# STM32F10X函式庫之GPIO驅動介紹

*support.wuyang@gmail.com*

***WU-YANG***

***Technology Co., Ltd.***

# Declared Version

| | |
|---|---|
| ***Training Only*** | |
| ***Declare*** | |
| ***Document Version*** | ***1.00*** |
| ***Release Date*** | ***2009.08*** |
| ***Document Title*** | **STM32F10X函式庫之GPIO驅動介紹** |
| ***Exercise Time*** | **------** |
| ***Platform*** | ■***MIAT_STM32***<br>■ ***MIAT_IOB*** |
| ***Peripheral*** | ***Key Switch, LED*** |
| ***Author*** | ■ ***WU-YANG Technology Co., Ltd.*** |

# Outline

- GPIO Features
- GPIO Configuration Modes
- GPIO registers介紹
- GPIO標準函式庫驅動介紹

# GPIO Features

- Up to 112 multifunction bi-direction I/O ports
  - Standard I/Os 5V tolerant
  - The GPIOs can sink 25mA ( total currents suck is 150mA)
  - Configurable Output Speed up to 50 MHz
  - Up to 16/21 Analog Inputs
  - Alternate Functions pins (like USARTx, TIMx, I2Cx, SPIx, CAN, USB…)
  - Software remapping of I/O alternate functions
  - All mappable on 16 external interrupt vectors
  - One I/O can be used as Wake-Up from STANDBY (PA0)
  - Atomic Bit Set and Bit Reset using BSRR and BRR registers
  - The locking mechanism allows the IO configuration to be frozen. When the LOCK sequence has been applied on a port bit, it is no longer possible to modify the value of the port bit until the next reset.

