
00555  
00556     gpio_init_structure.Pin = AUDIO_OUT_SAIx_SD_PIN;  
00557     gpio_init_structure.Mode = GPIO_MODE_AF_PP;  
;  
00558     gpio_init_structure.Pull = GPIO_NOPULL;  
00559     gpio_init_structure.Speed = GPIO_SPEED_HIGH;  
00560     gpio_init_structure.Alternate = AUDIO_OUT_SAIx_FS_SD_MCLK_AF;  
00561     HAL_GPIO_Init(AUDIO_OUT_SAIx_SCK_SD_GPIO_PORT, &gpio_init_structure);  
00562  
00563     gpio_init_structure.Pin = AUDIO_OUT_SAIx_MCLK_PIN;  
00564     gpio_init_structure.Mode = GPIO_MODE_AF_PP;  
;  
00565     gpio_init_structure.Pull = GPIO_NOPULL;  
00566     gpio_init_structure.Speed = GPIO_SPEED_HIGH;  
00567     gpio_init_structure.Alternate = AUDIO_OUT_SAIx_FS_SD_MCLK_AF;  
00568     HAL_GPIO_Init(AUDIO_OUT_SAIx_MCLK_GPIO_PORT, &gpio_init_structure);  
00569  
00570     /* Enable the DMA clock */  
00571     AUDIO_OUT_SAIx_DMAX_CLK_ENABLE();  
00572  
00573     if(hsai->Instance == AUDIO_OUT_SAIx)
```

```

00574     {
00575         /* Configure the hdma_saiTx handle parameters */
00576         hdma_sai_tx.Init.Channel = A
00577             UDIO_OUT_SAIx_DMAX_CHANNEL;
00578         hdma_sai_tx.Init.Direction = D
00579             MA_MEMORY_TO_PERIPH;
00580         hdma_sai_tx.Init.PeriphInc = D
00581             MA_PINC_DISABLE;
00582         hdma_sai_tx.Init.MemInc = D
00583             MA_MINC_ENABLE;
00584         hdma_sai_tx.Init.PeriphDataAlignment = A
00585             UDIO_OUT_SAIx_DMAX_PERIPH_DATA_SIZE;
00586         hdma_sai_tx.Init.MemDataAlignment = A
00587             UDIO_OUT_SAIx_DMAX_MEM_DATA_SIZE;
00588         hdma_sai_tx.Init.Mode = D
00589             MA_CIRCULAR;
00590         hdma_sai_tx.Init.Priority = D
00591             MA_PRIORITY_HIGH;
00592         hdma_sai_tx.Init.FIFOMode = D
00593             MA_FIFOMODE_ENABLE;
00594         hdma_sai_tx.Init.FIFOThreshold = D
00595             MA_FIFO_THRESHOLD_FULL;
00596         hdma_sai_tx.Init.MemBurst = D
00597             MA_MBURST_SINGLE;
00598         hdma_sai_tx.InitPeriphBurst = D
00599             MA_PBURST_SINGLE;
00600
00601         hdma_sai_tx.Instance = AUDIO_OUT_SAIx_DM
00602             Ax_STREAM;
00603
00604         /* Associate the DMA handle */
00605         __HAL_LINKDMA(hsai, hdmatx, hdma_sai_tx)
00606     ;
00607
00608         /* Deinitialize the Stream for new trans
00609         fer */

```

```
00595     HAL_DMA_DeInit(&hdma_sai_tx);
00596
00597     /* Configure the DMA Stream */
00598     HAL_DMA_Init(&hdma_sai_tx);
00599 }
00600
00601     /* SAI DMA IRQ Channel configuration */
00602     HAL_NVIC_SetPriority(AUDIO_OUT_SAIx_DMAX_I
RQ, AUDIO_OUT_IRQ_PREPRIO, 0);
00603     HAL_NVIC_EnableIRQ(AUDIO_OUT_SAIx_DMAX_IRQ
);
00604 }
00605
00606 /**
00607 * @brief Deinitializes SAI MSP.
00608 * @param hsai: SAI handle
00609 * @param Params
00610 * @retval None
00611 */
00612 __weak void BSP_AUDIO_OUT_MspDeInit(SAI_Hand
leTypeDef *hsai, void *Params)
00613 {
00614     GPIO_InitTypeDef gpio_init_structure;
00615
00616     /* SAI DMA IRQ Channel deactivation */
00617     HAL_NVIC_DisableIRQ(AUDIO_OUT_SAIx_DMAX_
IRQ);
00618
00619     if(hsai->Instance == AUDIO_OUT_SAIx)
00620     {
00621         /* Deinitialize the DMA stream */
00622         HAL_DMA_DeInit(hsai->hdmatx);
00623     }
00624
00625     /* Disable SAI peripheral */
00626     __HAL_SAI_DISABLE(hsai);
00627
```

```
00628     /* Deactives CODEC_SAI pins FS, SCK, MCK  
00629     and SD by putting them in input mode */  
00629     gpio_init_structure.Pin = AUDIO_OUT_SAIx  
_FS_PIN;  
00630     HAL_GPIO_DeInit(AUDIO_OUT_SAIx_FS_GPIO_P  
ORT, gpio_init_structure.Pin);  
00631  
00632     gpio_init_structure.Pin = AUDIO_OUT_SAIx  
_SCK_PIN;  
00633     HAL_GPIO_DeInit(AUDIO_OUT_SAIx_SCK_SD_GP  
IO_PORT, gpio_init_structure.Pin);  
00634  
00635     gpio_init_structure.Pin = AUDIO_OUT_SAI  
x_SD_PIN;  
00636     HAL_GPIO_DeInit(AUDIO_OUT_SAIx_SCK_SD_GP  
IO_PORT, gpio_init_structure.Pin);  
00637  
00638     gpio_init_structure.Pin = AUDIO_OUT_SAIx  
_MCLK_PIN;  
00639     HAL_GPIO_DeInit(AUDIO_OUT_SAIx_MCLK_GPIO  
_PORT, gpio_init_structure.Pin);  
00640  
00641     /* Disable SAI clock */  
00642     AUDIO_OUT_SAIx_CLK_DISABLE();  
00643  
00644     /* GPIO pins clock and DMA clock can be  
shut down in the application  
00645         by surcharging this __weak function */  
  
00646 }  
00647  
00648 /**  
00649     * @brief Clock Config.  
00650     * @param hsai: might be required to set a  
udio peripheral predivider if any.  
00651     * @param AudioFreq: Audio frequency used  
to play the audio stream.
```

```

00652     * @param  Params
00653     * @note   This API is called by BSP_AUDIO_
00654     *           OUT_Init() and BSP_AUDIO_OUT_SetFrequency()
00655     *           Being __weak it can be overwritt
00656     * en by the application
00657     * @retval None
00658 */
00659 __weak void BSP_AUDIO_OUT_ClockConfig(SAI_Ha
ndleTypeDef *hsai, uint32_t AudioFreq, void *Param
s)
00660 {
00661     RCC_PeriphCLKInitTypeDef rcc_ex_clk_init_s
truct;
00662
00663     /* Set the PLL configuration according to
00664     * the audio frequency */
00665     if((AudioFreq == AUDIO_FREQUENCY_11K) || (
00666         AudioFreq == AUDIO_FREQUENCY_22K) || (AudioFreq ==
00667         AUDIO_FREQUENCY_44K))
00668     {
00669         /* Configure PLLI2S prescalers */
00670         /* PLLI2S_VCO: VCO_429M
00671         I2S_CLK(first level) = PLLI2S_VCO/PLLI2S
00672         Q = 429/2 = 214.5 Mhz
00673         I2S_CLK_x = I2S_CLK(first level)/PLLI2SD
00674         IVQ = 214.5/19 = 11.289 Mhz */
00675         rcc_ex_clk_init_struct.PeriphClockSelect
00676         ion = RCC_PERIPHCLK_SAI2;
00677         rcc_ex_clk_init_struct.Sai2ClockSelectio
00678         n = RCC_SAI2CLKSOURCE_PLLI2S;
00679         rcc_ex_clk_init_struct.PLLI2S.PLLI2SN =
00680         429;
00681         rcc_ex_clk_init_struct.PLLI2S.PLLI2SQ =
00682         2;

```



```
00699 /**
00700  * @brief Initializes the output Audio Codec audio interface (SAI).
00701  * @param AudioFreq: Audio frequency to be configured for the SAI peripheral.
00702  * @note The default SlotActive configuration is set to CODEC_AUDIOFRAME_SLOT_0123
00703  *       and user can update this configuration using
00704  * @retval None
00705 */
00706 static void SAIx_Out_Init(uint32_t AudioFreq)
00707 {
00708     /* Initialize the haudio_out_sai Instance parameter */
00709     haudio_out_sai.Instance = AUDIO_OUT_SAIx;
00710
00711     /* Disable SAI peripheral to allow access to SAI internal registers */
00712     __HAL_SAI_DISABLE(&haudio_out_sai);
00713
00714     /* Configure SAI_Block_x
00715      LSBFirst: Disabled
00716      DataSize: 16 */
00717     haudio_out_sai.Init.AudioFrequency = AudioFreq;
00718     haudio_out_sai.Init.AudioMode = SAI_MODEMASTER_TX;
00719     haudio_out_sai.Init.NoDivider = SAI_MASTER_DIVIDER_ENABLED;
00720     haudio_out_sai.Init.Protocol = SAI_FREE_PROTOCOL;
00721     haudio_out_sai.Init.DataSize = SAI_DATASIZE_16;
00722     haudio_out_sai.Init.FirstBit = SAI_FIRSTBIT_MSB;
```

```
00723     haudio_out_sai.Init.ClockStrobing = SAI_CLOCKSTROBING_RISINGEDGE;
00724     haudio_out_sai.Init.Synchro = SAI_SYNCHRONOUS;
00725     haudio_out_sai.Init.OutputDrive = SAI_OUTPUTDRIVE_ENABLED;
00726     haudio_out_sai.Init.FIFOThreshold = SAI_FIFO_THRESHOLD_1QF;
00727
00728     /* Configure SAI_Block_x Frame
00729      Frame Length: 64
00730      Frame active Length: 32
00731      FS Definition: Start frame + Channel Side
identification
00732      FS Polarity: FS active Low
00733      FS Offset: FS asserted one bit before the
first bit of slot 0 */
00734     haudio_out_sai.FrameInit.FrameLength = 64;

00735     haudio_out_sai.FrameInit.ActiveFrameLength
= 32;
00736     haudio_out_sai.FrameInit.FSDefinition = SAI_FS_CHANNEL_IDENTIFICATION;
00737     haudio_out_sai.FrameInit.FSPolarity = SAI_FS_ACTIVE_LOW;
00738     haudio_out_sai.FrameInit.FSOffset = SAI_FS_BEFOREFIRSTBIT;
00739
00740     /* Configure SAI Block_x Slot
00741      Slot First Bit Offset: 0
00742      Slot Size : 16
00743      Slot Number: 4
00744      Slot Active: All slot actives */
00745     haudio_out_sai.SlotInit.FirstBitOffset = 0
;
00746     haudio_out_sai.SlotInit.SlotSize = SAI_SLOT_SIZE_DATASIZE;
```

```
00747     haudio_out_sai.SlotInit.SlotNumber = 4;
00748     haudio_out_sai.SlotInit.SlotActive = CODEC
00749         _AUDIOFRAME_SLOT_0123;
00749
00750     HAL_SAI_Init(&haudio_out_sai);
00751
00752     /* Enable SAI peripheral to generate MCLK
00752 */
00753     __HAL_SAI_ENABLE(&haudio_out_sai);
00754 }
00755
00756
00757
00758 /**
00759     * @brief Deinitializes the output Audio C
00759     * odec audio interface (SAI).
00760     * @retval None
00761 */
00762 static void SAIx_Out_DeInit(void)
00763 {
00764     /* Initialize the haudio_out_sai Instance
00764     * parameter */
00765     haudio_out_sai.Instance = AUDIO_OUT_SAIx;
00766
00767     /* Disable SAI peripheral */
00768     __HAL_SAI_DISABLE(&haudio_out_sai);
00769
00770     HAL_SAI_DeInit(&haudio_out_sai);
00771 }
00772
00773 /**
00774     * @}
00775 */
00776
00777 /** @defgroup STM32746G_DISCOVERY_AUDIO_Out_
00778     Private_Functions STM32746G_DISCOVERY_AUDIO_Out_Pr
00779     iate_Functions
```

```
00778 * @{
00779 */
00780
00781 /**
00782 * @brief Initializes wave recording.
00783 * @param AudioFreq: Audio frequency to be
00784 configured for the SAI peripheral.
00785 * @param ChnlNbr: Channel number.
00786 * @retval AUDIO_OK if correct communication
00787 else wrong communication
00788 uint8_t BSP_AUDIO_IN_Init(uint32_t AudioFreq,
00789 , uint32_t BitRes, uint32_t ChnlNbr)
00790 return BSP_AUDIO_IN_InitEx(INPUT_DEVICE_DIGITAL_MICROPHONE_2, AudioFreq, BitRes, ChnlNbr);
00791 }
00792
00793 /**
00794 * @brief Initializes wave recording.
00795 * @param InputDevice: INPUT_DEVICE_DIGITAL_MICROPHONE_2 or INPUT_DEVICE_INPUT_LINE_1
00796 * @param AudioFreq: Audio frequency to be
00797 configured for the SAI peripheral.
00798 * @param ChnlNbr: Channel number.
00799 * @retval AUDIO_OK if correct communication
00800 else wrong communication
00801 uint8_t BSP_AUDIO_IN_InitEx(uint16_t InputDevice,
00802 uint32_t AudioFreq, uint32_t BitRes, uint32_
00803 t ChnlNbr)
00804 {
00805     uint8_t ret = AUDIO_ERROR;
```

```
00804     uint32_t deviceid = 0x00;
00805     uint32_t slot_active;
00806
00807     if ((InputDevice != INPUT_DEVICE_INPUT_LINE_1) && /* Only INPUT_LINE_1 and MICROPHONE_2 inputs supported */
00808         (InputDevice != INPUT_DEVICE_DIGITAL_MICROPHONE_2))
00809     {
00810         ret = AUDIO_ERROR;
00811     }
00812     else
00813     {
00814         /* Disable SAI */
00815         SAIx_In_DeInit();
00816
00817         /* PLL clock is set depending on the AudioFreq (44.1khz vs 48khz groups) */
00818         BSP_AUDIO_OUT_ClockConfig(&haudio_in_sai,
00819             AudioFreq, NULL); /* Clock config is shared between AUDIO IN and OUT */
00820
00821         /* SAI data transfer preparation:
00822             Prepare the Media to be used for the audio transfer from SAI peripheral to memory */
00823         haudio_in_sai.Instance = AUDIO_IN_SAIx;
00824         if(HAL_SAI_GetState(&haudio_in_sai) == HAL_SAI_STATE_RESET)
00825         {
00826             /* Init the SAI MSP: this __weak function can be redefined by the application*/
00827             BSP_AUDIO_OUT_MspInit(&haudio_in_sai,
00828                 NULL); /* Initialize GPIOs for SAI2 block A Master signals */
00829             BSP_AUDIO_IN_MspInit(&haudio_in_sai, NULL);
00830         }
```

```
00829
00830     /* Configure SAI in master RX mode :
00831         * - SAI2_block_A in master RX mode
00832         * - SAI2_block_B in slave RX mode synchronous from SAI2_block_A
00833         */
00834     if (InputDevice == INPUT_DEVICE_DIGITAL_
MICROPHONE_2)
00835     {
00836         slot_active = CODEC_AUDIOFRAME_SLOT_13
;
00837     }
00838 else
00839 {
00840     slot_active = CODEC_AUDIOFRAME_SLOT_02
;
00841 }
00842     SAIx_In_Init(SAI_MODEMASTER_RX, slot_act
ive, AudioFreq);
00843
00844     /* wm8994 codec initialization */
00845     deviceid = wm8994_drv.ReadID(AUDIO_I2C_A
DDRESS);
00846
00847     if((deviceid) == WM8994_ID)
00848     {
00849         /* Reset the Codec Registers */
00850         wm8994_drv.Reset(AUDIO_I2C_ADDRESS);
00851         /* Initialize the audio driver structu
re */
00852         audio_drv = &wm8994_drv;
00853         ret = AUDIO_OK;
00854     }
00855 else
00856 {
00857     ret = AUDIO_ERROR;
00858 }
```

```
00859
00860     if(ret == AUDIO_OK)
00861     {
00862         /* Initialize the codec internal regis
00863         ters */
00864         audio_drv->Init(AUDIO_I2C_ADDRESS, Inp
00865         utDevice, 100, AudioFreq);
00866     }
00867 }
00868
00869 /**
00870 * @brief Initializes wave recording and p
00871 * layback in parallel.
00872 * @param InputDevice: INPUT_DEVICE_DIGITA
00873 L_MICROPHONE_2
00874 * @param OutputDevice: OUTPUT_DEVICE_SPEA
00875 KER, OUTPUT_DEVICE_HEADPHONE,
00876 *                                     or OUTPUT_DEVICE_B
00877 OTH.
00878 * @param AudioFreq: Audio frequency to be
00879 configured for the SAI peripheral.
00880 * @param BitRes: Audio frequency to be co
00881 nfigured.
00882 * @param ChnlNbr: Channel number.
00883 * @retval AUDIO_OK if correct communicatio
00884 n, else wrong communication
00885 */
00879 uint8_t BSP_AUDIO_IN_OUT_Init(uint16_t Input
00880 Device, uint16_t OutputDevice, uint32_t AudioFreq,
00881 uint32_t BitRes, uint32_t ChnlNbr)
00882 {
00883     uint8_t ret = AUDIO_ERROR;
00884     uint32_t deviceid = 0x00;
00885     uint32_t slot_active;
```

```
00885     if (InputDevice != INPUT_DEVICE_DIGITAL_MICROPHONE_2) /* Only MICROPHONE_2 input supported */
00886     {
00887         ret = AUDIO_ERROR;
00888     }
00889     else
00890     {
00891         /* Disable SAI */
00892         SAIx_In_DeInit();
00893         SAIx_Out_DeInit();
00894
00895         /* PLL clock is set depending on the AudioFreq (44.1khz vs 48khz groups) */
00896         BSP_AUDIO_OUT_ClockConfig(&haudio_in_sai,
00897             AudioFreq, NULL); /* Clock config is shared between AUDIO IN and OUT */
00897
00898         /* SAI data transfer preparation:
00899             Prepare the Media to be used for the audio transfer from SAI peripheral to memory */
00900         haudio_in_sai.Instance = AUDIO_IN_SAIx;
00901         if(HAL_SAI_GetState(&haudio_in_sai) == HAL_SAI_STATE_RESET)
00902         {
00903             /* Init the SAI MSP: this __weak function can be redefined by the application*/
00904             BSP_AUDIO_IN_MspInit(&haudio_in_sai, NULL);
00905         }
00906
00907         /* SAI data transfer preparation:
00908             Prepare the Media to be used for the audio transfer from memory to SAI peripheral */
00909         haudio_out_sai.Instance = AUDIO_OUT_SAIx
00910         ;
00910         if(HAL_SAI_GetState(&haudio_out_sai) ==
```

```
HAL_SAI_STATE_RESET)
00911      {
00912          /* Init the SAI MSP: this __weak function can be redefined by the application*/
00913          BSP_AUDIO_OUT_MspInit(&haudio_out_sai,
00914              NULL);
00915      }
00916      /* Configure SAI in master mode :
00917          * - SAI2_block_A in master TX mode
00918          * - SAI2_block_B in slave RX mode synchronous from SAI2_block_A
00919          */
00920      if (InputDevice == INPUT_DEVICE_DIGITAL_MICROPHONE_2)
00921      {
00922          slot_active = CODEC_AUDIOFRAME_SLOT_13
00923      ;
00924      }
00925      else
00926      {
00927          slot_active = CODEC_AUDIOFRAME_SLOT_02
00928      ;
00929      }
00930      SAIx_In_Init(SAI_MODEMASTER_TX, slot_active, AudioFreq);
00931      /* wm8994 codec initialization */
00932      deviceid = wm8994_drv.ReadID(AUDIO_I2C_ADDRESS);
00933      if((deviceid) == WM8994_ID)
00934      {
00935          /* Reset the Codec Registers */
00936          wm8994_drv.Reset(AUDIO_I2C_ADDRESS);
00937          /* Initialize the audio driver structure */
00938      }
```

