10 General purpose I/O (GPIO)

The GPIO driver can be used for several purposes, including pin configuration, single bit set/reset, lock mechanism, reading from a port pin, and writing data into a port pin.

Section 10.1: GPIO register structure describes the data structures used in the GPIO Firmware Library. *Section 10.2: Firmware library functions* presents the Firmware Library functions.

10.1 GPIO register structure

The GPIO register structure, GPIO_*TypeDef,* is defined in the *stm32f10x_map.h* file as follows:

```
typedef struct
{
  vu32 CRL;
  vu32 CRH;
  vu32 IDR;
  vu32 ODR;
  vu32 BSRR;
  vu32 BRR;
  vu32 LCKR;
} GPIO_TypeDef;

typedef struct
{
  vu32 EVCR;
  vu32 MAPR;
  vu32 EXTICR[4];
} AFIO_TypeDef;
```

Table 178 gives the list of the GPIO registers:

Table 178. GPIO registers

Register	Description
CRL	Port Control Register low
CRH	Port Control Register High
IDR	Input Data Register
ODR	Output Data Register
BSRR	Bit Set Reset Register
BRR	Bit Reset Register
LCKR	Lock Register
EVCR	Event Control Register
MAPR	Remap Debug and AF Register
EXTICR	EXTI Line 0 to Line 15 Configuration Register

The five GPIO peripherals are declared in *stm32f10x_map*.h:

```
...
#define PERIPH_BASE           ((u32)0x40000000)
#define APB1PERIPH_BASE       PERIPH_BASE
#define APB2PERIPH_BASE       (PERIPH_BASE + 0x10000)
#define AHBPERIPH_BASE        (PERIPH_BASE + 0x20000)
...
#define AFIO_BASE             (APB2PERIPH_BASE + 0x0000)
#define GPIOA_BASE            (APB2PERIPH_BASE + 0x0800)
#define GPIOB_BASE            (APB2PERIPH_BASE + 0x0C00)
#define GPIOC_BASE            (APB2PERIPH_BASE + 0x1000)
#define GPIOD_BASE            (APB2PERIPH_BASE + 0x1400)
#define GPIOE_BASE            (APB2PERIPH_BASE + 0x1800)

#ifndef DEBUG
...
#ifdef _AFIO
  #define AFIO                  ((AFIO_TypeDef *) AFIO_BASE)
#endif /*_AFIO */

#ifdef _GPIOA
  #define GPIOA                 ((GPIO_TypeDef *) GPIOA_BASE)
#endif /*_GPIOA */

#ifdef _GPIOB
  #define GPIOB                 ((GPIO_TypeDef *) GPIOB_BASE)
#endif /*_GPIOB */

#ifdef _GPIOC
  #define GPIOC                 ((GPIO_TypeDef *) GPIOC_BASE)
#endif /*_GPIOC */

#ifdef _GPIOD
  #define GPIOD                 ((GPIO_TypeDef *) GPIOD_BASE)
#endif /*_GPIOD */

#ifdef _GPIOE
  #define GPIOE                 ((GPIO_TypeDef *) GPIOE_BASE)
#endif /*_GPIOE */
...
#else   /* DEBUG */
...
#ifdef _AFIO
  EXT AFIO_TypeDef            *AFIO;
#endif /*_AFIO */

#ifdef _GPIOA
  EXT GPIO_TypeDef            *GPIOA;
#endif /*_GPIOA */

#ifdef _GPIOB
  EXT GPIO_TypeDef            *GPIOB;
```

```
#endif /*_GPIOB */

#ifdef _GPIOC
  EXT GPIO_TypeDef            *GPIOC;
#endif /*_GPIOC */

#ifdef _GPIOD
  EXT GPIO_TypeDef            *GPIOD;
#endif /*_GPIOD */

#ifdef _GPIOE
  EXT GPIO_TypeDef            *GPIOE;
#endif /*_GPIOE */
...
#endif
```