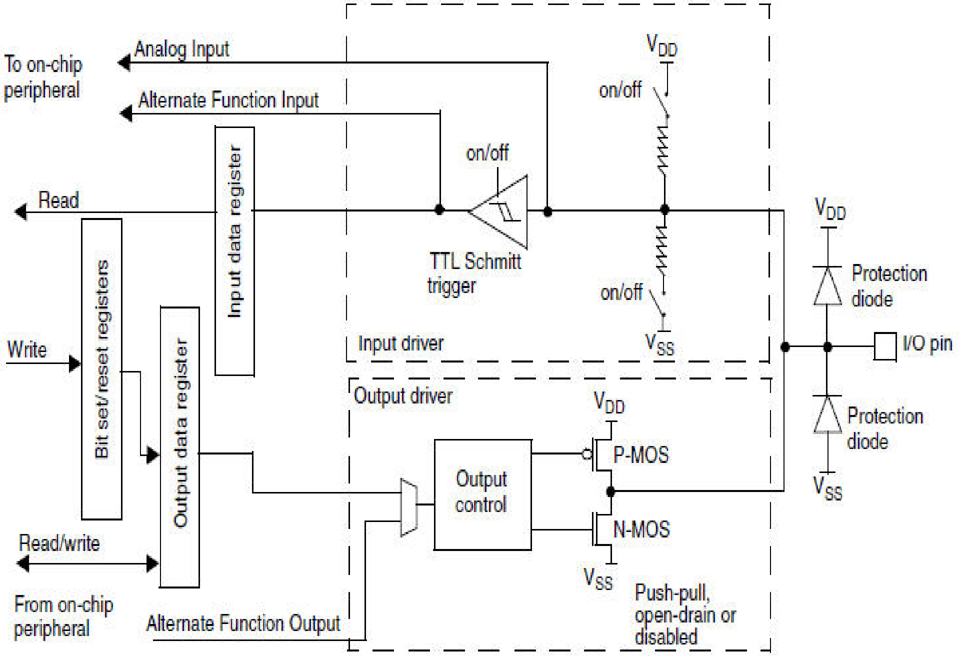
# GPIO configuration Modes

**Table 17. Port bit configuration table**

| Configuration mode | | CNF1 | CNF0 | MODE1 | MODE0 | PxODR register |
|---|---|---|---|---|---|---|
| General purpose output | Push-pull | 0 | 0 | 01<br>10<br>11<br>see *Table 18* | | 0 or 1 |
| | Open-drain | | 1 | | | 0 or 1 |
| Alternate Function output | Push-pull | 1 | 0 | | | don't care |
| | Open-drain | | 1 | | | don't care |
| Input | Analog input | 0 | 0 | 00 | | don't care |
| | Input floating | | 1 | | | don't care |
| | Input pull-down | 1 | 0 | | | 0 |
| | Input pull-up | | | | | 1 |

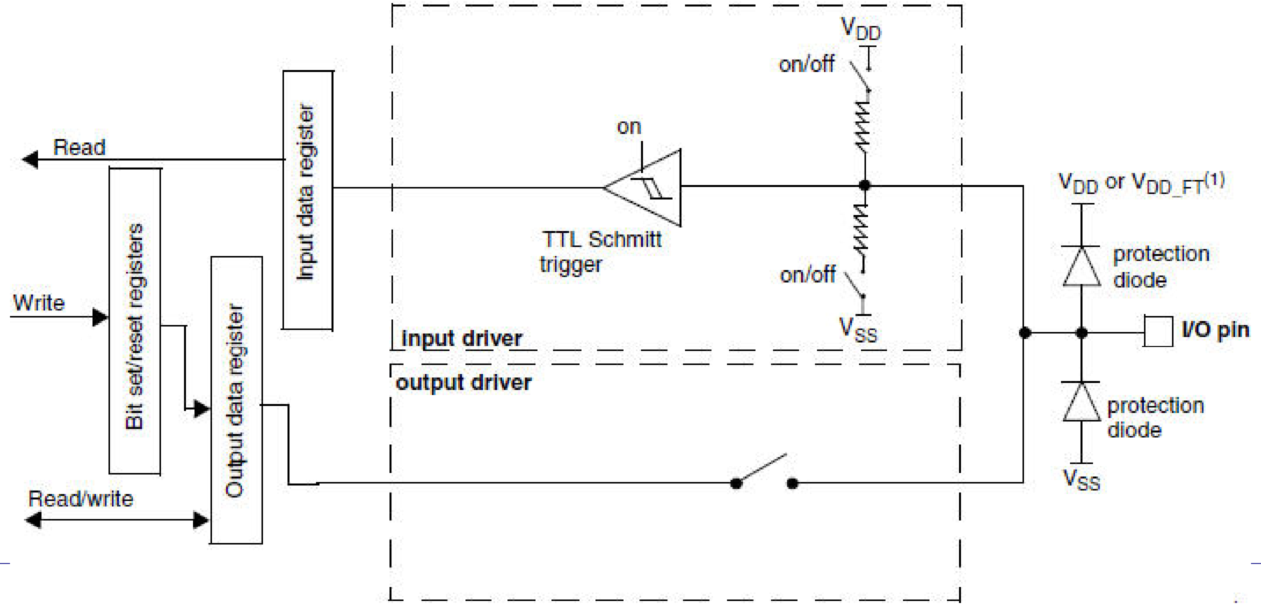
**Table 18. Output MODE bits**

| MODE[1:0] | Meaning |
|---|---|
| 00 | Reserved |
| 01 | Max. output speed 10 MHz |
| 10 | Max. output speed 2 MHz |
| 11 | Max. output speed 50 MHz |