```
00938         audio_drv = &wm8994_drv;
00939         ret = AUDIO_OK;
00940     }
00941     else
00942     {
00943         ret = AUDIO_ERROR;
00944     }
00945
00946     if(ret == AUDIO_OK)
00947     {
00948         /* Initialize the codec internal registers */
00949         audio_drv->Init(AUDIO_I2C_ADDRESS, InputDevice | OutputDevice, 100, AudioFreq);
00950     }
00951 }
00952 return ret;
00953 }
00954
00955
00956 /**
00957 * @brief Starts audio recording.
00958 * @param pbuf: Main buffer pointer for the recorded data storing
00959 * @param size: size of the recorded buffer in number of elements (typically number of half-words)
00960 * Be careful that it is not the same unit than BSP_AUDIO_OUT_Play function
00961 * @retval AUDIO_OK if correct communication, else wrong communication
00962 */
00963 uint8_t  BSP_AUDIO_IN_Record(uint16_t* pbuf,
00964                               uint32_t size)
00964 {
00965     uint32_t ret = AUDIO_ERROR;
00966
```

```
00967     /* Start the process receive DMA */
00968     HAL_SAI_Receive_DMA(&haudio_in_sai, (uint8
00969     _t*)pbuf, size);
00970     /* Return AUDIO_OK when all operations are
00971     correctly done */
00972     ret = AUDIO_OK;
00973     return ret;
00974 }
00975
00976 /**
00977     * @brief Stops audio recording.
00978     * @param Option: could be one of the foll
00979     * owing parameters
00980     *           - CODEC_PDWN_SW: for software
00981     * power off (by writing registers).
00982     *           Then no need
00983     * to reconfigure the Codec after power on.
00984     *           - CODEC_PDWN_HW: completely sh
00985     * ut down the codec (physically).
00986     *           Then need to
00987     * reconfigure the Codec after power on.
00988     * @retval AUDIO_OK if correct communicatio
00989     * n, else wrong communication
00990     */
00991     uint8_t BSP_AUDIO_IN_Stop(uint32_t Option)
00992 {
00993     /* Call the Media layer stop function */
00994     HAL_SAI_DMASStop(&haudio_in_sai);
00995
00996     /* Call Audio Codec Stop function */
00997     if(audio_drv->Stop(AUDIO_I2C_ADDRESS, Opti
00998     on) != 0)
00999     {
01000         return AUDIO_ERROR;
01001     }
```

```
00995     else
00996     {
00997         if(Option == CODEC_PDWN_HW)
00998         {
00999             /* Wait at least 100us */
01000             HAL_Delay(1);
01001         }
01002         /* Return AUDIO_OK when all operations are
01003         correctly done */
01004         return AUDIO_OK;
01005     }
01006
01007 /**
01008     * @brief Pauses the audio file stream.
01009     * @retval AUDIO_OK if correct communication,
01010     * else wrong communication
01011 */
01011 uint8_t BSP_AUDIO_IN_Pause(void)
01012 {
01013     /* Call the Media layer pause function */
01014     HAL_SAI_DMAPause(&haudio_in_sai);
01015     /* Return AUDIO_OK when all operations are
01016     correctly done */
01017     return AUDIO_OK;
01018 }
01019 /**
01020     * @brief Resumes the audio file stream.
01021     * @retval AUDIO_OK if correct communication,
01022     * else wrong communication
01023 */
01023 uint8_t BSP_AUDIO_IN_Resume(void)
01024 {
01025     /* Call the Media layer pause/resume function */
01026     HAL_SAI_DMAResume(&haudio_in_sai);
```

```
01027     /* Return AUDIO_OK when all operations are
01028    correctly done */
01029    return AUDIO_OK;
01030
01031 /**
01032   * @brief Controls the audio in volume level.
01033   * @param Volume: Volume level in range 0( Mute)..80(+0dB)..100(+17.625dB)
01034   * @retval AUDIO_OK if correct communication, else wrong communication
01035 */
01036 uint8_t BSP_AUDIO_IN_SetVolume(uint8_t Volume)
01037 {
01038     /* Call the codec volume control function
01039    with converted volume value */
01040     if(audio_drv->SetVolume(AUDIO_I2C_ADDRESS,
01041     Volume) != 0)
01042     {
01041         return AUDIO_ERROR;
01042     }
01043     else
01044     {
01045         /* Set the Global variable AudioInVolume
01046        */
01046         AudioInVolume = Volume;
01047         /* Return AUDIO_OK when all operations are
01048        correctly done */
01048         return AUDIO_OK;
01049     }
01050 }
01051
01052 /**
01053   * @brief Deinit the audio IN peripherals.
01054   * @retval None
```

```
01055      */
01056 void BSP_AUDIO_IN_DeInit(void)
01057 {
01058     SAIx_In_DeInit();
01059     /* DeInit the SAI MSP : this __weak function can be rewritten by the application */
01060     BSP_AUDIO_IN_MspDeInit(&haudio_in_sai, NULL);
01061 }
01062
01063 /**
01064     * @brief Rx Transfer completed callbacks.
01065     * @param hsai: SAI handle
01066     * @retval None
01067 */
01068 void HAL_SAI_RxCpltCallback(SAI_HandleTypeDefDef *hsai)
01069 {
01070     /* Call the record update function to get the next buffer to fill and its size (size is ignored) */
01071     BSP_AUDIO_IN_TransferComplete_CallBack();
01072 }
01073
01074 /**
01075     * @brief Rx Half Transfer completed callbacks.
01076     * @param hsai: SAI handle
01077     * @retval None
01078 */
01079 void HAL_SAI_RxHalfCpltCallback(SAI_HandleTypeDefDef *hsai)
01080 {
01081     /* Manage the remaining file size and new address offset: This function
01082         should be coded by user (its prototype is already declared in stm32746g_discovery_audio.h
```

```
) */
01083     BSP_AUDIO_IN_HalfTransfer_CallBack();
01084 }
01085
01086 /**
01087     * @brief User callback when record buffer
01088     * is filled.
01089     * @retval None
01090     */
01090 __weak void BSP_AUDIO_IN_TransferComplete_Ca
llBack(void)
01091 {
01092     /* This function should be implemented by
01093     * the user application.
01094     * It is called into this driver when the
01095     * current buffer is filled
01096     * to prepare the next buffer pointer and
01097     * its size. */
01098 /**
01099     * @brief Manages the DMA Half Transfer co
mplete event.
01100     * @retval None
01101     */
01101 __weak void BSP_AUDIO_IN_HalfTransfer_CallBa
ck(void)
01102 {
01103     /* This function should be implemented by
01104     * the user application.
01105     * It is called into this driver when the
01106     * current buffer is filled
01107     * to prepare the next buffer pointer and
01108     * its size. */
01108 */
```

```
01109     * @brief  Audio IN Error callback function.  
01110     * @retval None  
01111     */  
01112 __weak void BSP_AUDIO_IN_Error_CallBack(void  
)  
01113 {  
01114     /* This function is called when an Interru  
pt due to transfer error on or peripheral  
error occurs. */  
01116 }  
01117  
01118 /**  
01119     * @brief  Initializes BSP_AUDIO_IN MSP.  
01120     * @param  hsai: SAI handle  
01121     * @param  Params  
01122     * @retval None  
01123     */  
01124 __weak void BSP_AUDIO_IN_MspInit(SAI_HandleTypeDef  
*hsai, void *Params)  
01125 {  
01126     static DMA_HandleTypeDef hdma_sai_rx;  
01127     GPIO_InitTypeDef gpio_init_structure;  
01128  
01129     /* Enable SAI clock */  
01130     AUDIO_IN_SAIx_CLK_ENABLE();  
01131  
01132     /* Enable SD GPIO clock */  
01133     AUDIO_IN_SAIx_SD_ENABLE();  
01134     /* CODEC_SAI pin configuration: SD pin */  
01135     gpio_init_structure.Pin = AUDIO_IN_SAIx_SD  
_PIN;  
01136     gpio_init_structure.Mode = GPIO_MODE_AF_PP  
;  
01137     gpio_init_structure.Pull = GPIO_NOPULL;  
01138     gpio_init_structure.Speed = GPIO_SPEED_FAS  
T;
```

```
01139     gpio_init_structure.Alternate = AUDIO_IN_S  
AIx_SD_AF;  
01140     HAL_GPIO_Init(AUDIO_IN_SAIx_SD_GPIO_PORT,  
&gpio_init_structure);  
01141  
01142     /* Enable Audio INT GPIO clock */  
01143     AUDIO_IN_INT_GPIO_ENABLE();  
01144     /* Audio INT pin configuration: input */  
01145     gpio_init_structure.Pin = AUDIO_IN_INT_GPI  
O_PIN;  
01146     gpio_init_structure.Mode = GPIO_MODE_INPUT  
;  
01147     gpio_init_structure.Pull = GPIO_NOPULL;  
01148     gpio_init_structure.Speed = GPIO_SPEED_FAS  
T;  
01149     HAL_GPIO_Init(AUDIO_IN_INT_GPIO_PORT, &gpi  
o_init_structure);  
01150  
01151     /* Enable the DMA clock */  
01152     AUDIO_IN_SAIx_DMAx_CLK_ENABLE();  
01153  
01154     if(hsai->Instance == AUDIO_IN_SAIx)  
01155     {  
01156         /* Configure the hdma_sai_rx handle para  
meters */  
01157         hdma_sai_rx.Init.Channel = A  
UDIO_IN_SAIx_DMAx_CHANNEL;  
01158         hdma_sai_rx.Init.Direction = D  
MA_PERIPH_TO_MEMORY;  
01159         hdma_sai_rx.InitPeriphInc = D  
MA_PINC_DISABLE;  
01160         hdma_sai_rx.Init.MemInc = D  
MA_MINC_ENABLE;  
01161         hdma_sai_rx.Init.PeriphDataAlignment = A  
UDIO_IN_SAIx_DMAx_PERIPH_DATA_SIZE;  
01162         hdma_sai_rx.Init.MemDataAlignment = A  
UDIO_IN_SAIx_DMAx_MEM_DATA_SIZE;
```

```
01163     hdma_sai_rx.Init.Mode          = D
MA_CIRCULAR;
01164     hdma_sai_rx.Init.Priority      = D
MA_PRIORITY_HIGH;
01165     hdma_sai_rx.Init.FIFOMode     = D
MA_FIFOMODE_DISABLE;
01166     hdma_sai_rx.Init.FIFOThreshold = D
MA_FIFO_THRESHOLD_FULL;
01167     hdma_sai_rx.Init.MemBurst      = D
MA_MBURST_SINGLE;
01168     hdma_sai_rx.Init.PeriphBurst   = D
MA_MBURST_SINGLE;
01169
01170     hdma_sai_rx.Instance = AUDIO_IN_SAIx_DMA
x_STREAM;
01171
01172     /* Associate the DMA handle */
01173     __HAL_LINKDMA(hsai, hdmarx, hdma_sai_rx)
01174 ;
01175     /* Deinitialize the Stream for new trans
fer */
01176     HAL_DMA_DeInit(&hdma_sai_rx);
01177
01178     /* Configure the DMA Stream */
01179     HAL_DMA_Init(&hdma_sai_rx);
01180 }
01181
01182     /* SAI DMA IRQ Channel configuration */
01183     HAL_NVIC_SetPriority(AUDIO_IN_SAIx_DMAX_IRQ
, AUDIO_IN_IRQ_PREPRIO, 0);
01184     HAL_NVIC_EnableIRQ(AUDIO_IN_SAIx_DMAX_IRQ)
01185 ;
01186     /* Audio INT IRQ Channel configuration */
01187     HAL_NVIC_SetPriority(AUDIO_IN_INT_IRQ, AUD
IO_IN_IRQ_PREPRIO, 0);
```

```
01188     HAL_NVIC_EnableIRQ(AUDIO_IN_INT IRQ);
01189 }
01190
01191 /**
01192 * @brief DeInitializes BSP_AUDIO_IN MSP.
01193 * @param hsai: SAI handle
01194 * @param Params
01195 * @retval None
01196 */
01197 __weak void BSP_AUDIO_IN_MspDeInit(SAI_HandleTypeDefDef *hsai, void *Params)
01198 {
01199     GPIO_InitTypeDef gpio_init_structure;
01200
01201     static DMA_HandleTypeDefDef hdma_sai_rx;
01202
01203     /* SAI IN DMA IRQ Channel deactivation */
01204     HAL_NVIC_DisableIRQ(AUDIO_IN_SAIx_DMAx_IRQ
01205 );
01206     if(hsai->Instance == AUDIO_IN_SAIx)
01207     {
01208         /* Deinitialize the Stream for new trans
fer */
01209         HAL_DMA_DeInit(&hdma_sai_rx);
01210     }
01211
01212     /* Disable SAI block */
01213     __HAL_SAI_DISABLE(hsai);
01214
01215     /* Disable pin: SD pin */
01216     gpio_init_structure.Pin = AUDIO_IN_SAIx_SD
01217 _PIN;
01218     HAL_GPIO_DeInit(AUDIO_IN_SAIx_SD_GPIO_PORT
01219 , gpio_init_structure.Pin);
01220
01221     /* Disable SAI clock */
```

```
01220     AUDIO_IN_SAIx_CLK_DISABLE();
01221
01222     /* GPIO pins clock and DMA clock can be sh
ut down in the application
01223         by surcharging this __weak function */
01224 }
01225
01226
01227 /*****
01228 **** Static Functions
01229 ****
01230
01231 /**
01232     * @brief Initializes the input Audio Code
c audio interface (SAI).
01233     * @param SaiOutMode: SAI_MODEMASTER_TX (f
or record and playback in parallel)
01234     *                               or SAI_MODEMASTER_RX
(for record only).
01235     * @param SlotActive: CODEC_AUDIOFRAME_SLOT
T_02 or CODEC_AUDIOFRAME_SLOT_13
01236     * @param AudioFreq: Audio frequency to be
configured for the SAI peripheral.
01237     * @retval None
01238 */
01239 static void SAIx_In_Init(uint32_t SaiOutMode
, uint32_t SlotActive, uint32_t AudioFreq)
01240 {
01241     /* Initialize SAI2 block A in MASTER RX */
01242     /* Initialize the haudio_out_sai Instance
parameter */
01243     haudio_out_sai.Instance = AUDIO_OUT_SAIx;
01244
01245     /* Disable SAI peripheral to allow access
to SAI internal registers */
```

```
01246     __HAL_SAI_DISABLE(&haudio_out_sai);
01247
01248     /* Configure SAI_Block_x
01249     LSBFirst: Disabled
01250     DataSize: 16 */
01251     haudio_out_sai.Init.AudioFrequency = Audio
01252     Freq;
01253     haudio_out_sai.Init.AudioMode = SaiOutMode
01254     ;
01255     haudio_out_sai.Init.NoDivider = SAI_MASTER
01256     DIVIDER_ENABLED;
01257     haudio_out_sai.Init.Protocol = SAI_FREE_PR
01258     OTOCOL;
01259     haudio_out_sai.Init.DataSize = SAI_DATASIZ
01260     E_16;
01261     haudio_out_sai.Init.FirstBit = SAI_FIRSTBI
01262     T_MSB;
01263     haudio_out_sai.Init.ClockStrobing = SAI_CL
01264     OCKSTROBING_RISINGEDGE;
01265     haudio_out_sai.Init.Synchro = SAI_SYNCHRO
01266     NOUS;
01267     haudio_out_sai.Init.OutputDrive = SAI_OUTP
01268     UTDRIVE_ENABLED;
01269     haudio_out_sai.Init.FIFOThreshold = SAI_FI
01270     FO_THRESHOLD_1QF;
01271
01272     /* Configure SAI_Block_x Frame
01273     Frame Length: 64
01274     Frame active Length: 32
01275     FS Definition: Start frame + Channel Side
01276     identification
01277     FS Polarity: FS active Low
01278     FS Offset: FS asserted one bit before the
01279     first bit of slot 0 */
01280     haudio_out_sai.FrameInit.FrameLength = 64;
01281     haudio_out_sai.FrameInit.ActiveFrameLength
01282     = 32;
```

```
01270     haudio_out_sai.FrameInit.FSDefinition = SA
I_FS_CHANNEL_IDENTIFICATION;
01271     haudio_out_sai.FrameInit.FSPolarity = SAI_
FS_ACTIVE_LOW;
01272     haudio_out_sai.FrameInit.FSOffset = SAI_FS
_BEFOREFIRSTBIT;
01273
01274     /* Configure SAI Block_x Slot
01275      Slot First Bit Offset: 0
01276      Slot Size : 16
01277      Slot Number: 4
01278      Slot Active: All slot actives */
01279     haudio_out_sai.SlotInit.FirstBitOffset = 0
;
01280     haudio_out_sai.SlotInit.SlotSize = SAI_SLO
TSIZE_DATASIZE;
01281     haudio_out_sai.SlotInit.SlotNumber = 4;
01282     haudio_out_sai.SlotInit.SlotActive = SlotA
ctive;
01283
01284     HAL_SAI_Init(&haudio_out_sai);
01285
01286     /* Initialize SAI2 block B in SLAVE RX syn
chronous from SAI2 block A */
01287     /* Initialize the haudio_in_sai Instance p
arameter */
01288     haudio_in_sai.Instance = AUDIO_IN_SAIx;
01289
01290     /* Disable SAI peripheral to allow access
to SAI internal registers */
01291     __HAL_SAI_DISABLE(&haudio_in_sai);
01292
01293     /* Configure SAI_Block_x
01294      LSBFirst: Disabled
01295      DataSize: 16 */
01296     haudio_in_sai.Init.AudioFrequency = AudioF
req;
```

```
01297     haudio_in_sai.Init.AudioMode = SAI_MODESLA  
VE_RX;  
01298     haudio_in_sai.Init.NoDivider = SAI_MASTERD  
IVIDER_ENABLED;  
01299     haudio_in_sai.Init.Protocol = SAI_FREE_ PRO  
TOCOL;  
01300     haudio_in_sai.Init.DataSize = SAI_DATASIZE  
_16;  
01301     haudio_in_sai.Init.FirstBit = SAI_FIRSTBIT  
_MSB;  
01302     haudio_in_sai.Init.ClockStrobing = SAI_CLO  
CKSTROBING_RISINGEDGE;  
01303     haudio_in_sai.Init.Synchro = SAI_SYNCHRONO  
US;  
01304     haudio_in_sai.Init.OutputDrive = SAI_OUTPU  
TDRIVE_DISABLED;  
01305     haudio_in_sai.Init.FIFOThreshold = SAI_FIF  
OTHRESHOLD_1QF;  
01306  
01307     /* Configure SAI_Block_x Frame  
01308     Frame Length: 64  
01309     Frame active Length: 32  
01310     FS Definition: Start frame + Channel Side  
identification  
01311     FS Polarity: FS active Low  
01312     FS Offset: FS asserted one bit before the  
first bit of slot 0 */  
01313     haudio_in_sai.FrameInit.FrameLength = 64;  
01314     haudio_in_sai.FrameInit.ActiveFrameLength  
= 32;  
01315     haudio_in_sai.FrameInit.FSDefinition = SAI  
_FS_CHANNEL_IDENTIFICATION;  
01316     haudio_in_sai.FrameInit.FSPolarity = SAI_F  
S_ACTIVE_LOW;  
01317     haudio_in_sai.FrameInit.FSOffset = SAI_FS_  
BEFOREFIRSTBIT;  
01318
```

```
01319 /* Configure SAI Block_x Slot
01320 Slot First Bit Offset: 0
01321 Slot Size : 16
01322 Slot Number: 4
01323 Slot Active: All slot active */
01324 haudio_in_sai.SlotInit.FirstBitOffset = 0;
01325 haudio_in_sai.SlotInit.SlotSize = SAI_SLOT
SIZE_DATASIZE;
01326 haudio_in_sai.SlotInit.SlotNumber = 4;
01327 haudio_in_sai.SlotInit.SlotActive = SlotAc
tive;
01328
01329 HAL_SAI_Init(&haudio_in_sai);
01330
01331 /* Enable SAI peripheral to generate MCLK
*/
01332 __HAL_SAI_ENABLE(&haudio_out_sai);
01333
01334 /* Enable SAI peripheral */
01335 __HAL_SAI_ENABLE(&haudio_in_sai);
01336 }
01337
01338
01339
01340 /**
01341 * @brief Deinitializes the output Audio C
01342 odec audio interface (SAI).
01343 * @retval None
01344 */
01345 static void SAIx_In_DeInit(void)
01346 {
01347 /* Initialize the haudio_in_sai Instance p
arameter */
01348 haudio_in_sai.Instance = AUDIO_IN_SAIx;
01349
01350 /* Disable SAI peripheral */
01351 __HAL_SAI_DISABLE(&haudio_in_sai);
```