When using the Debug mode, _AFIO, _GPIOA, _GPIOB, _GPIOC, _GPIOD and _GPIOE pointers are initialized in *stm32f10x_lib.c* file:

```
#ifdef _GPIOA
  GPIOA = (GPIO_TypeDef *)  GPIOA_BASE;
#endif /*_GPIOA */

#ifdef _GPIOB
  GPIOB = (GPIO_TypeDef *)  GPIOB_BASE;
#endif /*_GPIOB */

#ifdef _GPIOC
  GPIOC = (GPIO_TypeDef *)  GPIOC_BASE;
#endif /*_GPIOC */

#ifdef _GPIOD
  GPIOD = (GPIO_TypeDef *)  GPIOD_BASE;
#endif /*_GPIOD */

#ifdef _GPIOE
  GPIOE = (GPIO_TypeDef *)  GPIOE_BASE;
#endif /*_GPIOE */

#ifdef _AFIO
  AFIO = (AFIO_TypeDef *)  AFIO_BASE;
#endif /*_AFIO */
```

To access the GPIO registers, _GPIO, _AFIO, _GPIOA, _GPIOB, _GPIOC, _GPIOD and _GPIOE must be defined in stm32f10x_conf.h:

```
#define _GPIO
#define _GPIOA
#define _GPIOB
#define _GPIOC
#define _GPIOD
#define _GPIOE
#define _AFIO
```

10.2 Firmware library functions

Table 179 gives the list of the GPIO firmware library functions.

Table 179. GPIO firmware library functions

Function name	Description
GPIO_DeInit	Resets the GPIOx peripheral registers to their default reset values.
GPIO_AFIODeInit	Resets the Alternate Functions (remap, event control and EXTI configuration) registers to their default reset values.
GPIO_Init	Initializes the GPIOx peripheral according to the specified parameters in the GPIO_InitStruct.
GPIO_StructInit	Fills each GPIO_InitStruct member with its default value.
GPIO_ReadInputDataBit	Reads the specified input port pin
GPIO_ReadInputData	Reads the specified GPIO input data port
GPIO_ReadOutputDataBit	Reads the specified output data port bit
GPIO_ReadOutputData	Reads the specified GPIO output data port
GPIO_SetBits	Sets the selected data port bits
GPIO_ResetBits	Clears the selected data port bits
GPIO_WriteBit	Sets or clears the selected data port bit
GPIO_Write	Writes data to the specified GPIO data port
GPIO_PinLockConfig	Locks GPIO Pins configuration registers
GPIO_EventOutputConfig	Selects the GPIO pin used as Event output.
GPIO_EventOutputCmd	Enables or disables the Event Output.
GPIO_PinRemapConfig	Changes the mapping of the specified pin.
GPIO_EXTILineConfig	Selects the GPIO pin used as EXTI Line.

10.2.1 GPIO_DeInit function

Table 180 describes the GPIO_DeInit function.

Table 180. GPIO_DeInit function

Function name	GPIO_DeInit
Function prototype	void GPIO_DeInit(GPIO_TypeDef* GPIOx)
Behavior description	Resets the GPIOx peripheral registers to their default reset values.
Input parameter	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	RCC_APB2PeriphResetCmd()

Example:

```
/* Resets the GPIOA peripheral registers to their default reset
values */
GPIO_DeInit(GPIOA);
```

10.2.2 GPIO_AFIODeInit function

Table 181 describes the GPIO_AFIODeInit function.

Table 181. GPIO_AFIODeInit function

Function name	GPIO_AFIODeInit
Function prototype	void GPIO_AFIODeInit(void)
Behavior description	Resets the Alternate functions registers (remap, event control and EXTI configuration) to their default reset values.
Input parameter	None
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	RCC_APB2PeriphResetCmd()

Example:

```
/* Resets the Alternate functions registers to their default reset
values */
GPIO_AFIODeInit();
```

10.2.3 GPIO_Init function

Table 182 describes the GPIO_Init function.

Table 182. GPIO_Init function

Function name	GPIO_Init
Function prototype	void GPIO_Init(GPIO_TypeDef* GPIOx, GPIO_InitTypeDef* GPIO_InitStruct)
Behavior description	Initializes the GPIOx peripheral according to the specified parameters in the GPIO_InitStruct.
Input parameter1	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Input parameter2	GPIO_InitStruct: pointer to a GPIO_InitTypeDef structure that contains the configuration information for the specified GPIO peripheral. Refer to *Section : GPIO_InitTypeDef structure* for more details on the allowed values of this parameter.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