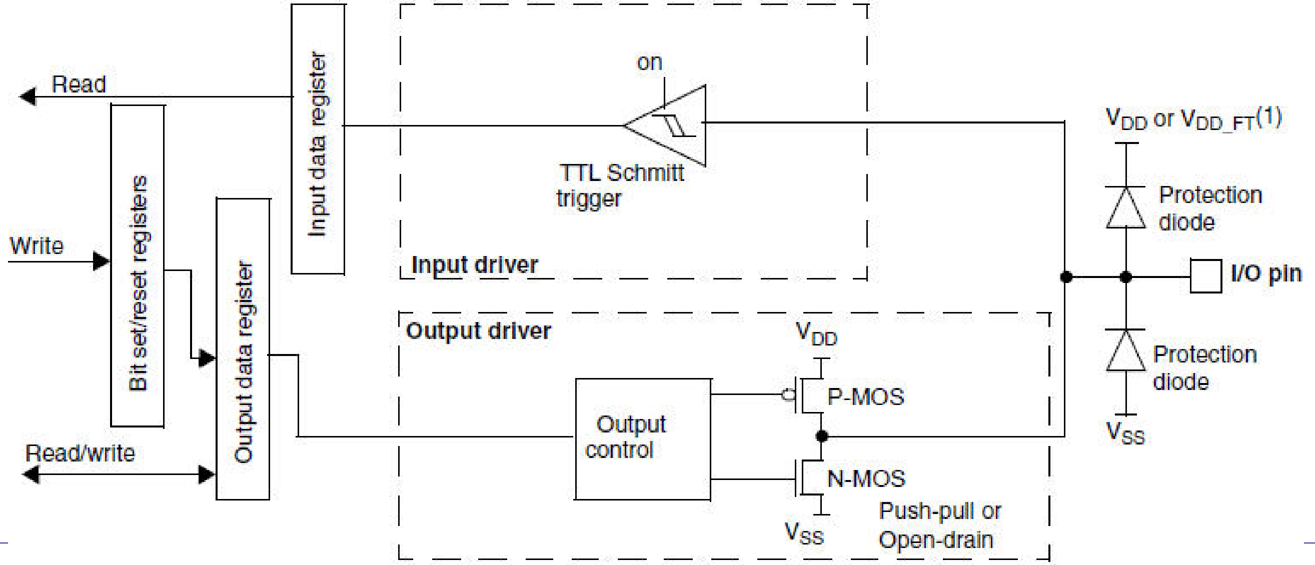
# Input configuration

- When the I/O Port is programmed as Input:
  - The Output Buffer is disabled
  - The Schmitt Trigger Input is activated
  - The weak pull-up and pull-down resistors are activated or not depending on input configuration (pull-up, pull-down or floating):
  - The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
  - A read access to the Input Data Register obtains the I/O State.