```
01351
01352     HAL_SAI_DeInit(&haudio_in_sai);
01353 }
01354
01355
01356 /**
01357 * @}
01358 */
01359 /**
01360 * @}
01361 */
01362 */
01363
01364 /**
01365 * @}
01366 */
01367
01368 /**
01369 * @}
01370 */
01371
01372 /***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/
```

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_camera.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_camera.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains the common d
efines and functions prototypes for
00008 *           the stm32746g_discovery_camera.
c driver.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and bin
ary forms, with or without modification,
00015 * are permitted provided that the followin
g conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038      */
00039 /* Define to prevent recursive inclusion ---
00040 -----
00041 #ifndef __STM32746G_DISCOVERY_CAMERA_H
00042 #define __STM32746G_DISCOVERY_CAMERA_H
00043
00044 #ifdef __cplusplus
00045     extern "C" {
00046
00047 /* Includes -----
00048 -----
00049 /* Include Camera component Driver */
00050 #include "../Components/ov9655/ov9655.h"
00051 #include "stm32746g_discovery.h"
00052 /**
00053 * @{
00054 */
00055
00056 /**
00057 * @{
00058 */
00059
00060 /**
00061 * @{
00062 */
00063
00064 /**
00065 * @{
00066 */
```

```
00067
00068 /**
00069   * @brief Camera State structures definition
00070 */
00071 typedef enum
00072 {
00073   CAMERA_OK          = 0x00,
00074   CAMERA_ERROR       = 0x01,
00075   CAMERA_TIMEOUT     = 0x02,
00076   CAMERA_NOT_DETECTED = 0x03,
00077   CAMERA_NOT_SUPPORTED = 0x04
00078
00079 } Camera_StatusTypeDef;
00080
00081 #define RESOLUTION_R160x120           CAMERA_R160
00082             /* QQVGA Resolution */           /
00083 #define RESOLUTION_R320x240           CAMERA_R320
00084             /* QVGA Resolution */           /
00085 #define RESOLUTION_R480x272           CAMERA_R480
00086             /* 480x272 Resolution */          /
00087 #define RESOLUTION_R640x480           CAMERA_R640
00088             /* VGA Resolution */           /
00089 /**
00090   * @}
00091 */
00092
00093 /** @defgroup STM32746G_DISCOVERY_CAMERA_Exported_Constants STM32746G_DISCOVERY_CAMERA Exported Constants
00094   * @{
00095 */
00096 #define BSP_CAMERA_IRQHandler      DCMI IRQH
00097 #define BSP_CAMERA_DMA_IRQHandler DMA2 Stream1 IRQHandler
00098
00099
```

```
00095 /**
00096     * @}
00097     */
00098
00099 /** @addtogroup STM32746G_DISCOVERY_CAMERA_Exported_Functions
00100     * @{
00101     */
00102 uint8_t BSP_CAMERA_Init(uint32_t Resolution);
00103
00104 void    BSP_CAMERA_ContinuousStart(uint8_t *buff);
00105 void    BSP_CAMERA_SnapshotStart(uint8_t *buff);
00106 void    BSP_CAMERA_Suspend(void);
00107 void    BSP_CAMERA_Resume(void);
00108 uint8_t BSP_CAMERA_Stop(void);
00109 void    BSP_CAMERA_PwrUp(void);
00110 void    BSP_CAMERA_PwrDown(void);
00111 void    BSP_CAMERA_LineEventCallback(void);
00112 void    BSP_CAMERA_VsyncEventCallback(void);
00113 void    BSP_CAMERA_FrameEventCallback(void);
00114 void    BSP_CAMERA_ErrorCallback(void);
00115
00116 /* Camera features functions prototype */
00117 void    BSP_CAMERA_ContrastBrightnessConfig(
00118     uint32_t contrast_level, uint32_t brightness_level
00119 );
00120
00121 void    BSP_CAMERA_BlackwhiteConfig(uint32_t Mode);
00122 void    BSP_CAMERA_ColorEffectConfig(uint32_t Effect);
00123
00124 /* These functions can be modified in case the current settings (e.g. DMA stream)
00125 need to be changed for specific application
```

```

on needs */
00123 void BSP_CAMERA_MspInit(DCMI_HandleTypeDefDef * 
hdcmi, void *Params);
00124 void BSP_CAMERA_MspDeInit(DCMI_HandleTypeDefDef 
*hdcmi, void *Params);
00125
00126 /**
00127  * @}
00128 */
00129 */
00130
00131 /**
00132  * @}
00133 */
00134
00135 /**
00136  * @}
00137 */
00138
00139 /**
00140  * @}
00141 */
00142
00143 #ifdef __cplusplus
00144 }
00145 #endif
00146
00147 #endif /* __STM32746G_DISCOVERY_CAMERA_H */
00148
00149 /***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/

```

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_camera.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_camera.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file includes the driver for Camera modules mounted on
00008 *           STM32746G-Discovery board.
00009 @verbatim
00010     How to use this driver:
00011 -----
00012     - This driver is used to drive the camera.
00013     - The OV9655 component driver MUST be included with this driver.
00014
00015     Driver description:
00016 -----
00017     + Initialization steps:
00018         o Initialize the camera using the BSP
```

```
_CAMERA_Init() function.  
00019          o Start the camera capture/snapshot u  
sing the CAMERA_Start() function.  
00020          o Suspend, resume or stop the camera  
capture using the following functions:  
00021          - BSP_CAMERA_Suspend()  
00022          - BSP_CAMERA_Resume()  
00023          - BSP_CAMERA_Stop()  
00024  
00025      + Options  
00026          o Increase or decrease on the fly the  
brightness and/or contrast  
00027          using the following function:  
00028          - BSP_CAMERA_ContrastBrightnessConf  
ig  
00029          o Add a special effect on the fly usi  
ng the following functions:  
00030          - BSP_CAMERA_BlackWhiteConfig()  
00031          - BSP_CAMERA_ColorEffectConfig()  
00032  @endverbatim  
00033  *****  
*****  
00034  * @attention  
00035  *  
00036  * <h2><center>&copy; COPYRIGHT(c) 2016 STM  
icroelectronics</center></h2>  
00037  *  
00038  * Redistribution and use in source and bin  
ary forms, with or without modification,  
00039  * are permitted provided that the followin  
g conditions are met:  
00040  *   1. Redistributions of source code must  
retain the above copyright notice,  
00041  *       this list of conditions and the fol  
lowing disclaimer.  
00042  *   2. Redistributions in binary form must  
reproduce the above copyright notice,
```

00043 * this list of conditions and the following disclaimer in the documentation

00044 * and/or other materials provided with the distribution.

00045 * 3. Neither the name of STMicroelectronics nor the names of its contributors

00046 * may be used to endorse or promote products derived from this software

00047 * without specific prior written permission.

00048 *

00049 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

00050 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

00051 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

00052 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

00053 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

00054 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

00055 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

00056 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

00057 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

00058 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

00059 *

00060 *****

00061 */

00062

00063 /* Includes -----

```

----- */
00064 #include "stm32746g_discovery_camera.h"
00065 #include "stm32746g_discovery.h"
00066
00067 /** @addtogroup BSP
00068 * @{
00069 */
00070
00071 /** @addtogroup STM32746G_DISCOVERY
00072 * @{
00073 */
00074
00075 /** @addtogroup STM32746G_DISCOVERY_CAMERA
00076 * @{
00077 */
00078
00079 /** @defgroup STM32746G_DISCOVERY_CAMERA_Private_TypesDefinitions STM32746G_DISCOVERY_CAMERA Private Types Definitions
00080 * @{
00081 */
00082 /**
00083 * @}
00084 */
00085
00086 /** @defgroup STM32746G_DISCOVERY_CAMERA_Private_Defines STM32746G_DISCOVERY_CAMERA Private Defines
00087 * @{
00088 */
00089 #define CAMERA_VGA_RES_X 640
00090 #define CAMERA_VGA_RES_Y 480
00091 #define CAMERA_480x272_RES_X 480
00092 #define CAMERA_480x272_RES_Y 272
00093 #define CAMERA_QVGA_RES_X 320
00094 #define CAMERA_QVGA_RES_Y 240
00095 #define CAMERA_QQVGA_RES_X 160

```

```
00096 #define CAMERA_QQVGA_RES_Y 120
00097 /**
00098 * @}
00099 */
00100
00101 /** @defgroup STM32746G_DISCOVERY_CAMERA_Private_Macros STM32746G_DISCOVERY_CAMERA Private Macros
00102 * @{
00103 */
00104 /**
00105 * @}
00106 */
00107
00108 /** @defgroup STM32746G_DISCOVERY_CAMERA_Private_Variables STM32746G_DISCOVERY_CAMERA Private Variables
00109 * @{
00110 */
00111 DCMI_HandleTypeDef hDcmiHandler;
00112 CAMERA_DrvTypeDef *camera_drv;
00113 /* Camera current resolution naming (QQVGA, VGA, ...) */
00114 static uint32_t CameraCurrentResolution;
00115
00116 /* Camera module I2C HW address */
00117 static uint32_t CameraHwAddress;
00118 /**
00119 * @}
00120 */
00121
00122 /** @defgroup STM32746G_DISCOVERY_CAMERA_Private_FunctionPrototypes STM32746G_DISCOVERY_CAMERA Private Function Prototypes
00123 * @{
00124 */
00125 static uint32_t GetSize(uint32_t resolution);
```

```
00126 /**
00127     * @}
00128     */
00129
00130 /** @defgroup STM32746G_DISCOVERY_CAMERA_Exported_Functions STM32746G_DISCOVERY_CAMERA Exported Functions
00131     * @{
00132     */
00133
00134 /**
00135     * @brief Initializes the camera.
00136     * @param Resolution : camera sensor requested resolution (x, y) : standard resolution
00137     *           naming QQVGA, QVGA, VGA ...
00138     * @retval Camera status
00139     */
00140 uint8_t BSP_CAMERA_Init(uint32_t Resolution)
00141 {
00142     DCMI_HandleTypeDef *phdcmi;
00143     uint8_t status = CAMERA_ERROR;
00144
00145     /* Get the DCMI handle structure */
00146     phdcmi = &hDcmiHandler;
00147
00148     /*** Configures the DCMI to interface with
00149      * the camera module ***/
00150     /* DCMI configuration */
00151     phdcmi->Init.CaptureRate      = DCMI_CR_AL
00152     L_FRAME;
00153     phdcmi->Init.HSPolarity      = DCMI_HSPOL
00154     ARITY_LOW;
00155     phdcmi->Init.SynchroMode    = DCMI_SYNCH
00156     RO_HARDWARE;
00157     phdcmi->Init.VSPolarity      = DCMI_VSPOL
00158     ARITY_HIGH;
```

```
00154     phdcmi->Init.ExtendedDataMode = DCMI_EXTEN  
D_DATA_8B;  
00155     phdcmi->Init.PCKPolarity      = DCMI_PCKPO  
LARITY_RISING;  
00156     phdcmi->Instance           = DCMI;  
00157  
00158     /* Power up camera */  
00159     BSP_CAMERA_PwrUp();  
00160  
00161     /* Read ID of Camera module via I2C */  
00162     if(ov9655_ReadID(CAMERA_I2C_ADDRESS) == OV  
9655_ID)  
00163     {  
00164         /* Initialize the camera driver structur  
e */  
00165         camera_drv = &ov9655_drv;  
00166         CameraHwAddress = CAMERA_I2C_ADDRESS;  
00167  
00168         /* DCMI Initialization */  
00169         BSP_CAMERA_MspInit(&hDcmiHandler, NULL);  
00170         HAL_DCMI_Init(phdcmi);  
00171  
00172         /* Camera Module Initialization via I2C  
to the wanted 'Resolution' */  
00173         if (Resolution == CAMERA_R480x272)  
00174             { /* For 480x272 resolution, the OV9  
655 sensor is set to VGA resolution  
00175                 * as OV9655 doesn't supports 480x  
272 resolution,  
00176                 * then DCMI is configured to outp  
ut a 480x272 cropped window */  
00177             camera_drv->Init(CameraHwAddress, CAME  
RA_R640x480);  
00178             HAL_DCMI_ConfigCROP(phdcmi,  
/* Crop in the middle of the VGA picture */  
00179                         (CAMERA_VGA_RES_X -  
CAMERA_480x272_RES_X)/2,
```

```
00180                                         (CAMERA_VGA_RES_Y -  
CAMERA_480x272_RES_Y)/2,  
00181                                         (CAMERA_480x272_RES  
_X * 2) - 1,  
00182                                         CAMERA_480x272_RES  
_Y - 1);  
00183         HAL_DCMI_EnableCROP(phdcmi);  
00184     }  
00185     else  
00186     {  
00187         camera_drv->Init(CameraHwAddress, Reso  
lution);  
00188         HAL_DCMI_DisableCROP(phdcmi);  
00189     }  
00190  
00191     CameraCurrentResolution = Resolution;  
00192  
00193     /* Return CAMERA_OK status */  
00194     status = CAMERA_OK;  
00195 }  
00196 else  
00197 {  
00198     /* Return CAMERA_NOT_SUPPORTED status */  
00199     status = CAMERA_NOT_SUPPORTED;  
00200 }  
00201  
00202     return status;  
00203 }  
00204  
00205 /**  
00206     * @brief DeInitializes the camera.  
00207     * @retval Camera status  
00208     */  
00209 uint8_t BSP_CAMERA_DeInit(void)  
00210 {  
00211     hDcmiHandler.Instance             = DCMI;  
00212
```

```
00213     HAL_DCMI_DeInit(&hDcmiHandler);
00214     BSP_CAMERA_MspDeInit(&hDcmiHandler, NULL);
00215     return CAMERA_OK;
00216 }
00217
00218 /**
00219  * @brief Starts the camera capture in continuous mode.
00220  * @param buff: pointer to the camera output buffer
00221  * @retval None
00222 */
00223 void BSP_CAMERA_ContinuousStart(uint8_t *buff)
00224 {
00225     /* Start the camera capture */
00226     HAL_DCMI_Start_DMA(&hDcmiHandler, DCMI_MODE_CONTINUOUS, (uint32_t)buff, GetSize(CameraCurrentResolution));
00227 }
00228
00229 /**
00230  * @brief Starts the camera capture in snapshot mode.
00231  * @param buff: pointer to the camera output buffer
00232  * @retval None
00233 */
00234 void BSP_CAMERA_SnapshotStart(uint8_t *buff)
00235 {
00236     /* Start the camera capture */
00237     HAL_DCMI_Start_DMA(&hDcmiHandler, DCMI_MODE_SNAPSHOT, (uint32_t)buff, GetSize(CameraCurrentResolution));
00238 }
00239
00240 /**
```

```
00241     * @brief Suspend the CAMERA capture
00242     * @retval None
00243     */
00244 void BSP_CAMERA_Suspend(void)
00245 {
00246     /* Suspend the Camera Capture */
00247     HAL_DCMI_Suspend(&hDcmiHandler);
00248 }
00249
00250 /**
00251     * @brief Resume the CAMERA capture
00252     * @retval None
00253     */
00254 void BSP_CAMERA_Resume(void)
00255 {
00256     /* Start the Camera Capture */
00257     HAL_DCMI_Resume(&hDcmiHandler);
00258 }
00259
00260 /**
00261     * @brief Stop the CAMERA capture
00262     * @retval Camera status
00263     */
00264 uint8_t BSP_CAMERA_Stop(void)
00265 {
00266     uint8_t status = CAMERA_ERROR;
00267
00268     if(HAL_DCMI_Stop(&hDcmiHandler) == HAL_OK)
00269     {
00270         status = CAMERA_OK;
00271     }
00272
00273     /* Set Camera in Power Down */
00274     BSP_CAMERA_PwrDown();
00275
00276     return status;
00277 }
```

```
00278
00279 /**
00280     * @brief CANERA power up
00281     * @retval None
00282 */
00283 void BSP_CAMERA_PwrUp(void)
00284 {
00285     GPIO_InitTypeDef gpio_init_structure;
00286
00287     /* Enable GPIO clock */
00288     __HAL_RCC_GPIOH_CLK_ENABLE();
00289
00290     /*** Configure the GPIO ***/
00291     /* Configure DCMI GPIO as alternate function */
00292     gpio_init_structure.Pin      = GPIO_PIN_13;
00293     gpio_init_structure.Mode    = GPIO_MODE_OUTPUT_PP;
00294     gpio_init_structure.Pull    = GPIO_NOPULL;
00295     gpio_init_structure.Speed  = GPIO_SPEED_HIGH;
00296     HAL_GPIO_Init(GPIOH, &gpio_init_structure);
00297
00298     /* De-assert the camera POWER_DOWN pin (active high) */
00299     HAL_GPIO_WritePin(GPIOH, GPIO_PIN_13, GPIO_PIN_RESET);
00300
00301     HAL_Delay(3);      /* POWER_DOWN de-asserted during 3ms */
00302 }
00303
00304 /**
00305     * @brief CANERA power down
```

```
00306     * @retval None
00307     */
00308 void BSP_CAMERA_PwrDown(void)
00309 {
00310     GPIO_InitTypeDef gpio_init_structure;
00311
00312     /* Enable GPIO clock */
00313     __HAL_RCC_GPIOH_CLK_ENABLE();
00314
00315     /*** Configure the GPIO ***/
00316     /* Configure DCMI GPIO as alternate function */
00317     gpio_init_structure.Pin      = GPIO_PIN_13;
00318     gpio_init_structure.Mode    = GPIO_MODE_OUTPUT_PP;
00319     gpio_init_structure.Pull    = GPIO_NOPULL;
00320     gpio_init_structure.Speed   = GPIO_SPEED_HIGH;
00321     HAL_GPIO_Init(GPIOH, &gpio_init_structure);
00322
00323     /* Assert the camera POWER_DOWN pin (active high) */
00324     HAL_GPIO_WritePin(GPIOH, GPIO_PIN_13, GPIO_PIN_SET);
00325 }
00326
00327 /**
00328     * @brief Configures the camera contrast and brightness.
00329     * @param contrast_level: Contrast level
00330     *           This parameter can be one of the following values:
00331     *           @arg CAMERA_CONTRAST_LEVEL4:
00332         for contrast +2
```

```
00332      * @arg  CAMERA_CONTRAST_LEVEL3:  
00333      * @arg  CAMERA_CONTRAST_LEVEL2:  
00334      * @arg  CAMERA_CONTRAST_LEVEL1:  
00335      * @arg  CAMERA_CONTRAST_LEVEL0:  
00336      * @param brightness_level: Contrast level  
00337      * This parameter can be one of th  
e following values:  
00338      * @arg  CAMERA_BRIGHTNESS_LEVEL  
4: for brightness +2  
00339      * @arg  CAMERA_BRIGHTNESS_LEVEL  
3: for brightness +1  
00340      * @arg  CAMERA_BRIGHTNESS_LEVEL  
2: for brightness  0  
00341      * @arg  CAMERA_BRIGHTNESS_LEVEL  
1: for brightness -1  
00342      * @arg  CAMERA_BRIGHTNESS_LEVEL  
0: for brightness -2  
00343      * @retval None  
00344      */  
00345 void BSP_CAMERA_ContrastBrightnessConfig(uint32_t contrast_level, uint32_t brightness_level)  
00346 {  
00347     if(camera_drv->Config != NULL)  
00348     {  
00349         camera_drv->Config(CameraHwAddress, CAME  
RA_CONTRAST_BRIGHTNESS, contrast_level, brightness  
_level);  
00350     }  
00351 }  
00352  
00353 /**  
00354     * @brief Configures the camera white bal  
ance.
```

```
00355 * @param Mode: black_white mode
00356 * This parameter can be one of th
e following values:
00357 * @arg CAMERA_BLACK_WHITE_BW
00358 * @arg CAMERA_BLACK_WHITE_NEGA
TIVE
00359 * @arg CAMERA_BLACK_WHITE_BW_N
EGATIVE
00360 * @arg CAMERA_BLACK_WHITE_NORM
AL
00361 * @retval None
00362 */
00363 void BSP_CAMERA_BlackwhiteConfig(uint32_t Mo
de)
00364 {
00365     if(camera_drv->Config != NULL)
00366     {
00367         camera_drv->Config(CameraHwAddress, CAME
RA_BLACK_WHITE, Mode, 0);
00368     }
00369 }
00370
00371 /**
00372 * @brief Configures the camera color effe
ct.
00373 * @param Effect: Color effect
00374 * This parameter can be one of th
e following values:
00375 * @arg CAMERA_COLOR_EFFECT_ANT
IQUE
00376 * @arg CAMERA_COLOR_EFFECT_BLU
E
00377 * @arg CAMERA_COLOR_EFFECT_GRE
EN
00378 * @arg CAMERA_COLOR_EFFECT_RED
00379 * @retval None
```