GPIO_InitTypeDef structure

The GPIO_InitTypeDef structure is defined in the *stm32f10x_gpio.h* file:

```
typedef struct
{
  u16 GPIO_Pin;
  GPIOSpeed_TypeDef GPIO_Speed;
  GPIOMode_TypeDef GPIO_Mode;
} GPIO_InitTypeDef;
```

GPIO_Pin

This member selects the GPIO pins to configure. Multiple-pin configuration can be performed by using the '|' operator. Any combination of the following values can be used:

Table 183. GPIO_Pin values

GPIO_Pin	Description
GPIO_Pin_None	No pin selected
GPIO_Pin_0	Pin 0 Selected
GPIO_Pin_1	Pin 1 Selected
GPIO_Pin_2	Pin 2 Selected
GPIO_Pin_3	Pin 3 Selected
GPIO_Pin_4	Pin 4 Selected
GPIO_Pin_5	Pin 5 Selected
GPIO_Pin_6	Pin 6 Selected
GPIO_Pin_7	Pin 7 Selected
GPIO_Pin_8	Pin 8 Selected
GPIO_Pin_9	Pin 9 Selected
GPIO_Pin_10	Pin 10 Selected
GPIO_Pin_11	Pin 11 Selected
GPIO_Pin_12	Pin 12 Selected
GPIO_Pin_13	Pin 13 Selected
GPIO_Pin_14	Pin 14 Selected
GPIO_Pin_15	Pin 15 Selected
GPIO_Pin_All	All Pins Selected

GPIO_Speed

GPIO_Speed is used to configure the speed for the selected pins.See *Table 184* for the values taken by this member.

Table 184. GPIO_Speed values

GPIO_Speed	Description
GPIO_Speed_10MHz	Output Maximum Frequency = 10 MHz
GPIO_Speed_2MHz	Output Maximum Frequency = 2 MHz
GPIO_Speed_50MHz	Output Maximum Frequency = 50 MHz

GPIO_Mode

GPIO_Mode configures the operating mode for the selected pins.See *Table 185* for the values taken by this member.

Table 185. GPIO_Mode values

GPIO_Mode	Description
GPIO_Mode_AIN	Analog Input
GPIO_Mode_IN_FLOATING	Input Floating
GPIO_Mode_IPD	Input Pull-Down
GPIO_Mode_IPU	Input Pull-up
GPIO_Mode_Out_OD	Open Drain Output
GPIO_Mode_Out_PP	Push-Pull Output
GPIO_Mode_AF_OD	Open Drain Output Alternate-Function
GPIO_Mode_AF_PP	Push-Pull Output Alternate-Function

Note: 1 When a pin is configured in input pull-up or pull-down mode, the Px_BSRR and Px_BRR registers are used.

2 GPIO_Mode allows to configure both the GPIO direction (Input/Output) and the corresponding input/output configuration: bits[7:4] GPIO_Mode configure the GPIO direction, while bits [4:0] define the configuration. The GPIO direction have the following indexes:

- *GPIO in input mode = 0x00*
- *GPIO in output mode = 0x01*

Table 186 shows all the GPIO_Mode indexes and codes.

Table 186. GPIO_Mode indexes and codes

GPIO Direction	Index	Mode	Configuration	Mode Code
GPIO Input	0x00	GPIO_Mode_AIN	0x00	0x00
		GPIO_Mode_IN_FLOATING	0x04	0x04
		GPIO_Mode_IPD	0x08	0x28
		GPIO_Mode_IPU	0x08	0x48
GPIO Output	0x01	GPIO_Mode_Out_OD	0x04	0x14
		GPIO_Mode_Out_PP	0x00	0x10
		GPIO_Mode_AF_OD	0x0C	0x1C
		GPIO_Mode_AF_PP	0x08	0x18

Example:

```
/* Configure all the GPIOA in Input Floating mode */
GPIO_InitTypeDef GPIO_InitStructure;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_All;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_10MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

10.2.4 GPIO_StructInit function

Table 187 describes the GPIO_StructInit function.

Table 187. GPIO_StructInit function

Function name	GPIO_StructInit
Function prototype	void GPIO_StructInit(GPIO_InitTypeDef* GPIO_InitStruct)
Behavior description	Fills each GPIO_InitStruct member with its default value.
Input parameter	GPIO_InitStruct: pointer to a GPIO_InitTypeDef structure which will be initialized.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

The GPIO_InitStruct default values are given in *Table 188*.

Table 188. GPIO_InitStruct default values

Member	Default value
GPIO_Pin	GPIO_Pin_All
GPIO_Speed	GPIO_Speed_2MHz
GPIO_Mode	GPIO_Mode_IN_FLOATING

Example:

```
/* Initialize the GPIO Init Structure parameters */
GPIO_InitTypeDef GPIO_InitStructure;
GPIO_StructInit(&GPIO_InitStructure);
```

10.2.5 GPIO_ReadInputDataBit function

Table 189 describes the GPIO_ReadInputDataBit function.

Table 189. GPIO_ReadInputDataBit function

Function name	GPIO_ReadInputDataBit
Function prototype	u8 GPIO_ReadInputDataBit(GPIO_TypeDef* GPIOx, u16 GPIO_Pin)
Behavior description	Reads the specified input port pin.
Input parameter1	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Input parameter2	GPIO_Pin: port bit to be read. Refer to *Section : GPIO_Pin* for more details on the allowed values of this parameter.
Output parameter	None
Return parameter	The input port pin value.
Required preconditions	None
Called functions	None

Example:

```
/* Reads the seventh pin of the GPIOB and store it in ReadValue
variable */
u8 ReadValue;
ReadValue = GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_7);
```

10.2.6 GPIO_ReadInputData function

Table 190 describes the GPIO_ReadInputData function.

Table 190. GPIO_ReadInputData function

Function name	GPIO_ReadInputData
Function prototype	u16 GPIO_ReadInputData(GPIO_TypeDef* GPIOx)
Behavior description	Reads the specified GPIO input data port.
Input parameter	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Output parameter	None
Return parameter	GPIO input data port value.
Required preconditions	None
Called functions	None

Example:

```
/*Read the GPIOC input data port and store it in ReadValue
variable*/
u16 ReadValue;
ReadValue = GPIO_ReadInputData(GPIOC);
```

10.2.7 GPIO_ReadOutputDataBit function

Table 191 describes the GPIO_ReadOutputDataBit function.

Table 191. GPIO_ReadOutputDataBit function

Function name	GPIO_ReadOutputDataBit
Function prototype	u8 GPIO_ReadOutputDataBit(GPIO_TypeDef* GPIOx, u16 GPIO_Pin)
Behavior description	Reads the specified output data port bit.
Input parameter1	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Input parameter2	GPIO_Pin: port bit to read. Refer to *Section : GPIO_Pin* for more details on the allowed values of this parameter.
Output parameter	None
Return parameter	The output port pin value.
Required preconditions	None
Called functions	None

Example:

```
/* Reads the seventh pin of the GPIOB and store it in ReadValue
variable */
u8 ReadValue;
ReadValue = GPIO_ReadOutputDataBit(GPIOB, GPIO_Pin_7);
```

10.2.8 GPIO_ReadOutputData function

Table 192 describes the GPIO_ReadOutputData function.

Table 192. GPIO_ReadOutputData function

Function name	GPIO_ReadOutputData
Function prototype	u16 GPIO_ReadOutputData(GPIO_TypeDef* GPIOx)
Behavior description	Reads the specified GPIO output data port.
Input parameter	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Output parameter	None
Return parameter	GPIO output data port value.
Required preconditions	None
Called functions	None

Example:

```
/* Read the GPIOC output data port and store it in ReadValue
variable */
u16 ReadValue;
ReadValue = GPIO_ReadOutputData(GPIOC);
```

10.2.9 GPIO_SetBits

Table 192 describes the GPIO_SetBits function.

Table 193. GPIO_SetBits function

Function name	GPIO_SetBits
Function prototype	void GPIO_SetBits(GPIO_TypeDef* GPIOx, u16 GPIO_Pin)
Behavior description	Sets the selected data port bits.
Input parameter1	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Input parameter2	GPIO_Pin: specifies the port bits to be written. This parameter can be any combination of GPIO_Pin_x where x can be (0..15). Refer to *Section : GPIO_Pin* for more details on the allowed values of this parameter.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

Example:

```
/* Set the GPIOA port pin 10 and pin 15 */
GPIO_SetBits(GPIOA, GPIO_Pin_10 | GPIO_Pin_15);
```

10.2.10 GPIO_ResetBits

Table 194 describes the GPIO_ResetBits function.

Table 194. GPIO_ResetBits function

Function name	GPIO_ResetBits
Function prototype	void GPIO_ResetBits(GPIO_TypeDef* GPIOx, u16 GPIO_Pin)
Behavior description	Clears the selected data port bits.
Input parameter1	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Input parameter2	GPIO_Pin: specifies the port bits to be written. This parameter can be any combination of GPIO_Pin_x where x can be (0..15). Refer to *Section : GPIO_Pin* for more details on the allowed values of this parameter.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

Example:

```
/* Clears the GPIOA port pin 10 and pin 15 */
GPIO_ResetBits(GPIOA, GPIO_Pin_10 | GPIO_Pin_15);
```

10.2.11 GPIO_WriteBit function

Table 195 describes the GPIO_WriteBit function.

Table 195. GPIO_WriteBit function

Function name	GPIO_WriteBit
Function prototype	void GPIO_WriteBit(GPIO_TypeDef* GPIOx, u16 GPIO_Pin, BitAction BitVal)
Behavior description	Sets or clears the selected data port bit.
Input parameter1	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Input parameter2	GPIO_Pin: port bit to be written. Refer to *Section : GPIO_Pin* for more details on the allowed values of this parameter.
Input parameter3	BitVal: this parameter specifies the value to be written to the selected bit. BitVal must be one of the BitAction enum values: Bit_RESET: to clear the port pin. Bit_SET: to set the port pin.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

Example:

```
/* Set the GPIOA port pin 15 */
GPIO_WriteBit(GPIOA, GPIO_Pin_15, Bit_SET);
```

10.2.12 GPIO_Write function

Table 196 describes the GPIO_Write function.

Table 196. GPIO_Write function

Function name	GPIO_Write
Function prototype	void GPIO_Write(GPIO_TypeDef* GPIOx, u16 PortVal)
Behavior description	Writes the passed value in the selected data GPIOx port register.
Input parameter1	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Input parameter2	PortVal: the value to be written to the data port register.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

Example:

```
/* Write data to GPIOA data port */
GPIO_Write(GPIOA, 0x1101);
```

10.2.13 GPIO_PinLockConfig function

Table 197 describes the GPIO_PinLockConfig function.

Table 197. GPIO_PinLockConfig function

Function name	GPIO_PinLockConfig
Function prototype	void GPIO_PinLockConfig(GPIO_TypeDef* GPIOx, u16 GPIO_Pin)
Behavior description	Locks GPIO pins configuration registers.
Input parameter1	GPIOx: where x can be A, B, C, D or E to select the GPIO peripheral.
Input parameter2	GPIO_Pin: port bit to be written. Refer to *Section : GPIO_Pin* for more details on the allowed values of this parameter.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

Example:

```
/* Lock GPIOA Pin0 and Pin1 */
GPIO_PinLockConfig(GPIOA, GPIO_Pin_0 | GPIO_Pin_1);
```

10.2.14 GPIO_EventOutputConfig function

Table 198 describes the GPIO_EventOutputConfig function.

Table 198. GPIO_EventOutputConfig function

Function name	GPIO_EventOuputConfig
Function prototype	void GPIO_EventOutputConfig(u8 GPIO_PortSource, u8 GPIO_PinSource)
Behavior description	Selects the GPIO pin used as Event output.
Input parameter1	GPIO_PortSource: selects the GPIO port to be used as source for Event output. Refer to *Section : GPIO_PortSource* for more details on the allowed values of this parameter.
Input parameter2	GPIO_PinSource: pin for the Event output. This parameter can be GPIO_PinSourcex where x can be (0..15).
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

GPIO_PortSource

This parameter is used to select the GPIO port source used as Event output. See *Table 199* for the values taken by GPIO_PortSource.

Table 199. GPIO_PortSource values

GPIO_PortSource	Description
GPIO_PortSourceGPIOA	GPIOA Selected
GPIO_PortSourceGPIOB	GPIOB Selected
GPIO_PortSourceGPIOC	GPIOC Selected
GPIO_PortSourceGPIOD	GPIOD Selected
GPIO_PortSourceGPIOE	GPIOE Selected

Example:

```
/* Selects the GPIOE pin 5 for EVENT output */
GPIO_EventOutputConfig(GPIO_PortSourceGPIOE, GPIO_PinSource5);
```

10.2.15 GPIO_EventOutputCmd function

Table 200 describes the GPIO_EventOutputCmd function.

Table 200. GPIO_EventOutputCmd function

Function name	GPIO_EventOuputCmd
Function prototype	void GPIO_EventOutputCmd(FunctionalState NewState)
Behavior description	Enables or disables the Event Output.
Input parameter	NewState: new state of the Event output. This parameter can be: ENABLE or DISABLE.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

Example:

```
/* Enable Event Ouput to the GPIOC pin 6 */
GPIO_EventOutputConfig(GPIO_PortSourceGPIOC, GPIO_PinSource6);
GPIO_EventOutputCmd(ENABLE);
```

10.2.16 GPIO_PinRemapConfig function

Table 201 describes the GPIO_PinRemapConfig function.

Table 201. GPIO_PinRemapConfig function

Function name	GPIO_PinRemapConfig
Function prototype	void GPIO_PinRemapConfig(u32 GPIO_Remap, FunctionalState NewState)
Behavior description	Changes the mapping of the specified pin.
Input parameter1	GPIO_Remap: selects the pin to remap. Refer to *Section : GPIO_Remap* for more details on the allowed values of this parameter.
Input parameter2	NewState: new state of the port pin remapping. This parameter can be set to ENABLE or DISABLE.
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

GPIO_Remap

GPIO_Remap parameter is used to change the alternate function mapping. See *Table 202* for the values taken by this parameter.

Table 202. GPIO_Remap values

GPIO_Remap	Description
GPIO_Remap_SPI1	SPI1 Alternate Function mapping
GPIO_Remap_I2C1	I2C1 Alternate Function mapping
GPIO_Remap_USART1	USART1 Alternate Function mapping
GPIO_Remap_USART2	USART2 Alternate Function mapping
GPIO_PartialRemap_USART3	USART3 Partial Alternate Function mapping
GPIO_FullRemap_USART3	USART3 Full Alternate Function mapping
GPIO_PartialRemap_TIM1	TIM1 Partial Alternate Function mapping
GPIO_FullRemap_TIM1	TIM1 Full Alternate Function mapping
GPIO_PartialRemap1_TIM2	TIM2 Partial1 Alternate Function mapping
GPIO_PartialRemap2_TIM2	TIM2 Partial2 Alternate Function mapping
GPIO_FullRemap_TIM2	TIM2 Full Alternate Function mapping
GPIO_PartialRemap_TIM3	TIM3 Partial Alternate Function mapping
GPIO_FullRemap_TIM3	TIM3 Full Alternate Function mapping
GPIO_Remap_TIM4	TIM4 Alternate Function mapping
GPIO_Remap1_CAN	CAN Alternate Function mapping
GPIO_Remap2_CAN	CAN Alternate Function mapping
GPIO_Remap_PD01	PD01 Alternate Function mapping
GPIO_Remap_SWJ_NoJTRST	Full SWJ Enabled (JTAG-DP + SW-DP) but without JTRST
GPIO_Remap_SWJ_JTAGDisable	JTAG-DP Disabled and SW-DP Enabled
GPIO_Remap_SWJ_Disable	Full SWJ Disabled (JTAG-DP + SW-DP)

Example:

```
/* I2C1_SCL on PB.08, I2C1_SDA on PB.09 */
GPIO_PinRemapConfig(GPIO_Remap_I2C1, ENABLE);
```

10.2.17 GPIO_EXTILineConfig function

Table 203 describes the GPIO_EXTILineConfig function.

Table 203. GPIO_EXTILineConfig function

Function name	GPIO_EXTILineConfig
Function prototype	void GPIO_EXTILineConfig(u8 GPIO_PortSource, u8 GPIO_PinSource)
Behavior description	Selects the GPIO pin used as EXTI Line.
Input parameter1	GPIO_PortSource: selects the GPIO port to be used as source for EXTI lines. Refer to *Section : GPIO_PortSource* for more details on the allowed values of this parameter.
Input parameter2	GPIO_PinSource: EXTI line to be configured. This parameter can be GPIO_PinSourcex where x can be (0..15).
Output parameter	None
Return parameter	None
Required preconditions	None
Called functions	None

Example:

```
/* Selects PB.08 as EXTI Line 8 */
GPIO_EXTILineConfig(GPIO_PortSource_GPIOB, GPIO_PinSource8);
```