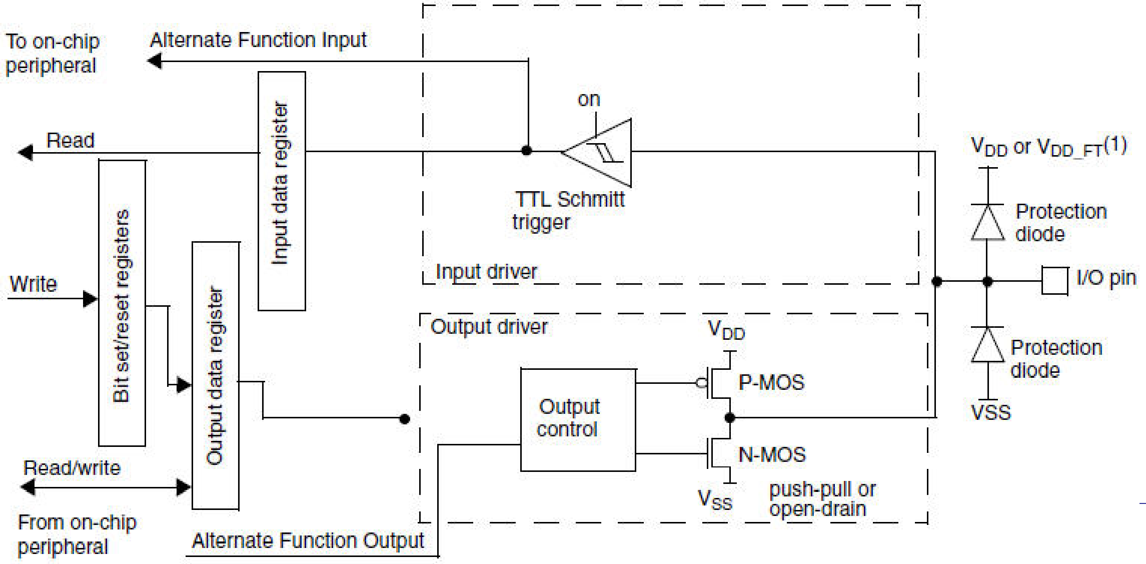
# Output configuration

- When the I/O Port is programmed as Output:
  - The Output Buffer is enabled:
    - Open Drain Mode: A “0” in the Output register activates the N-MOS while a “1” in the Output register leaves the port in Hi-Z. (the P-MOS is never activated)
    - Push-Pull Mode: A “0” in the Output register activates the N-MOS while a “1” in the Output register activates the P-MOS.
  - The Schmitt Trigger Input is activated.
  - The weak pull-up and pull-down resistors are disabled.
  - The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
  - A read access to the Input Data Register gets the I/O state in open drain mode
  - A read access to the Output Data register gets the last written value in Push-Pull mode



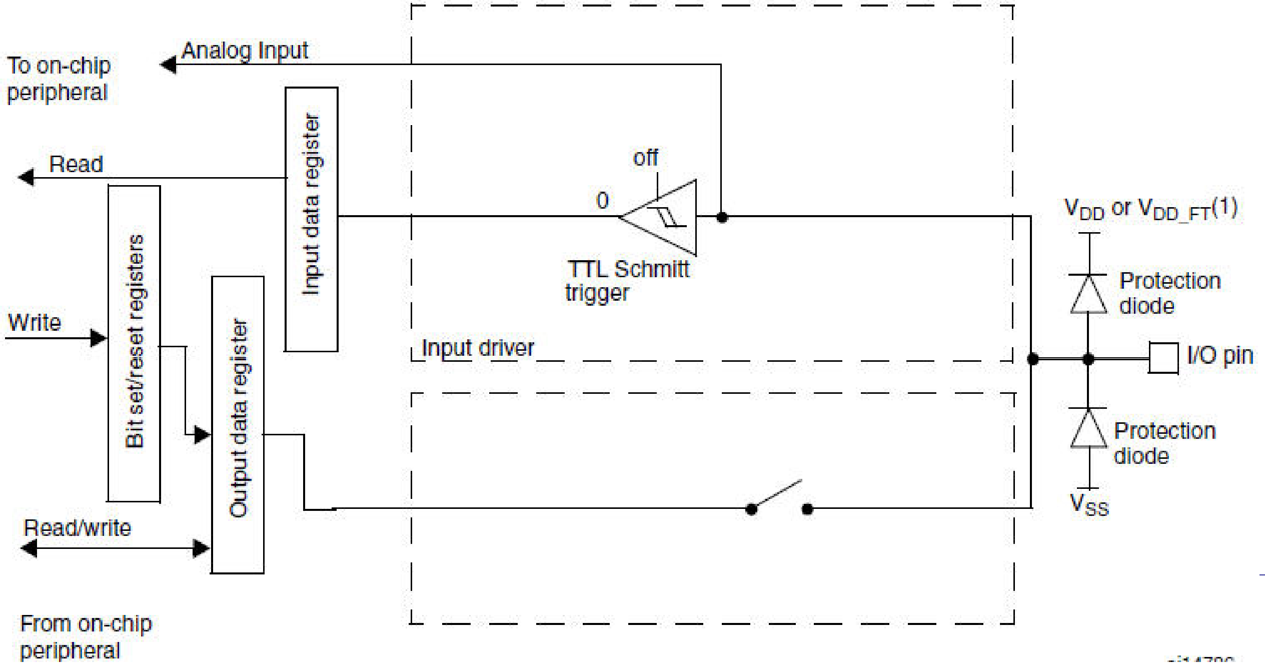
# Alternate function configuration

- When the I/O Port is programmed as Alternate Function:
  - The Output Buffer is turned on in Open Drain or Push-Pull configuration
  - The Output Buffer is driven by the signal coming from the peripheral (alternate function out)
  - The Schmitt Trigger Input is activated
  - The weak pull-up and pull-down resistors are disabled.
  - The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
  - A read access to the Input Data Register gets the I/O state in open drain mode
  - A read access to the Output Data register gets the last written value in Push-Pull mode