```
00380     */
00381 void BSP_CAMERA_ColorEffectConfig(uint32_t Effect)
00382 {
00383     if(camera_drv->Config != NULL)
00384     {
00385         camera_drv->Config(CameraHwAddress, CAMERA_COLOR_EFFECT, Effect, 0);
00386     }
00387 }
00388
00389 /**
00390 * @brief Get the capture size in pixels unit.
00391 * @param resolution: the current resolution.
00392 * @retval capture size in pixels unit.
00393 */
00394 static uint32_t GetSize(uint32_t resolution)
00395 {
00396     uint32_t size = 0;
00397
00398     /* Get capture size */
00399     switch (resolution)
00400     {
00401         case CAMERA_R160x120:
00402             {
00403                 size = 0x2580;
00404             }
00405             break;
00406         case CAMERA_R320x240:
00407             {
00408                 size = 0x9600;
00409             }
00410             break;
00411         case CAMERA_R480x272:
00412             {
```

```
00413         size = 0xFF00;
00414     }
00415     break;
00416 case CAMERA_R640x480:
00417 {
00418     size = 0x25800;
00419 }
00420 break;
00421 default:
00422 {
00423     break;
00424 }
00425 }
00426
00427 return size;
00428 }
00429
00430 /**
00431 * @brief Initializes the DCMI MSP.
00432 * @param hdcmi: HDMI handle
00433 * @param Params
00434 * @retval None
00435 */
00436 __weak void BSP_CAMERA_MspInit(DCMI_HandleTypeDefDef *hdcmi, void *Params)
00437 {
00438     static DMA_HandleTypeDefDef hdma_handler;
00439     GPIO_InitTypeDef gpio_init_structure;
00440
00441     /*** Enable peripherals and GPIO clocks ***/
00442     /* Enable DCMI clock */
00443     __HAL_RCC_DCMI_CLK_ENABLE();
00444
00445     /* Enable DMA2 clock */
00446     __HAL_RCC_DMA2_CLK_ENABLE();
00447 }
```

```
00448     /* Enable GPIO clocks */
00449     __HAL_RCC_GPIOA_CLK_ENABLE();
00450     __HAL_RCC_GPIOD_CLK_ENABLE();
00451     __HAL_RCC_GPIOE_CLK_ENABLE();
00452     __HAL_RCC_GPIOG_CLK_ENABLE();
00453     __HAL_RCC_GPIOH_CLK_ENABLE();
00454
00455     /*** Configure the GPIO ***/
00456     /* Configure DCMI GPIO as alternate function */
00457     gpio_init_structure.Pin      = GPIO_PIN_4
00458     | GPIO_PIN_6;
00459     gpio_init_structure.Mode    = GPIO_MODE_
00460     AF_PP;
00461     gpio_init_structure.Pull    = GPIO_PULLU
00462     P;
00463     gpio_init_structure.Speed   = GPIO_SPEED_
00464     _HIGH;
00465     gpio_init_structure.Alternate = GPIO_AF13_
00466     DCMI;
00467     HAL_GPIO_Init(GPIOA, &gpio_init_structure)
00468 ;
00469
00470     gpio_init_structure.Pin      = GPIO_PIN_3
00471 ;
00472     gpio_init_structure.Mode    = GPIO_MODE_
00473     AF_PP;
00474     gpio_init_structure.Pull    = GPIO_PULLU
00475     P;
00476     gpio_init_structure.Speed   = GPIO_SPEED_
00477     _HIGH;
00478     gpio_init_structure.Alternate = GPIO_AF13_
00479     DCMI;
00480     HAL_GPIO_Init(GPIOD, &gpio_init_structure)
00481 ;
00482
00483     gpio_init_structure.Pin      = GPIO_PIN_5
```

```
    | GPIO_PIN_6;
00472  gpio_init_structure.Mode      = GPIO_MODE_
AF_PP;
00473  gpio_init_structure.Pull      = GPIO_PULLU
P;
00474  gpio_init_structure.Speed     = GPIO_SPEED
_HIGH;
00475  gpio_init_structure.Alternate = GPIO_AF13_
DCMI;
00476  HAL_GPIO_Init(GPIOE, &gpio_init_structure)
;
00477
00478  gpio_init_structure.Pin        = GPIO_PIN_9
;
00479  gpio_init_structure.Mode      = GPIO_MODE_
AF_PP;
00480  gpio_init_structure.Pull      = GPIO_PULLU
P;
00481  gpio_init_structure.Speed     = GPIO_SPEED
_HIGH;
00482  gpio_init_structure.Alternate = GPIO_AF13_
DCMI;
00483  HAL_GPIO_Init(GPIOG, &gpio_init_structure)
;
00484
00485  gpio_init_structure.Pin        = GPIO_PIN_9
| GPIO_PIN_10 | GPIO_PIN_11 | \
00486                                GPIO_PIN_1
2 | GPIO_PIN_14;
00487  gpio_init_structure.Mode      = GPIO_MODE_
AF_PP;
00488  gpio_init_structure.Pull      = GPIO_PULLU
P;
00489  gpio_init_structure.Speed     = GPIO_SPEED
_HIGH;
00490  gpio_init_structure.Alternate = GPIO_AF13_
DCMI;
```

```
00491     HAL_GPIO_Init(GPIOH, &gpio_init_structure)
00492 ;
00493     /*** Configure the DMA ***/
00494     /* Set the parameters to be configured */
00495     hdma_handler.Init.Channel      = DM
A_CHANNEL_1;
00496     hdma_handler.Init.Direction    = DM
A_PERIPH_TO_MEMORY;
00497     hdma_handler.Init.PeriphInc   = DM
A_PINC_DISABLE;
00498     hdma_handler.Init.MemInc      = DM
A_MINC_ENABLE;
00499     hdma_handler.InitPeriphDataAlignment = DM
A_PDATAALIGN_WORD;
00500     hdma_handler.Init.MemDataAlignment = DM
A_MDATAALIGN_WORD;
00501     hdma_handler.Init.Mode        = DM
A_CIRCULAR;
00502     hdma_handler.Init.Priority    = DM
A_PRIORITY_HIGH;
00503     hdma_handler.Init.FIFOMode    = DM
A_FIFOMODE_DISABLE;
00504     hdma_handler.Init.FIFOThreshold = DM
A_FIFO_THRESHOLD_FULL;
00505     hdma_handler.Init.MemBurst    = DM
A_MBURST_SINGLE;
00506     hdma_handler.InitPeriphBurst  = DM
A_PBURST_SINGLE;
00507
00508     hdma_handler.Instance = DMA2_Stream1;
00509
00510     /* Associate the initialized DMA handle to
       the DCMI handle */
00511     __HAL_LINKDMA(hdcmi, DMA_Handle, hdma_handler);
00512
```

```
00513     /*** Configure the NVIC for DCMI and DMA */
00514     /* NVIC configuration for DCMI transfer complete interrupt */
00515     HAL_NVIC_SetPriority(DCMI_IRQn, 0x0F, 0);
00516     HAL_NVIC_EnableIRQ(DCMI_IRQn);
00517
00518     /* NVIC configuration for DMA2D transfer complete interrupt */
00519     HAL_NVIC_SetPriority(DMA2_Stream1_IRQn, 0x0F, 0);
00520     HAL_NVIC_EnableIRQ(DMA2_Stream1_IRQn);
00521
00522     /* Configure the DMA stream */
00523     HAL_DMA_Init(hdcmi->DMA_Handle);
00524 }
00525
00526
00527 /**
00528 * @brief DeInitializes the DCMI MSP.
00529 * @param hdcmi: HDMI handle
00530 * @param Params
00531 * @retval None
00532 */
00533 __weak void BSP_CAMERA_MspDeInit(DCMI_HandleTypeDef *(hdcmi, void *Params)
00534 {
00535     /* Disable NVIC for DCMI transfer complete interrupt */
00536     HAL_NVIC_DisableIRQ(DCMI_IRQn);
00537
00538     /* Disable NVIC for DMA2 transfer complete interrupt */
00539     HAL_NVIC_DisableIRQ(DMA2_Stream1_IRQn);
00540
00541     /* Configure the DMA stream */
00542     HAL_DMA_DeInit(hdcmi->DMA_Handle);
```

```
00543
00544     /* Disable DCMI clock */
00545     __HAL_RCC_DCMI_CLK_DISABLE();
00546
00547     /* GPIO pins clock and DMA clock can be sh
ut down in the application
00548         by surcharging this __weak function */
00549 }
00550
00551 /**
00552     * @brief Line event callback
00553     * @param hdcmi: pointer to the DCMI handl
e
00554     * @retval None
00555 */
00556 void HAL_DCMI_LineEventCallback(DCMI_HandleTypeDefDef *(hdcmi))
00557 {
00558     BSP_CAMERA_LineEventCallback();
00559 }
00560
00561 /**
00562     * @brief Line Event callback.
00563     * @retval None
00564 */
00565 __weak void BSP_CAMERA_LineEventCallback(void)
00566 {
00567     /* NOTE : This function Should not be modi
fied, when the callback is needed,
00568             the HAL_DCMI_LineEventCallback c
ould be implemented in the user file
00569 */
00570 }
00571
00572 /**
00573     * @brief VSYNC event callback
```

```
00574     * @param  hdcmi: pointer to the DCMI handle
00575     * @retval None
00576     */
00577 void HAL_DCMI_VsyncEventCallback(DCMI_HandleTypeDefDef *(hdcmi)
00578 {
00579     BSP_CAMERA_VsyncEventCallback();
00580 }
00581
00582 /**
00583     * @brief VSYNC Event callback.
00584     * @retval None
00585     */
00586 __weak void BSP_CAMERA_VsyncEventCallback(void)
00587 {
00588     /* NOTE : This function Should not be modified, when the callback is needed,
00589             the HAL_DCMI_VsyncEventCallback could be implemented in the user file
00590     */
00591 }
00592
00593 /**
00594     * @brief Frame event callback
00595     * @param  hdcmi: pointer to the DCMI handle
00596     * @retval None
00597     */
00598 void HAL_DCMI_FrameEventCallback(DCMI_HandleTypeDefDef *(hdcmi)
00599 {
00600     BSP_CAMERA_FrameEventCallback();
00601 }
00602
00603 /**
```

```
00604     * @brief  Frame Event callback.
00605     * @retval None
00606     */
00607 __weak void BSP_CAMERA_FrameEventCallback(vo
id)
00608 {
00609     /* NOTE : This function Should not be modi
fied, when the callback is needed,
00610             the HAL_DCMI_FrameEventCallback
could be implemented in the user file
00611     */
00612 }
00613
00614 /**
00615     * @brief  Error callback
00616     * @param  hdcmi: pointer to the DCMI handl
e
00617     * @retval None
00618     */
00619 void HAL_DCMI_ErrorCallback(DCMI_HandleTypeD
ef *(hdcmi)
00620 {
00621     BSP_CAMERA_ErrorCallback();
00622 }
00623
00624 /**
00625     * @brief  Error callback.
00626     * @retval None
00627     */
00628 __weak void BSP_CAMERA_ErrorCallback(void)
00629 {
00630     /* NOTE : This function Should not be modi
fied, when the callback is needed,
00631             the HAL_DCMI_ErrorCallback could
be implemented in the user file
00632     */
00633 }
```

```
00634
00635  /**
00636      * @}
00637      */
00638
00639  /**
00640      * @}
00641      */
00642
00643  /**
00644      * @}
00645      */
00646
00647  /**
00648      * @}
00649      */
00650
00651 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_eeprom.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_eeprom.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains all the func-
tions prototypes for
00008 *           the stm32746g_discovery_eeprom.
c firmware driver.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and bin-
ary forms, with or without modification,
00015 * are permitted provided that the followin-
g conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038      */
00039 /* Define to prevent recursive inclusion ---
00040 -----
00041 #ifndef __STM32746G_DISCOVERY_EEPROM_H
00042 #define __STM32746G_DISCOVERY_EEPROM_H
00043
00044 #ifdef __cplusplus
00045     extern "C" {
00046 #endif
00047
00048 /* Includes -----
00049 -----
00050 /** @addtogroup BSP
00051     * @{
00052     */
00053
00054 /** @addtogroup STM32746G_DISCOVERY
00055     * @{
00056     */
00057
00058 /** @addtogroup STM32746G_DISCOVERY_EEPROM
00059     * @brief This file includes the I2C EEPROM
00060     * driver of STM32746G-Discovery board.
00061     * @{
00062     */
00063 /** @defgroup STM32746G_DISCOVERY_EEPROM_Exported_Types STM32746G_DISCOVERY_EEPROM Exported Types
00064     * @{
00065     */
```

```
00066 /**
00067     * @}
00068     */
00069
00070 /** @defgroup STM32746G_DISCOVERY_EEPROM_Exported_Constants STM32746G_DISCOVERY_EEPROM Exported Constants
00071     * @{
00072     */
00073 /* EEPROM hardware address and page size */
00074 #define EEPROM_PAGESIZE ((uint8_t)4)
00075 #define EEPROM_MAX_SIZE ((uint16_t)0x2000) /* 64Kbit */
00076
00077
00078 /* Maximum number of trials for EEPROM_WaitEEPROMStandbyState() function */
00079 #define EEPROM_MAX_TRIALS ((uint32_t)3000)
00080
00081 #define EEPROM_OK ((uint32_t)0)
00082 #define EEPROM_FAIL ((uint32_t)1)
00083 #define EEPROM_TIMEOUT ((uint32_t)2)
00084 /**
00085     * @}
00086     */
00087
00088 /** @defgroup STM32746G_DISCOVERY_EEPROM_Exported_Macros STM32746G_DISCOVERY_EEPROM Exported Macros
00089     * @{
00090     */
00091 /**
```

```
00092     * @}
00093     */
00094
00095 /** @addtogroup STM32746G_DISCOVERY_EEPROM_E
00096 xported_Functions
00097 */
00098 uint32_t BSP_EEPROM_Init(void);
00099 uint8_t BSP_EEPROM_DeInit(void);
00100 uint32_t BSP_EEPROM_ReadBuffer(uint8_t* pBuffer, uint16_t ReadAddr, uint16_t* NumByteToRead);
00101 uint32_t BSP_EEPROM_WritePage(uint8_t* pBuffer, uint16_t WriteAddr, uint8_t* NumByteToWrite);
00102 uint32_t BSP_EEPROM_WriteBuffer(uint8_t* pBuffer, uint16_t WriteAddr, uint16_t NumByteToWrite)
00103 ;
00104 uint32_t BSP_EEPROM_WaitEepromStandbyState(v
00105 oid);
00106
00107 /* USER Callbacks: This function is declared
00108 as __weak in EEPROM driver and
00109 should be implemented into user applicati
00110 on.
00111     BSP_EEPROM_TIMEOUT_UserCallback() functio
00112 n is called whenever a timeout condition
00113 occurs during communication (waiting on a
00114 n event that doesn't occur, bus
00115 errors, busy devices ...). */
00116 void      BSP_EEPROM_TIMEOUT_UserCallback(void
00117 );
00118
00119 /* Link function for I2C EEPROM peripheral */
00120
00121 void          EEPROM_IO_Init(void);
00122 HAL_StatusTypeDef EEPROM_IO_WriteData(uint16
00123 _t DevAddress, uint16_t MemAddress, uint8_t *pBuff
00124 er, uint32_t BufferSize);
```

```
00115 HAL_StatusTypeDef EEPROM_IO_ReadData(uint16_t DevAddress, uint16_t MemAddress, uint8_t *pBuffer, uint32_t BufferSize);
00116 HAL_StatusTypeDef EEPROM_IO_IsDeviceReady(uint16_t DevAddress, uint32_t Trials);
00117
00118 /**
00119  * @}
00120 */
00121
00122 /**
00123  * @}
00124 */
00125
00126 /**
00127  * @}
00128 */
00129
00130 /**
00131  * @}
00132 */
00133
00134 #ifdef __cplusplus
00135 }
00136 #endif
00137
00138 #endif /* __STM32746G_DISCOVERY_EEPROM_H */
00139
00140 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```



STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_eeprom.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_eeprom.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file provides a set of functions needed to manage an I2C M24LR64
00008 *           EEPROM memory.
00009 @verbatim
00010           To be able to use this driver,
00011           the switch EE_M24LR64 must be defined
00012           in your toolchain compiler prep
00013           processor
00014           =====
00015           Notes:
00016           - The I2C EEPROM memory (M24LR
00017             64) is available on separate daughter
00018             board ANT7-M24LR-A, which is
```

not provided with the STM32746G_DISCOVERY
00017 board.
00018 To use this driver you have
to connect the ANT7-M24LR-A to CN3
00019 connector of STM32746G_DISCO
VERY board.

00020 =====
=====

00021
00022 It implements a high level communication layer for read and write
00023 from/to this memory. The needed
STM32F7xx hardware resources (I2C and
00024 GPIO) are defined in stm32746g_
discovery.h file, and the initialization is
00025 performed in EEPROM_IO_Init() function declared in stm32746g_discovery.c
00026 file.

00027 You can easily tailor this driver to any other development board,
00028 by just adapting the defines for hardware resources and
00029 EEPROM_IO_Init() function.

00030
00031 @note In this driver, basic read and write functions (BSP_EEPROM_ReadBuffer()
00032 and BSP_EEPROM_WritePage()
) use DMA mode to perform the data
00033 transfer to/from EEPROM memory.

00034
00035 @note Regarding BSP_EEPROM_WritePage(), it is an optimized function to perform
00036 small write (less than 1 page) BUT the number of bytes (combined to write start address) must not
00037 cross the EEPROM page bou

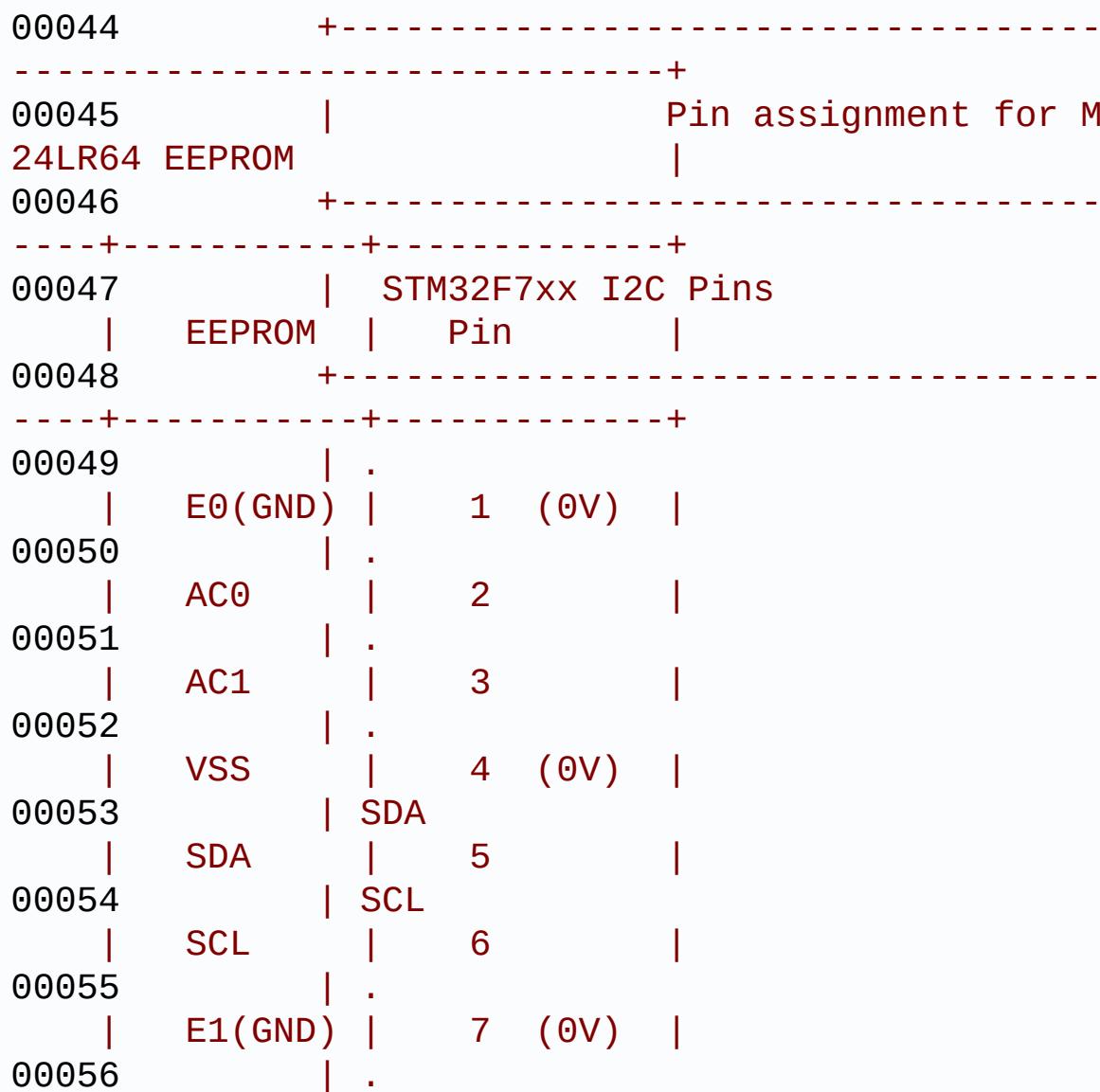
ndary. This function can only writes into
00038 the boundaries of an EEPROM
0M page.

00039 This function doesn't check on boundaries condition (in this driver
00040 the function BSP_EEPROM_WriteBuffer() which calls BSP_EEPROM_WritePage() is

00041 responsible of checking on Page boundaries).

00042

00043



		VDD		8 (3.3V)	
00057				+-----	-
-	-	-	-	-	-
00058		@endverbatim			
00059		*****			

00060		* @attention			
00061		*			
00062		* <h2><center>© COPYRIGHT(c) 2016 STM icroelectronics</center></h2>			
00063		*			
00064		* Redistribution and use in source and bin ary forms, with or without modification,			
00065		* are permitted provided that the followin g conditions are met:			
00066		* 1. Redistributions of source code must retain the above copyright notice,			
00067		* this list of conditions and the fol lowing disclaimer.			
00068		* 2. Redistributions in binary form must reproduce the above copyright notice,			
00069		* this list of conditions and the fol lowing disclaimer in the documentation			
00070		* and/or other materials provided wit h the distribution.			
00071		* 3. Neither the name of STMicroelectron ics nor the names of its contributors			
00072		* may be used to endorse or promote p roducts derived from this software			
00073		* without specific prior written perm ission.			
00074		*			
00075		* THIS SOFTWARE IS PROVIDED BY THE COPYRIG HT HOLDERS AND CONTRIBUTORS "AS IS"			
00076		* AND ANY EXPRESS OR IMPLIED WARRANTIES, I NCLUDING, BUT NOT LIMITED TO, THE			
00077		* IMPLIED WARRANTIES OF MERCHANTABILITY AN			

D FITNESS FOR A PARTICULAR PURPOSE ARE
00078 * DISCLAIMED. IN NO EVENT SHALL THE COPYRI-
GHT HOLDER OR CONTRIBUTORS BE LIABLE
00079 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP-
ECIAL, EXEMPLARY, OR CONSEQUENTIAL
00080 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR
00081 * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER
00082 * CAUSED AND ON ANY THEORY OF LIABILITY, W-
HETHER IN CONTRACT, STRICT LIABILITY,
00083 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI-
SE) ARISING IN ANY WAY OUT OF THE USE
00084 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

00085 *

00086 ****

00087 */

00088 /* Includes -----
-----*/

00089 #include "stm32746g_discovery_eeprom.h"

00090

00091 /** @addtogroup BSP
00092 * {@
00093 */

00094

00095 /** @addtogroup STM32746G_DISCOVERY
00096 * {@
00097 */

00098

00099 /** @addtogroup STM32746G_DISCOVERY_EEPROM
00100 * @brief This file includes the I2C EEPROM
driver of STM32746G-Discovery board.

00101 * {@
00102 */

00103

```
00104 /** @defgroup STM32746G_DISCOVERY_EEPROM_Private_Types STM32746G_DISCOVERY_EEPROM_Private_Types
00105 * @{
00106 */
00107 /**
00108 * @}
00109 */
00110
00111 /** @defgroup STM32746G_DISCOVERY_EEPROM_Private_Defines STM32746G_DISCOVERY_EEPROM_Private_Defines
00112 * @{
00113 */
00114 /**
00115 * @}
00116 */
00117
00118 /** @defgroup STM32746G_DISCOVERY_EEPROM_Private_Macros STM32746G_DISCOVERY_EEPROM_Private_Macros
00119 * @{
00120 */
00121 /**
00122 * @}
00123 */
00124
00125 /** @defgroup STM32746G_DISCOVERY_EEPROM_Private_Variables STM32746G_DISCOVERY_EEPROM_Private_Variables
00126 * @{
00127 */
00128 __IO uint16_t EEPROMAddress = 0;
00129 __IO uint16_t EEPROMDataRead;
00130 __IO uint8_t EEPROMDataWrite;
00131 /**
00132 * @}
```

```
00133  */
00134
00135 /** @defgroup STM32746G_DISCOVERY_EEPROM_Private_Function_Proto
types STM32746G_DISCOVERY_EEPROM Private Function Prototypes
00136  * @{
00137  */
00138 /**
00139  * @}
00140  */
00141
00142 /** @defgroup STM32746G_DISCOVERY_EEPROM_Exported_Functions STM32746G_DISCOVERY_EEPROM Exported Functions
00143  * @{
00144  */
00145
00146 /**
00147  * @brief Initializes peripherals used by the I2C EEPROM driver.
00148  *
00149  * @note There are 2 different versions of M24LR64 (A01 & A02).
00150  *           Then try to connect on 1st one (EEPROM_I2C_ADDRESS_A01)
00151  *           and if problem, check the 2nd one (EEPROM_I2C_ADDRESS_A02)
00152  *           @retval EEPROM_OK (0) if operation is correctly performed, else return value
00153  *           different from EEPROM_OK (0)
00154  */
00155 uint32_t BSP_EEPROM_Init(void)
00156 {
00157     /* I2C Initialization */
00158     EEPROM_IO_Init();
00159
00160     /* Select the EEPROM address for A01 and c
```

```
heck if OK */
00161     EEPROMAddress = EEPROM_I2C_ADDRESS_A01;
00162     if(EEPROM_IO_IsDeviceReady(EEPROMAddress,
EEPROM_MAX_TRIALS) != HAL_OK)
00163     {
00164         /* Select the EEPROM address for A02 and
check if OK */
00165         EEPROMAddress = EEPROM_I2C_ADDRESS_A02;
00166         if(EEPROM_IO_IsDeviceReady(EEPROMAddress
, EEPROM_MAX_TRIALS) != HAL_OK)
00167         {
00168             return EEPROM_FAIL;
00169         }
00170     }
00171     return EEPROM_OK;
00172 }
00173
00174 /**
00175 * @brief DeInitializes the EEPROM.
00176 * @retval EEPROM state
00177 */
00178 uint8_t BSP_EEPROM_DeInit(void)
00179 {
00180     /* I2C won't be disabled because common to
other functionalities */
00181     return EEPROM_OK;
00182 }
00183
00184 /**
00185 * @brief Reads a block of data from the E
EEPROM.
00186 * @param pBuffer: pointer to the buffer t
hat receives the data read from
00187 *                 the EEPROM.
00188 * @param ReadAddr: EEPROM's internal addr
ess to start reading from.
00189 * @param NumByteToRead: pointer to the va
```

```
riable holding number of bytes to
00190      *          be read from the EEPROM.
00191      *
00192      *          @note The variable pointed by Num
ByteToRead is reset to 0 when all the
00193      *          data are read from the EEPROM. Application should monitor this
00194      *          variable in order know when
the transfer is complete.
00195      *
00196      * @retval EEPROM_OK (0) if operation is co
rrectly performed, else return value
00197      *          different from EEPROM_OK (0) or
the timeout user callback.
00198      */
00199 uint32_t BSP_EEPROM_ReadBuffer(uint8_t* pBuffer, uint16_t ReadAddr, uint16_t* NumByteToRead)
00200 {
00201     uint32_t buffersize = *NumByteToRead;
00202
00203     /* Set the pointer to the Number of data t
o be read. This pointer will be used
00204         by the DMA Transfer Completer interrupt
Handler in order to reset the
00205         variable to 0. User should check on thi
s variable in order to know if the
00206         DMA transfer has been complete or not.
*/
00207     EEPROMDataRead = *NumByteToRead;
00208
00209     if(EEPROM_IO_ReadData(EEPROMAddress, ReadA
ddr, pBuffer, buffersize) != HAL_OK)
00210     {
00211         BSP_EEPROM_TIMEOUT_UserCallback();
00212         return EEPROM_FAIL;
00213     }
00214
```

```
00215     /* If all operations OK, return EEPROM_OK
00216     (0) */
00217 }
00218
00219 /**
00220     * @brief Writes more than one byte to the
00221     * EEPROM with a single WRITE cycle.
00222     *
00223     * @note The number of bytes (combined to
00224     * write start address) must not
00225     * cross the EEPROM page boundary.
00226     * This function can only write into
00227     * the boundaries of an EEPROM page.
00228     *
00229     * @param pBuffer: pointer to the buffer c
00230     * ontaining the data to be written to
00231     * the EEPROM.
00232     * @param WriteAddr: EEPROM's internal add
00233     * ress to write to.
00234     *
00235     * @note The variable pointed by Num
00236     * ByteToWrite is reset to 0 when all the
00237     * data are written to the EEPROM. Application should monitor this
00238     * variable in order know when
00239     * the transfer is complete.
```

```
00238      *
00239      *           @note This function just configur
e the communication and enable the DMA
00240      *           channel to transfer data. M
eanwhile, the user application may perform
00241      *           other tasks in parallel.
00242      *
00243      * @retval EEPROM_OK (0) if operation is co
rrectly performed, else return value
00244      *           different from EEPROM_OK (0) or
the timeout user callback.
00245      */
00246 uint32_t BSP_EEPROM_WritePage(uint8_t* pBuffer,
00247                               uint16_t WriteAddr, uint8_t* NumByteToWrite)
00247 {
00248     uint32_t buffersize = *NumByteToWrite;
00249     uint32_t status = EEPROM_OK;
00250
00251     /* Set the pointer to the Number of data t
o be written. This pointer will be used
00252         by the DMA Transfer Completer interrupt
t Handler in order to reset the
00253             variable to 0. User should check on th
is variable in order to know if the
00254                 DMA transfer has been complete or not.
00255 */
00255     EEPROMDataWrite = *NumByteToWrite;
00256
00257     if(EEPROM_IO_WriteData(EEPROMAddress, Writ
eAddr, pBuffer, buffersize) != HAL_OK)
00258     {
00259         BSP_EEPROM_TIMEOUT_UserCallback();
00260         status = EEPROM_FAIL;
00261     }
00262
00263     if(BSP_EEPROM_WaitEepromStandbyState() !=
EEPROM_OK)
```

```
00264  {
00265      return EEPROM_FAIL;
00266  }
00267
00268  /* If all operations OK, return EEPROM_OK
00269  (0) */
00270  return status;
00271
00272 /**
00273  * @brief Writes buffer of data to the I2C
00274  * EEPROM.
00275  * @param pBuffer: pointer to the buffer
00276  * containing the data to be written
00277  * to the EEPROM.
00278  * @param WriteAddr: EEPROM's internal address
00279  * to write to.
00280  * @param NumByteToWrite: number of bytes
00281  * to write to the EEPROM.
00282  * @retval EEPROM_OK (0) if operation is correctly
00283  * performed, else return value
00284  * different from EEPROM_OK (0) or
00285  * the timeout user callback.
00286  */
00287 uint32_t BSP_EEPROM_WriteBuffer(uint8_t *pBuffer,
00288  uint16_t WriteAddr, uint16_t NumByteToWrite)
00289 {
00290     uint16_t numofpage = 0, numofsingle = 0, count = 0;
00291     uint16_t addr = 0;
00292     uint8_t dataindex = 0;
00293     uint32_t status = EEPROM_OK;
00294
00295     addr = WriteAddr % EEPROM_PAGESIZE;
00296     count = EEPROM_PAGESIZE - addr;
00297     numofpage = NumByteToWrite / EEPROM_PAGESIZE;
```

```
00291     numofsingle = NumByteToWrite % EEPROM_PAGE  
SIZE;  
00292     /* If WriteAddr is EEPROM_PAGESIZE aligned  
 */  
00293     if(addr == 0)  
00294     {  
00295         /* If NumByteToWrite < EEPROM_PAGESIZE */  
  
00296         if(numofpage == 0)  
00297         {  
00298             /* Store the number of data to be writ  
ten */  
00299             dataindex = numofsingle;  
00300             /* Start writing data */  
00301             status = BSP_EEPROM_WritePage(pBuffer,  
WriteAddr, (uint8_t*)(&dataindex));  
00302             if(status != EEPROM_OK)  
00303             {  
00304                 return status;  
00305             }  
00306         }  
00307     }  
00308     /* If NumByteToWrite > EEPROM_PAGESIZE */  
  
00309     else  
00310     {  
00311         while(numofpage--)  
00312         {  
00313             /* Store the number of data to be wr  
itten */  
00314             dataindex = EEPROM_PAGESIZE;  
00315             status = BSP_EEPROM_WritePage(pBuffe  
r, WriteAddr, (uint8_t*)(&dataindex));  
00316             if(status != EEPROM_OK)  
00317             {  
00318                 return status;  
00319             }  
00320     }  
00321 }
```

```
00320
00321             WriteAddr += EEPROM_PAGESIZE;
00322             pBuffer += EEPROM_PAGESIZE;
00323         }
00324
00325     if(numofsingle!=0)
00326     {
00327         /* Store the number of data to be wr
itten */
00328         dataindex = numofsingle;
00329         status = BSP_EEPROM_WritePage(pBuffe
r, WriteAddr, (uint8_t*)(&dataindex));
00330         if(status != EEPROM_OK)
00331         {
00332             return status;
00333         }
00334     }
00335 }
00336 }
00337 /* If WriteAddr is not EEPROM_PAGESIZE ali
gned */
00338 else
00339 {
00340     /* If NumByteToWrite < EEPROM_PAGESIZE */

00341     if(numofpage== 0)
00342     {
00343         /* If the number of data to be written
is more than the remaining space
in the current page: */
00344         if(NumByteToWrite > count)
00345         {
00346             /* Store the number of data to be wr
itten */
00347             dataindex = count;
00348             /* Write the data contained in same
page */
```

```
00350         status = BSP_EEPROM_WritePage(pBuffe
r, WriteAddr, (uint8_t*)(&dataindex));
00351         if(status != EEPROM_OK)
00352         {
00353             return status;
00354         }
00355
00356         /* Store the number of data to be wr
itten */
00357         dataindex = (NumByteToWrite - count)
00358         ;
00359         /* Write the remaining data in the f
ollowing page */
00360         status = BSP_EEPROM_WritePage((uint8
_t*)(pBuffer + count), (WriteAddr + count), (uint8
_t*)(&dataindex));
00361         if(status != EEPROM_OK)
00362         {
00363             return status;
00364         }
00365     else
00366     {
00367         /* Store the number of data to be wr
itten */
00368         dataindex = numofsingle;
00369         status = BSP_EEPROM_WritePage(pBuffe
r, WriteAddr, (uint8_t*)(&dataindex));
00370         if(status != EEPROM_OK)
00371         {
00372             return status;
00373         }
00374     }
00375 }
00376 /* If NumByteToWrite > EEPROM_PAGESIZE */
00377 else
```

```
00378     {
00379         NumByteToWrite -= count;
00380         numofpage =  NumByteToWrite / EEPROM_P
AGESIZE;
00381         numofsingle = NumByteToWrite % EEPROM_
PAGESIZE;
00382
00383         if(count != 0)
00384     {
00385         /* Store the number of data to be wr
itten */
00386         dataindex = count;
00387         status = BSP_EEPROM_WritePage(pBuffe
r, WriteAddr, (uint8_t*)(&dataindex));
00388         if(status != EEPROM_OK)
00389     {
00390         return status;
00391     }
00392         WriteAddr += count;
00393         pBuffer += count;
00394     }
00395
00396     while(numofpage--)
00397     {
00398         /* Store the number of data to be wr
itten */
00399         dataindex = EEPROM_PAGESIZE;

00400         status = BSP_EEPROM_WritePage(pBuffe
r, WriteAddr, (uint8_t*)(&dataindex));
00401         if(status != EEPROM_OK)
00402     {
00403         return status;
00404     }
00405         WriteAddr += EEPROM_PAGESIZE;
00406         pBuffer += EEPROM_PAGESIZE;
00407     }
```

```
00408     if(numofsingle != 0)
00409     {
00410         /* Store the number of data to be wr
itten */
00411         dataindex = numofsingle;
00412         status = BSP_EEPROM_WritePage(pBuffe
r, WriteAddr, (uint8_t*)(&dataindex));
00413         if(status != EEPROM_OK)
00414         {
00415             return status;
00416         }
00417     }
00418 }
00419 }
00420
00421     /* If all operations OK, return EEPROM_OK
(0) */
00422     return EEPROM_OK;
00423 }
00424
00425 /**
00426     * @brief Wait for EEPROM Standby state.
00427     *
00428     * @note This function allows to wait and
check that EEPROM has finished the
00429     *       last operation. It is mostly used
after Write operation: after receiving
00430     *       the buffer to be written, the EEP
ROM may need additional time to actually
00431     *       perform the write operation. Duri
ng this time, it doesn't answer to
00432     *       I2C packets addressed to it. Once
the write operation is complete
00433     *       the EEPROM responds to its addres
s.
00434     *
00435     * @retval EEPROM_OK (0) if operation is co
```

```
rrectly performed, else return value
00436     * different from EEPROM_OK (0) or
the timeout user callback.
00437 */
00438 uint32_t BSP_EEPROM_WaitEepromStandbyState(v
oid)
00439 {
00440     /* Check if the maximum allowed number of
trials has bee reached */
00441     if(EEPROM_IO_IsDeviceReady(EEPROMAddress,
EEPROM_MAX_TRIALS) != HAL_OK)
00442     {
00443         /* If the maximum number of trials has b
een reached, exit the function */
00444         BSP_EEPROM_TIMEOUT_UserCallback();
00445         return EEPROM_TIMEOUT;
00446     }
00447     return EEPROM_OK;
00448 }
00449
00450 /**
00451     * @brief Basic management of the timeout
situation.
00452     * @retval None
00453 */
00454 __weak void BSP_EEPROM_TIMEOUT_UserCallback(
void)
00455 {
00456 }
00457
00458 /**
00459     * @}
00460 */
00461
00462 /**
00463     * @}
00464 */
```

```
00465
00466 /**
00467     * @}
00468     */
00469
00470 /**
00471     * @}
00472     */
00473
00474 /***** (C) COPYRIGHT STMi
croelectronics *****END OF FILE****/
```

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_sdram.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_sdram.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains the common d
efines and functions prototypes for
00008 *           the stm32746g_discovery_sdram.c
driver.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and bin
ary forms, with or without modification,
00015 * are permitted provided that the followin
g conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038
00039 /* Define to prevent recursive inclusion ---
00040 -----
00041 #ifndef __STM32746G_DISCOVERY_SDRAM_H
00042 #define __STM32746G_DISCOVERY_SDRAM_H
00043
00044 #ifdef __cplusplus
00045   extern "C" {
00046 #endif
00047
00048 /* Includes -----
00049
00050 /** @addtogroup BSP
00051   * @{
00052   */
00053
00054 /** @addtogroup STM32746G_DISCOVERY
00055   * @{
00056   */
00057
00058 /** @addtogroup STM32746G_DISCOVERY_SDRAM
00059   * @{
00060   */
00061
00062 /** @defgroup STM32746G_DISCOVERY_SDRAM_Expo
00063   * @{
00064   */
00065
00066 /**
```

```

00067  * @brief  SDRAM status structure definition
00068  */
00069 #define    SDRAM_OK          ((uint8_t)0x00)
00070 #define    SDRAM_ERROR        ((uint8_t)0x01)
00071
00072 /** @defgroup STM32746G_DISCOVERY_SDRAM_Exported_Constants STM32746G_DISCOVERY_SDRAM Exported Constants
00073  * @{
00074  */
00075 #define SDRAM_DEVICE_ADDR   ((uint32_t)0xC0000000)
00076 #define SDRAM_DEVICE_SIZE   ((uint32_t)0x800000) /* SDRAM device size in MBytes */
00077
00078 /* #define SDRAM_MEMORY_WIDTH          FMC
   _SDRAM_MEM_BUS_WIDTH_8   */
00079 #define SDRAM_MEMORY_WIDTH          FMC
   _SDRAM_MEM_BUS_WIDTH_16
00080
00081 #define SDCLOCK_PERIOD           FMC
   _SDRAM_CLOCK_PERIOD_2
00082 /* #define SDCLOCK_PERIOD           FMC
   _SDRAM_CLOCK_PERIOD_3   */
00083
00084 #define REFRESH_COUNT          (((uint32_t)0x0603) /* SDRAM refresh counter (100Mhz SD clock) */
00085
00086 #define SDRAM_TIMEOUT           (((uint32_t)0xFFFF))
00087
00088 /* DMA definitions for SDRAM DMA transfer */
00089 #define __DMAx_CLK_ENABLE         —
HAL_RCC_DMA2_CLK_ENABLE
00090 #define __DMAx_CLK_DISABLE        —

```

```
HAL_RCC_DMA2_CLK_DISABLE
00091 #define SDRAM_DMAX_CHANNEL DM
A_CHANNEL_0
00092 #define SDRAM_DMAX_STREAM DM
A2_Stream0
00093 #define SDRAM_DMAX_IRQn DM
A2_Stream0_IRQn
00094 #define BSP_SDRAM_DMA_IRQHandler DM
A2_Stream0_IRQHandler
00095 /**
00096  * @}
00097 */
00098 /**
00099 */
00100  * @brief FMC SDRAM Mode definition register defines
00101 */
00102 #define SDRAM_MODEREG_BURST_LENGTH_1
        ((uint16_t)0x0000)
00103 #define SDRAM_MODEREG_BURST_LENGTH_2
        ((uint16_t)0x0001)
00104 #define SDRAM_MODEREG_BURST_LENGTH_4
        ((uint16_t)0x0002)
00105 #define SDRAM_MODEREG_BURST_LENGTH_8
        ((uint16_t)0x0004)
00106 #define SDRAM_MODEREG_BURST_TYPE_SEQUENTIAL
        ((uint16_t)0x0000)
00107 #define SDRAM_MODEREG_BURST_TYPE_INTERLEAVED
        ((uint16_t)0x0008)
00108 #define SDRAM_MODEREG_CAS_LATENCY_2
        ((uint16_t)0x0020)
00109 #define SDRAM_MODEREG_CAS_LATENCY_3
        ((uint16_t)0x0030)
00110 #define SDRAM_MODEREG_OPERATING_MODE_STANDAR D
        ((uint16_t)0x0000)
00111 #define SDRAM_MODEREG_WRITEBURST_MODE_PROGRAMMED
        ((uint16_t)0x0000)
```

```
00112 #define SDRAM_MODEREG_WRITEBURST_MODE_SINGLE  
        ((uint16_t)0x0200)  
00113 /**  
00114     * @}  
00115     */  
00116  
00117 /** @defgroup STM32746G_DISCOVERY_SDRAM_Expo  
rted_Macro STM32746G_DISCOVERY_SDRAM Exported Macro  
00118     * @{  
00119     */  
00120 /**  
00121     * @}  
00122     */  
00123  
00124 /** @addtogroup STM32746G_DISCOVERY_SDRAM_Ex  
ported_Functions  
00125     * @{  
00126     */  
00127 uint8_t BSP_SDRAM_Init(void);  
00128 uint8_t BSP_SDRAM_DeInit(void);  
00129 void    BSP_SDRAM_Initialization_sequence(ui  
nt32_t RefreshCount);  
00130 uint8_t BSP_SDRAM_ReadData(uint32_t uwStartA  
ddress, uint32_t *pData, uint32_t uwDataSize);  
00131 uint8_t BSP_SDRAM_ReadData_DMA(uint32_t uwSt  
artAddress, uint32_t *pData, uint32_t uwDataSize);  
00132 uint8_t BSP_SDRAM_WriteData(uint32_t uwStart  
Address, uint32_t *pData, uint32_t uwDataSize);  
00133 uint8_t BSP_SDRAM_WriteData_DMA(uint32_t uwS  
tartAddress, uint32_t *pData, uint32_t uwDataSize)  
;  
00134 uint8_t BSP_SDRAM_Sendcmd(FMC_SDRAM_CommandT  
ypeDef *SdramCmd);  
00135  
00136 /* These functions can be modified in case t  
he current settings (e.g. DMA stream)
```

```
00137     need to be changed for specific application needs */
00138 void      BSP_SDRAM_MspInit(SDRAM_HandleTypeDefDef *hsdram, void *Params);
00139 void      BSP_SDRAM_MspDeInit(SDRAM_HandleTypeDefDef *hsdram, void *Params);
00140
00141
00142 /**
00143  * @}
00144 */
00145
00146 /**
00147  * @}
00148 */
00149
00150 /**
00151  * @}
00152 */
00153
00154 /**
00155  * @}
00156 */
00157
00158 #ifdef __cplusplus
00159 }
00160 #endif
00161
00162 #endif /* __STM32746G_DISCOVERY_SDRAM_H */
00163
00164 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_sd.h

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_sd.h
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file contains the common d
efines and functions prototypes for
00008 *           the stm32746g_discovery_sd.c dr
iver.
00009 * ****
00010 * @attention
00011 *
00012 * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00013 *
00014 * Redistribution and use in source and bin
ary forms, with or without modification,
00015 * are permitted provided that the followin
g conditions are met:
```

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

```
00035      *
00036      ****
00037      ****
00038
00039 /* Define to prevent recursive inclusion ---
00040 -----
00041 #ifndef __STM32746G_DISCOVERY_SD_H
00042 #define __STM32746G_DISCOVERY_SD_H
00043
00044 #ifdef __cplusplus
00045   extern "C" {
00046 #endif
00047
00048 /* Includes -----
00049 -----
00050 /** @addtogroup BSP
00051   * @{
00052   */
00053
00054 /** @addtogroup STM32746G_DISCOVERY
00055   * @{
00056   */
00057
00058 /** @addtogroup STM32746G_DISCOVERY_SD
00059   * @{
00060   */
00061
00062 /** @defgroup STM32746G_DISCOVERY_SD_Exported_Types
00063   * @{
00064   */
00065
00066 /**
00067   * @brief SD Card information structure
```

```
00068      */
00069 #define BSP_SD_CardInfo HAL_SD_CardInfoTypeD
ef
00070 /**
00071   * @}
00072 */
00073
00074 /**
00075   * @brief SD status structure definition
00076 */
00077 #define MSD_OK           ((uint
8_t)0x00)
00078 #define MSD_ERROR        ((uint
8_t)0x01)
00079 #define MSD_ERROR_SD_NOT_PRESENT ((uint
8_t)0x02)
00080
00081 /**
00082   * @brief SD transfer state definition
00083 */
00084 #define SD_TRANSFER_OK    ((ui
nt8_t)0x00)
00085 #define SD_TRANSFER_BUSY  ((ui
nt8_t)0x01)
00086
00087 /** @defgroup STM32746G_DISCOVERY_SD_Exporte
d_Constants STM32746G_DISCOVERY_SD Exported Consta
nts
00088   * @{
00089   */
00090 #define SD_PRESENT        ((uint8_t)0
x01)
00091 #define SD_NOT_PRESENT    ((uint8_t)0
x00)
00092
00093 #define SD_DATATIMEOUT   ((uint32_t)
100000000)
```

```
00094
00095 /* DMA definitions for SD DMA transfer */
00096 #define __DMAx_TxRx_CLK_ENABLE —
HAL_RCC_DMA2_CLK_ENABLE
00097 #define SD_DMAx_Tx_CHANNEL DM
A_CHANNEL_4
00098 #define SD_DMAx_Rx_CHANNEL DM
A_CHANNEL_4
00099 #define SD_DMAx_Tx_STREAM DM
A2_Stream6
00100 #define SD_DMAx_Rx_STREAM DM
A2_Stream3
00101 #define SD_DMAx_Tx_IRQHandler DM
A2_Stream6_IRQHandler
00102 #define SD_DMAx_Rx_IRQHandler DM
A2_Stream3_IRQHandler
00103 #define BSP_SDMMC_IRQHandler SD
MMC1_IRQHandler
00104 #define BSP_SDMMC_DMA_Tx_IRQHandler DM
A2_Stream6_IRQHandler
00105 #define BSP_SDMMC_DMA_Rx_IRQHandler DM
A2_Stream3_IRQHandler
00106 #define SD_DetectIRQHandler() HA
L_GPIO_EXTI_IRQHandler(SD_DETECT_PIN)
00107 /**
00108 * @}
00109 */
00110
00111 /** @defgroup STM32746G_DISCOVERY_SD_Expor
d_Macro STM32746G_DISCOVERY_SD Exported Macro
00112 * @{
00113 */
00114 /**
00115 * @}
00116 */
00117
00118 /** @addtogroup STM32746G_DISCOVERY_SD_Expor
```

```
ted_Functions
00119 * @{
00120 /*
00121 uint8_t BSP_SD_Init(void);
00122 uint8_t BSP_SD_DeInit(void);
00123 uint8_t BSP_SD_ITConfig(void);
00124 uint8_t BSP_SD_ReadBlocks(uint32_t *pData, u
int32_t ReadAddr, uint32_t NumOfBlocks, uint32_t T
imeout);
00125 uint8_t BSP_SD_WriteBlocks(uint32_t *pData,
uint32_t WriteAddr, uint32_t NumOfBlocks, uint32_t
Timeout);
00126 uint8_t BSP_SD_ReadBlocks_DMA(uint32_t *pData
, uint32_t ReadAddr, uint32_t NumOfBlocks);
00127 uint8_t BSP_SD_WriteBlocks_DMA(uint32_t *pData
, uint32_t WriteAddr, uint32_t NumOfBlocks);
00128 uint8_t BSP_SD_Erase(uint32_t StartAddr, uin
t32_t EndAddr);
00129 uint8_t BSP_SD_GetCardState(void);
00130 void BSP_SD_GetCardInfo(HAL_SD_CardInfoTy
peDef *CardInfo);
00131 uint8_t BSP_SD_IsDetected(void);
00132
00133 /* These functions can be modified in case t
he current settings (e.g. DMA stream)
00134     need to be changed for specific applicati
on needs */
00135 void BSP_SD_MspInit(SD_HandleTypeDef *hsd
, void *Params);
00136 void BSP_SD_Detect_MspInit(SD_HandleTypeDef
*hsd, void *Params);
00137 void BSP_SD_MspDeInit(SD_HandleTypeDef *hsd
, void *Params);
00138 void BSP_SD_AbortCallback(void);
00139 void BSP_SD_WriteCpltCallback(void);
00140 void BSP_SD_ReadCpltCallback(void);
00141 /**
```

```
00142     * @}
00143     */
00144
00145 /**
00146     * @}
00147     */
00148
00149 /**
00150     * @}
00151     */
00152
00153 /**
00154     * @}
00155     */
00156
00157 #ifdef __cplusplus
00158 }
00159 #endif
00160
00161 #endif /* __STM32746G_DISCOVERY_SD_H */
00162
00163 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_sd.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_sd.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file includes the uSD card
00008 *          driver mounted on STM32746G-Discovery
00009 *          board.
00010 * @verbatim
00011 * 1. How To use this driver:
00012 * -----
00013 *      - This driver is used to drive the micro SD external card mounted on STM32746G-Discovery
00014 *          board.
00015 *      - This driver does not need a specific
00016 *          component driver for the micro SD device
00017 *          to be included with.
00018 * -----
```

00019 + Initialization steps:
00020 o Initialize the micro SD card using
the BSP_SD_Init() function. This
00021 function includes the MSP layer ha
rdware resources initialization and the
00022 SDIO interface configuration to in
terface with the external micro SD. It
00023 also includes the micro SD initial
ization sequence.
00024 o To check the SD card presence you
can use the function BSP_SD_IsDetected() which
00025 returns the detection status
00026 o If SD presence detection interrupt
mode is desired, you must configure the
00027 SD detection interrupt mode by cal
ling the function BSP_SD_ITConfig(). The interrupt
00028 is generated as an external interr
upt whenever the micro SD card is
00029 plugged/unplugged in/from the boar
d.
00030 o The function BSP_SD_GetCardInfo()
is used to get the micro SD card information
00031 which is stored in the structure "
HAL_SD_CardInfoTypedef".
00032
00033 + Micro SD card operations
00034 o The micro SD card can be accessed
with read/write block(s) operations once
00035 it is ready for access. The access
can be performed whether using the polling
00036 mode by calling the functions BSP_
SD_ReadBlocks()/BSP_SD_WriteBlocks(), or by DMA
00037 transfer using the functions BSP_S
D_ReadBlocks_DMA()/BSP_SD_WriteBlocks_DMA()
00038 o The DMA transfer complete is used
with interrupt mode. Once the SD transfer

```
00039           is complete, the SD interrupt is han-
00040           dled using the function BSP_SD_IRQHandler(),
00041           the DMA Tx/Rx transfer complete are
00042           handled using the functions
00043           BSP_SD_DMA_Tx_IRQHandler() / BSP_SD_
00044           DMA_Rx_IRQHandler(). The corresponding user callbacks
00045           are implemented by the user at application level.
00046           o The SD erase block(s) is performed
00047           using the function BSP_SD_Erase() with specifying
00048           the number of blocks to erase.
00049           o The SD runtime status is returned
00050           when calling the function BSP_SD_GetCardState().
00051
00052           @endverbatim
00053
00054
00055
00056
00057
00058
00059
```


* @attention
*
* <h2><center>© COPYRIGHT(c) 2016 STMicroelectronics</center></h2>
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.

```
00060      *   3. Neither the name of STMicroelectron
ics nor the names of its contributors
00061      *       may be used to endorse or promote p
roducts derived from this software
00062      *       without specific prior written perm
ission.
00063      *
00064      * THIS SOFTWARE IS PROVIDED BY THE COPYRIG
HT HOLDERS AND CONTRIBUTORS "AS IS"
00065      * AND ANY EXPRESS OR IMPLIED WARRANTIES, I
NCLUDING, BUT NOT LIMITED TO, THE
00066      * IMPLIED WARRANTIES OF MERCHANTABILITY AN
D FITNESS FOR A PARTICULAR PURPOSE ARE
00067      * DISCLAIMED. IN NO EVENT SHALL THE COPYRI
GHT HOLDER OR CONTRIBUTORS BE LIABLE
00068      * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP
ECIAL, EXEMPLARY, OR CONSEQUENTIAL
00069      * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR
00070      * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER
00071      * CAUSED AND ON ANY THEORY OF LIABILITY, W
HETHER IN CONTRACT, STRICT LIABILITY,
00072      * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI
SE) ARISING IN ANY WAY OUT OF THE USE
00073      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
00074      *
00075      ****
00076      */
00077
00078 /* Includes -----
----- */
00079 #include "stm32746g_discovery_sd.h"
00080
00081 /** @addtogroup BSP
```

```
00082      * @@
00083      */
00084
00085 /** @addtogroup STM32746G_DISCOVERY
00086      * @@
00087      */
00088
00089 /** @defgroup STM32746G_DISCOVERY_SD STM3274
00090   6G_DISCOVERY_SD
00091      * @@
00092
00093
00094 /** @defgroup STM32746G_DISCOVERY_SD_Private
00095   _TypesDefinitions STM32746G_DISCOVERY_SD Private T
00096   ypes Definitions
00097      * @@
00098      */
00099
00100
00101 /** @defgroup STM32746G_DISCOVERY_SD_Private
00102   _Defines STM32746G_DISCOVERY_SD Private Defines
00103      * @@
00104      */
00105
00106
00107
00108 /** @defgroup STM32746G_DISCOVERY_SD_Private
00109   _Macros STM32746G_DISCOVERY_SD Private Macros
00110      * @@
00111      */
00112
00113
```

```
00114
00115 /** @defgroup STM32746G_DISCOVERY_SD_Private
 _Variables STM32746G_DISCOVERY_SD Private Variables
00116     * @{
00117     */
00118 SD_HandleTypeDef uSdHandle;
00119
00120 /**
00121     * @}
00122     */
00123
00124 /** @defgroup STM32746G_DISCOVERY_SD_Private
 _FunctionPrototypes STM32746G_DISCOVERY_SD Private
 Function Prototypes
00125     * @{
00126     */
00127 /**
00128     * @}
00129     */
00130
00131 /** @defgroup STM32746G_DISCOVERY_SD_Exports
 _Functions STM32746G_DISCOVERY_SD Exported Functions
00132     * @{
00133     */
00134
00135 /**
00136     * @brief Initializes the SD card device.
00137     * @retval SD status
00138     */
00139 uint8_t BSP_SD_Init(void)
00140 {
00141     uint8_t sd_state = MSD_OK;
00142
00143     /* uSD device interface configuration */
00144     uSdHandle.Instance = SDMMC1;
```

```
00145
00146     uSdHandle.Init.ClockEdge          = SDMMC
00147         _CLOCK_EDGE_RISING;
00148     uSdHandle.Init.ClockBypass        = SDMMC
00149         _CLOCK_BYPASS_DISABLE;
00150     uSdHandle.Init.ClockPowerSave     = SDMMC
00151         _CLOCK_POWER_SAVE_DISABLE;
00152     uSdHandle.Init.BusWide            = SDMMC
00153         _BUS_WIDE_1B;
00154     uSdHandle.Init.HardwareFlowControl = SDMMC
00155         _HARDWARE_FLOW_CONTROL_DISABLE;
00156     uSdHandle.Init.ClockDiv           = SDMMC
00157         _TRANSFER_CLK_DIV;
00158
00159
00160     /* Msp SD Detect pin initialization */
00161     BSP_SD_Detect_MspInit(&uSdHandle, NULL);
00162     if(BSP_SD_IsDetected() != SD_PRESENT) /* Check if SD card is present */
00163     {
00164         return MSD_ERROR_SD_NOT_PRESENT;
00165     }
00166
00167
00168
00169     /* Msp SD initialization */
00170     BSP_SD_MspInit(&uSdHandle, NULL);
00171
00172
00173     /* HAL SD initialization */
00174     if(HAL_SD_Init(&uSdHandle) != HAL_OK)
00175     {
00176         sd_state = MSD_ERROR;
00177     }
00178
00179
00180     /* Configure SD Bus width */
00181     if(sd_state == MSD_OK)
00182     {
00183         /* Enable wide operation */
00184         if(HAL_SD_ConfigWideBusOperation(&uSdHandle, SDMMC_BUS_WIDE_4B) != HAL_OK)
```

```
00174     {
00175         sd_state = MSD_ERROR;
00176     }
00177     else
00178     {
00179         sd_state = MSD_OK;
00180     }
00181 }
00182
00183     return sd_state;
00184 }
00185
00186 /**
00187 * @brief DeInitializes the SD card device.
00188 * @retval SD status
00189 */
00190 uint8_t BSP_SD_DeInit(void)
00191 {
00192     uint8_t sd_state = MSD_OK;
00193
00194     uSdHandle.Instance = SDMMC1;
00195
00196     /* HAL SD deinitialization */
00197     if(HAL_SD_DeInit(&uSdHandle) != HAL_OK)
00198     {
00199         sd_state = MSD_ERROR;
00200     }
00201
00202     /* Msp SD deinitialization */
00203     uSdHandle.Instance = SDMMC1;
00204     BSP_SD_MspDeInit(&uSdHandle, NULL);
00205
00206     return sd_state;
00207 }
00208
00209 /**
```

```
00210     * @brief Configures Interrupt mode for SD
00211     * detection pin.
00212     * @retval Returns MSD_OK
00213 uint8_t BSP_SD_ITConfig(void)
00214 {
00215     GPIO_InitTypeDef gpio_init_structure;
00216
00217     /* Configure Interrupt mode for SD detecti
00218     on pin */
00219     gpio_init_structure.Pin = SD_DETECT_PIN;
00220     gpio_init_structure.Pull = GPIO_PULLUP;
00221     gpio_init_structure.Speed = GPIO_SPEED_FAST;
00222     gpio_init_structure.Mode = GPIO_MODE_IT_RISING_FALLING;
00223
00224     /* Enable and set SD detect EXTI Interrupt
00225     to the lowest priority */
00226     HAL_GPIO_Init(SD_DETECT_GPIO_PORT, &gpio_init_structure);
00227
00228     return MSD_OK;
00229 }
00230
00231 /**
00232     * @brief Detects if SD card is correctly
00233     * plugged in the memory slot or not.
00234     * @retval Returns if SD is detected or not
00235 uint8_t BSP_SD_IsDetected(void)
00236 {
00237     __IO uint8_t         status = SD_PRESENT;
```

```
00238
00239     /* Check SD card detect pin */
00240     if (HAL_GPIO_ReadPin(SD_DETECT_GPIO_PORT,
00241         SD_DETECT_PIN) == GPIO_PIN_SET)
00241     {
00242         status = SD_NOT_PRESENT;
00243     }
00244
00245     return status;
00246 }
00247
00248 /**
00249     * @brief Reads block(s) from a specified
00250     * address in an SD card, in polling mode.
00251     * @param pData: Pointer to the buffer tha
00252     * t will contain the data to transmit
00253     * @param ReadAddr: Address from where dat
00254     * a is to be read
00255     * @param NumOfBlocks: Number of SD blocks
00256     * to read
00257     * @param Timeout: Timeout for read operat
00258     * ion
00259     * @retval SD status
00260
00261     */
00262
00263     uint8_t BSP_SD_ReadBlocks(uint32_t *pData, u
00264     int32_t ReadAddr, uint32_t NumOfBlocks, uint32_t T
00265     imeout)
00266 {
00267     if(HAL_SD_ReadBlocks(&uSdHandle, (uint8_t
00268     *)pData, ReadAddr, NumOfBlocks, Timeout) != HAL_OK
00269     )
00270     {
00271         return MSD_ERROR;
00272     }
00273     else
00274     {
00275         return MSD_OK;
00276     }
00277 }
```

```
00265     }
00266 }
00267
00268 /**
00269   * @brief Writes block(s) to a specified address in an SD card, in polling mode.
00270   * @param pData: Pointer to the buffer that will contain the data to transmit
00271   * @param WriteAddr: Address from where data is to be written
00272   * @param NumOfBlocks: Number of SD blocks to write
00273   * @param Timeout: Timeout for write operation
00274   * @retval SD status
00275 */
00276 uint8_t BSP_SD_WriteBlocks(uint32_t *pData,
00277                               uint32_t WriteAddr, uint32_t NumOfBlocks, uint32_t Timeout)
00277 {
00278     if(HAL_SD_WriteBlocks(&uSdHandle, (uint8_t *)pData, WriteAddr, NumOfBlocks, Timeout) != HAL_OK)
00279     {
00280         return MSD_ERROR;
00281     }
00282     else
00283     {
00284         return MSD_OK;
00285     }
00286 }
00287
00288 /**
00289   * @brief Reads block(s) from a specified address in an SD card, in DMA mode.
00290   * @param pData: Pointer to the buffer that will contain the data to transmit
```

```
00291     * @param  ReadAddr: Address from where dat
a is to be read
00292     * @param  NumOfBlocks: Number of SD blocks
to read
00293     * @retval SD status
00294     */
00295 uint8_t BSP_SD_ReadBlocks_DMA(uint32_t *pData
a, uint32_t ReadAddr, uint32_t NumOfBlocks)
00296 {
00297     /* Read block(s) in DMA transfer mode */
00298     if(HAL_SD_ReadBlocks_DMA(&uSdHandle, (uint
8_t *)pData, ReadAddr, NumOfBlocks) != HAL_OK)
00299     {
00300         return MSD_ERROR;
00301     }
00302     else
00303     {
00304         return MSD_OK;
00305     }
00306 }
00307
00308 /**
00309     * @brief Writes block(s) to a specified address in an SD card, in DMA mode.
00310     * @param pData: Pointer to the buffer tha
t will contain the data to transmit
00311     * @param WriteAddr: Address from where da
ta is to be written
00312     * @param NumOfBlocks: Number of SD blocks
to write
00313     * @retval SD status
00314     */
00315 uint8_t BSP_SD_WriteBlocks_DMA(uint32_t *pData
a, uint32_t WriteAddr, uint32_t NumOfBlocks)
00316 {
00317     /* Write block(s) in DMA transfer mode */
00318     if(HAL_SD_WriteBlocks_DMA(&uSdHandle, (uin
```

```
t8_t * )pData, WriteAddr, NumOfBlocks) != HAL_OK)
00319     {
00320         return MSD_ERROR;
00321     }
00322     else
00323     {
00324         return MSD_OK;
00325     }
00326 }
00327
00328 /**
00329     * @brief Erases the specified memory area
00330     *        of the given SD card.
00331     * @param StartAddr: Start byte address
00332     * @param EndAddr: End byte address
00333     * @retval SD status
00334 */
00334 uint8_t BSP_SD_Erase(uint32_t StartAddr, uint32_t EndAddr)
00335 {
00336     if(HAL_SD_Erase(&uSdHandle, StartAddr, End
00337 Addr) != HAL_OK)
00338     {
00339         return MSD_ERROR;
00340     }
00341     else
00342     {
00343         return MSD_OK;
00344     }
00345
00346 /**
00347     * @brief Initializes the SD MSP.
00348     * @param hsd: SD handle
00349     * @param Params
00350     * @retval None
00351 */

```

```
00352 __weak void BSP_SD_MspInit(SD_HandleTypeDef *hsd, void *Params)
00353 {
00354     static DMA_HandleTypeDef dma_rx_handle;
00355     static DMA_HandleTypeDef dma_tx_handle;
00356     GPIO_InitTypeDef gpio_init_structure;
00357
00358     /* Enable SDIO clock */
00359     __HAL_RCC_SDMMC1_CLK_ENABLE();
00360
00361     /* Enable DMA2 clocks */
00362     __DMAx_TxRx_CLK_ENABLE();
00363
00364     /* Enable GPIOs clock */
00365     __HAL_RCC_GPIOC_CLK_ENABLE();
00366     __HAL_RCC_GPIOD_CLK_ENABLE();
00367
00368     /* Common GPIO configuration */
00369     gpio_init_structure.Mode      = GPIO_MODE_
AF_PP;
00370     gpio_init_structure.Pull      = GPIO_PULLU
P;
00371     gpio_init_structure.Speed    = GPIO_SPEED
_HIGH;
00372     gpio_init_structure.Alternate = GPIO_AF12_
SDMMC1;
00373
00374     /* GPIOC configuration */
00375     gpio_init_structure.Pin = GPIO_PIN_8 | GPIO_PIN_9 | GPIO_PIN_10 | GPIO_PIN_11 | GPIO_PIN_12;
00376     HAL_GPIO_Init(GPIOC, &gpio_init_structure);
00377
00378     /* GPIOD configuration */
00379     gpio_init_structure.Pin = GPIO_PIN_2;
00380     HAL_GPIO_Init(GPIOD, &gpio_init_structure);
00381 }
```

```
00381
00382     /* NVIC configuration for SDIO interrupts */
00383     HAL_NVIC_SetPriority(SDMMC1_IRQn, 0x0E, 0)
00384 ;
00385     HAL_NVIC_EnableIRQ(SDMMC1_IRQn);
00386
00387     /* Configure DMA Rx parameters */
00388     dma_rx_handle.Init.Channel = S
00389     D_DMAX_Rx_CHANNEL;
00390     dma_rx_handle.Init.Direction = D
00391     MA_PERIPH_TO_MEMORY;
00392     dma_rx_handle.InitPeriphInc = D
00393     MA_PINC_DISABLE;
00394     dma_rx_handle.InitMemInc = D
00395     MA_MINC_ENABLE;
00396     dma_rx_handle.InitPeriphDataAlignment = D
00397     MA_PDATAALIGN_WORD;
00398     dma_rx_handle.InitMemDataAlignment = D
00399     MA_MDATAALIGN_WORD;
00400     dma_rx_handle.InitMode = D
00401     MA_PFCTRL;
00402     dma_rx_handle.InitPriority = D
00403     MA_PRIORITY VERY_HIGH;
00404     dma_rx_handle.InitFIFOMode = D
00405     MA_FIFOMODE_ENABLE;
00406     dma_rx_handle.InitFIFOThreshold = D
00407     MA_FIFO_THRESHOLD_FULL;
00408     dma_rx_handle.InitMemBurst = D
00409     MA_MBURST_INC4;
00410     dma_rx_handle.InitPeriphBurst = D
00411     MA_PBURST_INC4;
00412
00413     dma_rx_handle.Instance = SD_DMAX_Rx_STREAM
00414 ;
00415
00416     /* Associate the DMA handle */
```

```
00403     __HAL_LINKDMA(hsd, hdmarx, dma_rx_handle);
00404
00405     /* Deinitialize the stream for new transfer */
00406     HAL_DMA_DeInit(&dma_rx_handle);
00407
00408     /* Configure the DMA stream */
00409     HAL_DMA_Init(&dma_rx_handle);
00410
00411     /* Configure DMA Tx parameters */
00412     dma_tx_handle.Init.Channel = S
D_MAX_Tx_CHANNEL;
00413     dma_tx_handle.Init.Direction = D
MA_MEMORY_TO_PERIPH;
00414     dma_tx_handle.Init.PeriphInc = D
MA_PINC_DISABLE;
00415     dma_tx_handle.Init.MemInc = D
MA_MINC_ENABLE;
00416     dma_tx_handle.InitPeriphDataAlignment = D
MA_PDATAALIGN_WORD;
00417     dma_tx_handle.Init.MemDataAlignment = D
MA_MDATAALIGN_WORD;
00418     dma_tx_handle.Init.Mode = D
MA_PFCTRL;
00419     dma_tx_handle.Init.Priority = D
MA_PRIORITY VERY HIGH;
00420     dma_tx_handle.Init.FIFOMode = D
MA_FIFOMODE_ENABLE;
00421     dma_tx_handle.Init.FIFOThreshold = D
MA_FIFO_THRESHOLD_FULL;
00422     dma_tx_handle.Init.MemBurst = D
MA_MBURST_INC4;
00423     dma_tx_handle.Init.PeriphBurst = D
MA_PBURST_INC4;
00424
00425     dma_tx_handle.Instance = SD_DMAX_Tx_STREAM
;
```

```

00426
00427     /* Associate the DMA handle */
00428     __HAL_LINKDMA(hsd, hdmatx, dma_tx_handle);
00429
00430     /* Deinitialize the stream for new transfer */
00431     HAL_DMA_DeInit(&dma_tx_handle);
00432
00433     /* Configure the DMA stream */
00434     HAL_DMA_Init(&dma_tx_handle);
00435
00436     /* NVIC configuration for DMA transfer complete interrupt */
00437     HAL_NVIC_SetPriority(SD_DMAx_Rx_IRQn, 0x0F, 0);
00438     HAL_NVIC_EnableIRQ(SD_DMAx_Rx_IRQn);
00439
00440     /* NVIC configuration for DMA transfer complete interrupt */
00441     HAL_NVIC_SetPriority(SD_DMAx_Tx_IRQn, 0x0F, 0);
00442     HAL_NVIC_EnableIRQ(SD_DMAx_Tx_IRQn);
00443 }
00444
00445 /**
00446     * @brief Initializes the SD Detect pin MSP.
00447     * @param hsd: SD handle
00448     * @param Params
00449     * @retval None
00450 */
00451 __weak void BSP_SD_Detect_MspInit(SD_HandleTypeDef *hsd, void *Params)
00452 {
00453     GPIO_InitTypeDef gpio_init_structure;
00454
00455     SD_DETECT_GPIO_CLK_ENABLE();

```

```
00456
00457     /* GPIO configuration in input for uSD_Detect
   signal */
00458     gpio_init_structure.Pin      = SD_DETECT_
PIN;
00459     gpio_init_structure.Mode    = GPIO_MODE_
INPUT;
00460     gpio_init_structure.Pull    = GPIO_PULLU
P;
00461     gpio_init_structure.Speed  = GPIO_SPEED
_HIGH;
00462     HAL_GPIO_Init(SD_DETECT_GPIO_PORT, &gpio_i
nit_structure);
00463 }
00464
00465 /**
00466     * @brief DeInitializes the SD MSP.
00467     * @param hsd: SD handle
00468     * @param Params
00469     * @retval None
00470 */
00471 __weak void BSP_SD_MspDeInit(SD_HandleTypeDef
*hsd, void *Params)
00472 {
00473     static DMA_HandleTypeDef dma_rx_handle;
00474     static DMA_HandleTypeDef dma_tx_handle;
00475
00476     /* Disable NVIC for DMA transfer complete
   interrupts */
00477     HAL_NVIC_DisableIRQ(SD_DMAx_Rx_IRQn);
00478     HAL_NVIC_DisableIRQ(SD_DMAx_Tx_IRQn);
00479
00480     /* Deinitialize the stream for new transfe
r */
00481     dma_rx_handle.Instance = SD_DMAx_Rx_STREAM
;
00482     HAL_DMA_DeInit(&dma_rx_handle);
```

```
00483
00484     /* Deinitialize the stream for new transfer */
00485     dma_tx_handle.Instance = SD_DMAX_Tx_STREAM;
00486     HAL_DMA_DeInit(&dma_tx_handle);
00487
00488     /* Disable NVIC for SDIO interrupts */
00489     HAL_NVIC_DisableIRQ(SDMMC1_IRQn);
00490
00491     /* DeInit GPIO pins can be done in the application
00492         (by surcharging this __weak function) */
00493
00494     /* Disable SDMMC1 clock */
00495     __HAL_RCC_SDMMC1_CLK_DISABLE();
00496
00497     /* GPIO pins clock and DMA clocks can be shut down in the application
00498         by surcharging this __weak function */
00499 }
00500
00501 /**
00502     * @brief Gets the current SD card data status.
00503     * @retval Data transfer state.
00504     *           This value can be one of the following values:
00505     *           @arg SD_TRANSFER_OK: No data transfer is acting
00506     *           @arg SD_TRANSFER_BUSY: Data transfer is acting
00507     */
00508 uint8_t BSP_SD_GetCardState(void)
00509 {
00510     return((HAL_SD_GetCardState(&uSdHandle) ==
```

```
    HAL_SD_CARD_TRANSFER ) ? SD_TRANSFER_OK : SD_TRANSFER_BUSY);
00511 }
00512
00513
00514 /**
00515     * @brief Get SD information about specific SD card.
00516     * @param CardInfo: Pointer to HAL_SD_CardInfoTypedef structure
00517     * @retval None
00518 */
00519 void BSP_SD_GetCardInfo(HAL_SD_CardInfoTypedef *CardInfo)
00520 {
00521     /* Get SD card Information */
00522     HAL_SD_GetCardInfo(&uSdHandle, CardInfo);
00523 }
00524
00525 /**
00526     * @brief SD Abort callbacks
00527     * @param hsd: SD handle
00528     * @retval None
00529 */
00530 void HAL_SD_AbortCallback(SD_HandleTypeDef *hsd)
00531 {
00532     BSP_SD_AbortCallback();
00533 }
00534
00535 /**
00536     * @brief Tx Transfer completed callbacks
00537     * @param hsd: SD handle
00538     * @retval None
00539 */
00540 void HAL_SD_TxCpltCallback(SD_HandleTypeDef *hsd)
```

```
00541 {
00542     BSP_SD_WriteCpltCallback();
00543 }
00544
00545 /**
00546     * @brief Rx Transfer completed callbacks
00547     * @param hsd: SD handle
00548     * @retval None
00549 */
00550 void HAL_SD_RxCpltCallback(SD_HandleTypeDefDef
*hsd)
00551 {
00552     BSP_SD_ReadCpltCallback();
00553 }
00554
00555 /**
00556     * @brief BSP SD Abort callbacks
00557     * @retval None
00558 */
00559 __weak void BSP_SD_AbortCallback(void)
00560 {
00561
00562 }
00563
00564 /**
00565     * @brief BSP Tx Transfer completed callbacks
00566     * @retval None
00567 */
00568 __weak void BSP_SD_WriteCpltCallback(void)
00569 {
00570
00571 }
00572
00573 /**
00574     * @brief BSP Rx Transfer completed callbacks
```

```
00575     * @retval None
00576     */
00577 __weak void BSP_SD_ReadCpltCallback(void)
00578 {
00579
00580 }
00581
00582 /**
00583     * @}
00584     */
00585
00586 /**
00587     * @}
00588     */
00589
00590 /**
00591     * @}
00592     */
00593
00594 /**
00595     * @}
00596     */
00597
00598 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
File List	Globals		
Drivers	BSP	STM32746G-Discovery	

stm32746g_discovery_sdram.c

Go to the documentation of this file.

```
00001 /**
00002 * ****
00003 * @file    stm32746g_discovery_sdram.c
00004 * @author  MCD Application Team
00005 * @version V2.0.0
00006 * @date    30-December-2016
00007 * @brief   This file includes the SDRAM driver for the MT48LC4M32B2B5-7 memory
00008 *           device mounted on STM32746G-Discovery board.
00009 * @verbatim
00010 1. How To use this driver:
00011 -----
00012 - This driver is used to drive the MT48LC4M32B2B5-7 SDRAM external memory mounted
00013 on STM32746G-Discovery board.
00014 - This driver does not need a specific component driver for the SDRAM device
00015           to be included with.
00016
00017 2. Driver description:
```

00018 -----
00019 + Initialization steps:
00020 o Initialize the SDRAM external memory using the BSP_SDRAM_Init() function. This
00021 function includes the MSP layer hardware resources initialization and the
00022 FMC controller configuration to interface with the external SDRAM memory.
00023 o It contains the SDRAM initialization sequence to program the SDRAM external
00024 device using the function BSP_SDRAM_Initialization_sequence(). Note that this
00025 sequence is standard for all SDRAM devices, but can include some differences
00026 from a device to another. If it is the case, the right sequence should be
00027 implemented separately.
00028
00029 + SDRAM read/write operations
00030 o SDRAM external memory can be accessed with read/write operations once it is
00031 initialized.
00032 Read/write operation can be performed with AHB access using the functions
00033 BSP_SDRAM_ReadData()/BSP_SDRAM_WriteData(), or by DMA transfer using the functions
00034 BSP_SDRAM_ReadData_DMA()/BSP_SDRAM_WriteData_DMA().
00035 o The AHB access is performed with 32-bit width transaction, the DMA transfer
00036 configuration is fixed at single (no burst) word transfer (see the
00037 SDRAM_MspInit() static function).
00038 o User can implement his own functions for read/write access with his desired
00039 configurations.
00040 o If interrupt mode is used for DMA

```
transfer, the function BSP_SDRAM_DMA_IRQHandler()
00041                  is called in IRQ handler file, to
serve the generated interrupt once the DMA
00042                  transfer is complete.
00043                  o You can send a command to the SDRA
M device in runtime using the function
00044                  BSP_SDRAM_Sendcmd(), and giving th
e desired command as parameter chosen between
00045                  the predefined commands of the "FM
C_SDRAM_CommandTypeDef" structure.
00046
00047      @endverbatim
00048  ****
*****  

00049  * @attention
00050  *
00051  * <h2><center>&copy; COPYRIGHT(c) 2016 STM
icroelectronics</center></h2>
00052  *
00053  * Redistribution and use in source and bin
ary forms, with or without modification,
00054  * are permitted provided that the followin
g conditions are met:
00055  *   1. Redistributions of source code must
retain the above copyright notice,
00056  *       this list of conditions and the fol
lowing disclaimer.
00057  *   2. Redistributions in binary form must
reproduce the above copyright notice,
00058  *       this list of conditions and the fol
lowing disclaimer in the documentation
00059  *       and/or other materials provided wit
h the distribution.
00060  *   3. Neither the name of STMicroelectron
ics nor the names of its contributors
00061  *       may be used to endorse or promote p
roducts derived from this software
```

```
00062      *      without specific prior written perm  
ission.  
00063      *  
00064      * THIS SOFTWARE IS PROVIDED BY THE COPYRIG  
HT HOLDERS AND CONTRIBUTORS "AS IS"  
00065      * AND ANY EXPRESS OR IMPLIED WARRANTIES, I  
NCLUDING, BUT NOT LIMITED TO, THE  
00066      * IMPLIED WARRANTIES OF MERCHANTABILITY AN  
D FITNESS FOR A PARTICULAR PURPOSE ARE  
00067      * DISCLAIMED. IN NO EVENT SHALL THE COPYRI  
GHT HOLDER OR CONTRIBUTORS BE LIABLE  
00068      * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SP  
ECIAL, EXEMPLARY, OR CONSEQUENTIAL  
00069      * DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
PROCUREMENT OF SUBSTITUTE GOODS OR  
00070      * SERVICES; LOSS OF USE, DATA, OR PROFITS;  
OR BUSINESS INTERRUPTION) HOWEVER  
00071      * CAUSED AND ON ANY THEORY OF LIABILITY, W  
HETHER IN CONTRACT, STRICT LIABILITY,  
00072      * OR TORT (INCLUDING NEGLIGENCE OR OTHERWI  
SE) ARISING IN ANY WAY OUT OF THE USE  
00073      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.  
00074      *  
00075      *****  
*****  
00076      */  
00077  
00078 /* Includes -----  
-----*/  
00079 #include "stm32746g_discovery_sram.h"  
00080  
00081 /** @addtogroup BSP  
00082      * @{  
00083      */  
00084  
00085 /** @addtogroup STM32746G_DISCOVERY
```

```
00086      * @@
00087      */
00088
00089 /**
00090  ** @defgroup STM32746G_DISCOVERY_SDRAM STM3
00091  ** @{
00092  */
00093 /**
00094  ** @defgroup STM32746G_DISCOVERY_SDRAM_Priv
00095  ** ate_Types_Definitions STM32746G_DISCOVERY_SDRAM Pr
00096  ** ivate Types Definitions
00097  */
00098 /**
00099
00100 /**
00101  ** @defgroup STM32746G_DISCOVERY_SDRAM_Priv
00102  ** ate_Defines STM32746G_DISCOVERY_SDRAM Private Defi
00103  ** nes
00104  */
00105 /**
00106
00107 /**
00108  ** @defgroup STM32746G_DISCOVERY_SDRAM_Priv
00109  ** ate_Macros STM32746G_DISCOVERY_SDRAM Private Macros
00110 /**
00111  */
00112 /**
00113
00114 /**
00115  ** @defgroup STM32746G_DISCOVERY_SDRAM_Priv
00116  ** ate_Variables STM32746G_DISCOVERY_SDRAM Private Va
```

```
riables
00115     * @@
00116     */
00117 SDRAM_HandleTypeDef sdramHandle;
00118 static FMC_SDRAM_TimingTypeDef Timing;
00119 static FMC_SDRAM_CommandTypeDef Command;
00120 /**
00121     * @@
00122     */
00123
00124 /** @defgroup STM32746G_DISCOVERY_SDRAM_Private_Function_Proto
types STM32746G_DISCOVERY_SDRAM Private Function Prototypes
00125     * @@
00126     */
00127 /**
00128     * @@
00129     */
00130
00131 /** @defgroup STM32746G_DISCOVERY_SDRAM_Exported_Func
tions STM32746G_DISCOVERY_SDRAM Exported Functions
00132     * @@
00133     */
00134
00135 /**
00136     * @brief Initializes the SDRAM device.
00137     * @retval SDRAM status
00138     */
00139 uint8_t BSP_SDRAM_Init(void)
00140 {
00141     static uint8_t sdramstatus = SDRAM_ERROR;
00142     /* SDRAM device configuration */
00143     sdramHandle.Instance = FMC_SDRAM_DEVICE;
00144
00145     /* Timing configuration for 100Mhz as SD clock frequency (System clock is up to 200Mhz) */
```

```

00146     Timing.LoadToActiveDelay      = 2;
00147     Timing.ExitSelfRefreshDelay = 7;
00148     Timing.SelfRefreshTime     = 4;
00149     Timing.RowCycleDelay       = 7;
00150     Timing.WriteRecoveryTime  = 2;
00151     Timing.RPDelay            = 2;
00152     Timing.RCDDelay           = 2;
00153
00154     sdramHandle.Init.SDBank          = FMC_
SDRAM_BANK1;
00155     sdramHandle.Init.ColumnBitsNumber = FMC_
SDRAM_COLUMN_BITS_NUM_8;
00156     sdramHandle.Init.RowBitsNumber   = FMC_
SDRAM_ROW_BITS_NUM_12;
00157     sdramHandle.Init.MemoryDataWidth = SDRA
M_MEMORY_WIDTH;
00158     sdramHandle.Init.InternalBankNumber = FMC_
SDRAM_INTERN_BANKS_NUM_4;
00159     sdramHandle.Init.CASLatency      = FMC_
SDRAM_CAS_LATENCY_2;
00160     sdramHandle.Init.WriteProtection = FMC_
SDRAM_WRITE_PROTECTION_DISABLE;
00161     sdramHandle.Init.SDClockPeriod    = SDCL
OCK_PERIOD;
00162     sdramHandle.Init.ReadBurst        = FMC_
SDRAM_RBURST_ENABLE;
00163     sdramHandle.Init.ReadPipeDelay   = FMC_
SDRAM_RPIPE_DELAY_0;
00164
00165 /* SDRAM controller initialization */
00166
00167     BSP_SDRAM_MspInit(&sdramHandle, NULL); /*  

__weak function can be rewritten by the application */
00168
00169     if(HAL_SDRAM_Init(&sdramHandle, &Timing) != HAL_OK)

```

```
00170     {
00171         sdramstatus = SDRAM_ERROR;
00172     }
00173     else
00174     {
00175         sdramstatus = SDRAM_OK;
00176     }
00177
00178     /* SDRAM initialization sequence */
00179     BSP_SDRAM_Initialization_sequence(REFRESH_
COUNT);
00180
00181     return sdramstatus;
00182 }
00183
00184 /**
00185 * @brief DeInitializes the SDRAM device.
00186 * @retval SDRAM status
00187 */
00188 uint8_t BSP_SDRAM_DeInit(void)
00189 {
00190     static uint8_t sdramstatus = SDRAM_ERROR;
00191     /* SDRAM device de-initialization */
00192     sdramHandle.Instance = FMC_SDRAM_DEVICE;
00193
00194     if(HAL_SDRAM_DeInit(&sdramHandle) != HAL_O
K)
00195     {
00196         sdramstatus = SDRAM_ERROR;
00197     }
00198     else
00199     {
00200         sdramstatus = SDRAM_OK;
00201     }
00202
00203     /* SDRAM controller de-initialization */
00204     BSP_SDRAM_MspDeInit(&sdramHandle, NULL);
```

```
00205
00206     return sdramstatus;
00207 }
00208 /**
00209  * @brief Programs the SDRAM device.
00210  * @param RefreshCount: SDRAM refresh counter value
00211  * @retval None
00212  */
00214 void BSP_SDRAM_Initialization_sequence(uint3
2_t RefreshCount)
00215 {
00216     __IO uint32_t tmpmrd = 0;
00217
00218     /* Step 1: Configure a clock configuration enable command */
00219     Command.CommandMode          = FMC_SDRAM
_CMD_CLK_ENABLE;
00220     Command.CommandTarget       = FMC_SDRAM
_CMD_TARGET_BANK1;
00221     Command.AutoRefreshNumber   = 1;
00222     Command.ModeRegisterDefinition = 0;
00223
00224     /* Send the command */
00225     HAL_SDRAM_SendCommand(&sdramHandle, &Comma
nd, SDRAM_TIMEOUT);
00226
00227     /* Step 2: Insert 100 us minimum delay */
00228     /* Inserted delay is equal to 1 ms due to
systick time base unit (ms) */
00229     HAL_Delay(1);
00230
00231     /* Step 3: Configure a PALL (precharge all
) command */
00232     Command.CommandMode          = FMC_SDRAM
_CMD_PALL;
```

```

00233     Command.CommandTarget          = FMC_SDRAM
00234     _CMD_TARGET_BANK1;
00235     Command.AutoRefreshNumber      = 1;
00236     Command.ModeRegisterDefinition = 0;
00237
00238     /* Send the command */
00239     HAL_SDRAM_SendCommand(&sdramHandle, &Command,
00240                           SDRAM_TIMEOUT);
00241
00242     /* Step 4: Configure an Auto Refresh command */
00243     Command.CommandMode              = FMC_SDRAM
00244     _CMD_AUTOREFRESH_MODE;
00245     Command.CommandTarget          = FMC_SDRAM
00246     _CMD_TARGET_BANK1;
00247     Command.AutoRefreshNumber      = 8;
00248     Command.ModeRegisterDefinition = 0;
00249
00250     /* Step 5: Program the external memory mode register */
00251     tmpmrd = (uint32_t)SDRAM_MODEREG_BURST_LEN
00252             | \
00253             SDRAM_MODEREG_BURST_TYP;
00254             E_SEQUENTIAL | \
00255             SDRAM_MODEREG_CAS_LATEN;
00256             CY_2 | \
00257             SDRAM_MODEREG_OPERATING;
00258             _MODE_STANDARD | \
00259             SDRAM_MODEREG_WRITEBURS;
00260             T_MODE_SINGLE;
00261
00262     Command.CommandMode              = FMC_SDRAM
00263     _CMD_LOAD_MODE;

```

```

00257     Command.CommandTarget          = FMC_SDRAM
00258     _CMD_TARGET_BANK1;
00259     Command.AutoRefreshNumber      = 1;
00260     Command.ModeRegisterDefinition = tmpmrdr;
00261 
00262     /* Send the command */
00263     HAL_SDRAM_SendCommand(&sdramHandle, &Command,
00264                           SDRAM_TIMEOUT);
00265 
00266     /* Step 6: Set the refresh rate counter */
00267     /* Set the device refresh rate */
00268     HAL_SDRAM_ProgramRefreshRate(&sdramHandle,
00269                                   RefreshCount);
00270 }
00271 /**
00272  * @brief Reads an amount of data from the
00273  * SDRAM memory in polling mode.
00274  * @param uwStartAddress: Read start address
00275  * @param pData: Pointer to data to be read
00276  * @param uwDataSize: Size of read data from the memory
00277  * @retval SDRAM status
00278 */
00279 uint8_t BSP_SDRAM_ReadData(uint32_t uwStartAddress,
00280                            uint32_t *pData, uint32_t uwDataSize)
00281 {
00282     if(HAL_SDRAM_Read_32b(&sdramHandle, (uint32_t *)uwStartAddress, pData, uwDataSize) != HAL_OK)
00283     {
00284         return SDRAM_ERROR;
00285     }
00286     else
00287     {

```

```
00284     return SDRAM_OK;
00285 }
00286 }
00287
00288 /**
00289  * @brief Reads an amount of data from the
00290  * SDRAM memory in DMA mode.
00291  * @param uwStartAddress: Read start address
00292  * @param pData: Pointer to data to be read
00293  * @param uwDataSize: Size of read data from the memory
00294  * @retval SDRAM status
00295 */
00296 uint8_t BSP_SDRAM_ReadData_DMA(uint32_t uwStartAddress, uint32_t *pData, uint32_t uwDataSize)
00297 {
00298     if(HAL_SDRAM_Read_DMA(&sdramHandle, (uint32_t *)uwStartAddress, pData, uwDataSize) != HAL_OK)
00299     {
00300         return SDRAM_ERROR;
00301     }
00302     else
00303     {
00304         return SDRAM_OK;
00305     }
00306
00307 /**
00308  * @brief Writes an amount of data to the
00309  * SDRAM memory in polling mode.
00310  * @param uwStartAddress: Write start address
00311  * @param pData: Pointer to data to be written
```

```
00311     * @param uwDataSize: Size of written data
00312         from the memory
00313     */
00314 uint8_t BSP_SDRAM_WriteData(uint32_t uwStart
Address, uint32_t *pData, uint32_t uwDataSize)
00315 {
00316     if(HAL_SDRAM_Write_32b(&sdramHandle, (uint
32_t *)uwStartAddress, pData, uwDataSize) != HAL_0
K)
00317     {
00318         return SDRAM_ERROR;
00319     }
00320     else
00321     {
00322         return SDRAM_OK;
00323     }
00324 }
00325
00326 /**
00327     * @brief Writes an amount of data to the
SDRAM memory in DMA mode.
00328     * @param uwStartAddress: Write start addr
ess
00329     * @param pData: Pointer to data to be wri
tten
00330     * @param uwDataSize: Size of written data
from the memory
00331     * @retval SDRAM status
00332 */
00333 uint8_t BSP_SDRAM_WriteData_DMA(uint32_t uwS
tartAddress, uint32_t *pData, uint32_t uwDataSize)

00334 {
00335     if(HAL_SDRAM_Write_DMA(&sdramHandle, (uint
32_t *)uwStartAddress, pData, uwDataSize) != HAL_0
K)
```

```
00336     {
00337         return SDRAM_ERROR;
00338     }
00339     else
00340     {
00341         return SDRAM_OK;
00342     }
00343 }
00344
00345 /**
00346     * @brief Sends command to the SDRAM bank.
00347     * @param SdramCmd: Pointer to SDRAM command structure
00348     * @retval SDRAM status
00349 */
00350 uint8_t BSP_SDRAM_Sendcmd(FMC_SDRAM_CommandTypeDef *SdramCmd)
00351 {
00352     if(HAL_SDRAM_SendCommand(&sdramHandle, SdramCmd, SDRAM_TIMEOUT) != HAL_OK)
00353     {
00354         return SDRAM_ERROR;
00355     }
00356     else
00357     {
00358         return SDRAM_OK;
00359     }
00360 }
00361
00362 /**
00363     * @brief Initializes SDRAM MSP.
00364     * @param hsdram: SDRAM handle
00365     * @param Params
00366     * @retval None
00367 */
00368 __weak void BSP_SDRAM_MspInit(SDRAM_HandleTypeDefTypeDef *hsdram, void *Params)
```

```
00369 {  
00370     static DMA_HandleTypeDef dma_handle;  
00371     GPIO_InitTypeDef gpio_init_structure;  
00372  
00373     /* Enable FMC clock */  
00374     __HAL_RCC_FMC_CLK_ENABLE();  
00375  
00376     /* Enable chosen DMAx clock */  
00377     __DMax_CLK_ENABLE();  
00378  
00379     /* Enable GPIOs clock */  
00380     __HAL_RCC_GPIOC_CLK_ENABLE();  
00381     __HAL_RCC_GPIOD_CLK_ENABLE();  
00382     __HAL_RCC_GPIOE_CLK_ENABLE();  
00383     __HAL_RCC_GPIOF_CLK_ENABLE();  
00384     __HAL_RCC_GPIOG_CLK_ENABLE();  
00385     __HAL_RCC_GPIOH_CLK_ENABLE();  
00386  
00387     /* Common GPIO configuration */  
00388     gpio_init_structure.Mode      = GPIO_MODE_  
AF_PP;  
00389     gpio_init_structure.Pull      = GPIO_PULLU  
P;  
00390     gpio_init_structure.Speed    = GPIO_SPEED  
_FAST;  
00391     gpio_init_structure.Alternate = GPIO_AF12_  
FMC;  
00392  
00393     /* GPIOC configuration */  
00394     gpio_init_structure.Pin      = GPIO_PIN_3;  
00395     HAL_GPIO_Init(GPIOC, &gpio_init_structure)  
;  
00396  
00397     /* GPIOD configuration */  
00398     gpio_init_structure.Pin      = GPIO_PIN_0 | G  
PIO_PIN_1 | GPIO_PIN_8 | GPIO_PIN_9 |  
00399                                         GPIO_PIN_10 |
```

```
GPIO_PIN_14 | GPIO_PIN_15;
00400     HAL_GPIO_Init(GPIOD, &gpio_init_structure)
;
00401
00402     /* GPIOE configuration */
00403     gpio_init_structure.Pin    = GPIO_PIN_0 | G
PIO_PIN_1 | GPIO_PIN_7| GPIO_PIN_8 | GPIO_PIN_9 | \
00404                                     GPIO_PIN_10 |
GPIO_PIN_11 | GPIO_PIN_12 | GPIO_PIN_13 | GPIO_PIN
_14 | \
00405                                     GPIO_PIN_15;
00406     HAL_GPIO_Init(GPIOE, &gpio_init_structure)
;
00407
00408     /* GPIOF configuration */
00409     gpio_init_structure.Pin    = GPIO_PIN_0 | G
PIO_PIN_1 | GPIO_PIN_2| GPIO_PIN_3 | GPIO_PIN_4 | \
00410                                     GPIO_PIN_5 | G
PIO_PIN_11 | GPIO_PIN_12 | GPIO_PIN_13 | GPIO_PIN
_14 | \
00411                                     GPIO_PIN_15;
00412     HAL_GPIO_Init(GPIOF, &gpio_init_structure)
;
00413
00414     /* GPIOG configuration */
00415     gpio_init_structure.Pin    = GPIO_PIN_0 | G
PIO_PIN_1 | GPIO_PIN_4| GPIO_PIN_5 | GPIO_PIN_8 | \
00416                                     GPIO_PIN_15;
00417     HAL_GPIO_Init(GPIOG, &gpio_init_structure)
;
00418
00419     /* GPIOH configuration */
00420     gpio_init_structure.Pin    = GPIO_PIN_3 | G
PIO_PIN_5;
00421     HAL_GPIO_Init(GPIOH, &gpio_init_structure)
;
00422
```

```
00423 /* Configure common DMA parameters */
00424     dma_handle.Init.Channel          = SDRA
M_DMAMAX_CHANNEL;
00425     dma_handle.Init.Direction      = DMA_
MEMORY_TO_MEMORY;
00426     dma_handle.Init.PeriphInc      = DMA_
PINC_ENABLE;
00427     dma_handle.Init.MemInc        = DMA_
MINC_ENABLE;
00428     dma_handle.InitPeriphDataAlignment = DMA_
PDATAALIGN_WORD;
00429     dma_handle.Init.MemDataAlignment = DMA_
MDATAALIGN_WORD;
00430     dma_handle.Init.Mode          = DMA_
NORMAL;
00431     dma_handle.Init.Priority      = DMA_
PRIORITY_HIGH;
00432     dma_handle.Init.FIFOMode      = DMA_
FIFOMODE_DISABLE;
00433     dma_handle.Init.FIFOThreshold = DMA_
FIFO_THRESHOLD_FULL;
00434     dma_handle.Init.MemBurst      = DMA_
MBURST_SINGLE;
00435     dma_handle.InitPeriphBurst    = DMA_
PBURST_SINGLE;
00436
00437     dma_handle.Instance = SDRAM_DMAMAX_STREAM;
00438
00439 /* Associate the DMA handle */
00440     __HAL_LINKDMA(hsdram, hdma, dma_handle);
00441
00442 /* Deinitialize the stream for new transfe
r */
00443     HAL_DMA_DeInit(&dma_handle);
00444
00445 /* Configure the DMA stream */
00446     HAL_DMA_Init(&dma_handle);
```

```
00447
00448     /* NVIC configuration for DMA transfer complete interrupt */
00449     HAL_NVIC_SetPriority(SDRAM_DMAx_IRQn, 0x0F
00450     , 0);
00450     HAL_NVIC_EnableIRQ(SDRAM_DMAx_IRQn);
00451 }
00452
00453 /**
00454     * @brief DeInitializes SDRAM MSP.
00455     * @param hsdram: SDRAM handle
00456     * @param Params
00457     * @retval None
00458 */
00459 __weak void BSP_SDRAM_MspDeInit(SDRAM_HandleTypeDef
TypeDef *hsdram, void *Params)
00460 {
00461     static DMA_HandleTypeDef dma_handle;
00462
00463     /* Disable NVIC configuration for DMA interrupt */
00464     HAL_NVIC_DisableIRQ(SDRAM_DMAx_IRQn);
00465
00466     /* Deinitialize the stream for new transfer */
00467     dma_handle.Instance = SDRAM_DMAx_STREAM;
00468     HAL_DMA_DeInit(&dma_handle);
00469
00470     /* GPIO pins clock, FMC clock and DMA clock can be shut down in the applications
00471      by surcharging this __weak function */
00472 }
00473
00474 /**
00475     * @}
00476 */
```

```
00477
00478 /**
00479     * @}
00480     */
00481
00482 /**
00483     * @}
00484     */
00485
00486 /**
00487     * @}
00488     */
00489
00490 /***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories	Modules			
BSP				

Modules

STM32746G_DISCOVERY

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files
Directories			
BSP	Modules		

STM32746G_DISCOVERY

Modules

[STM32746G_DISCOVERY_LOW_LEVEL](#)

[STM32746G_DISCOVERY_AUDIO](#)

This file includes the low layer driver for wm8994 Audio Codec available on STM32746G-Discovery board(MB1191).

[STM32746G-Discovery_QSPI](#)

[STM32746G_DISCOVERY_SD](#)

[STM32746G_DISCOVERY_SDRAM](#)

[STM32746G_DISCOVERY_TS](#)

[STM32746G_DISCOVERY_CAMERA](#)

[STM32746G_DISCOVERY_EEPROM](#)

This file includes the I2C EEPROM driver of STM32746G-Discovery board.

[STM32746G_DISCOVERY_LCD](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery

BSP User Manual by **[doxygen](#)** 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
STM32746G_DISCOVERY_LOW_LEVEL Exported Constants				Modules
STM32746G_DISCOVERY_LOW_LEVEL				

Modules

STM32746G_DISCOVERY_LOW_LEVEL_LED

Define for STM32746G_Discovery board.

STM32746G_DISCOVERY_LOW_LEVEL_BUTTON

STM32746G_DISCOVERY_LOW_LEVEL_SIGNAL

STM32746G_DISCOVERY_LOW_LEVEL_COM

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [doxygen](#) 1.7.6.1

STM32746G-Discovery BSP User Manual

Main Page	Modules	Data Structures	Files	
Directories				
STM32746G-Discovery QSPI				Modules
STM32746G_DISCOVERY				

Modules

[**STM32746G_DISCOVERY_QSPI Private Variables**](#)
[**STM32746G_DISCOVERY_QSPI Private Functions**](#)
[**STM32746G_DISCOVERY_QSPI Exported Functions**](#)
[**STM32746G_DISCOVERY_QSPI Exported Constants**](#)
[**STM32746G_DISCOVERY_QSPI Exported Types**](#)

Generated on Fri Dec 30 2016 16:31:33 for STM32746G-Discovery
BSP User Manual by [**doxygen**](#) 1.7.6.1