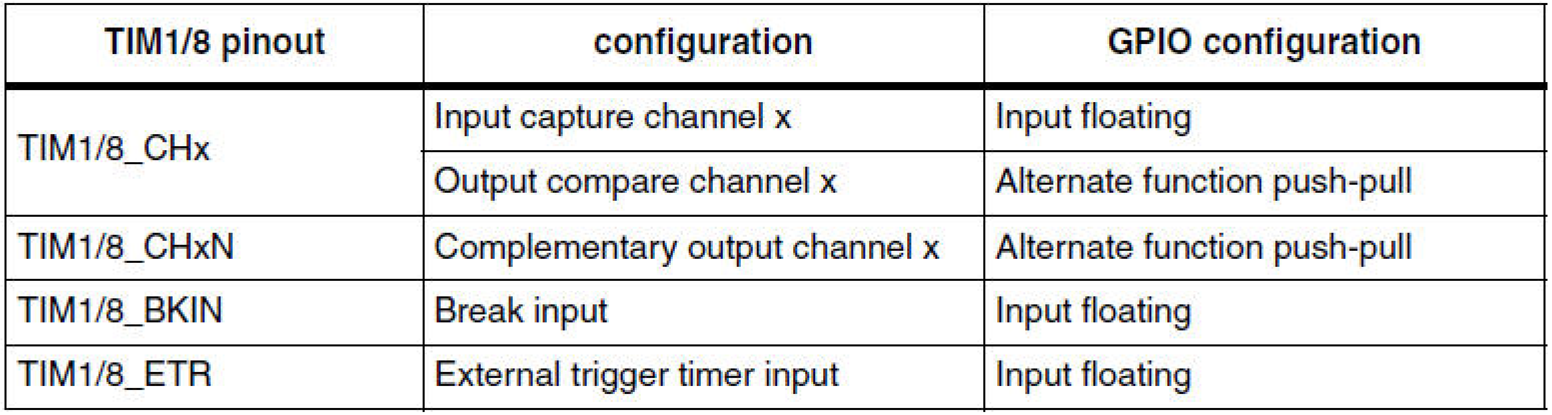


# Analog input configuration

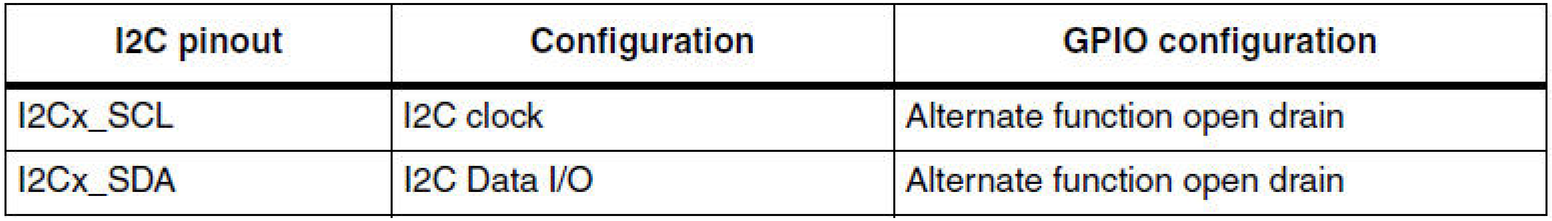
- When the I/O Port is programmed as Analog Input Configuration:
  - The Output Buffer is disabled.
  - The Schmitt Trigger Input is de-activated providing zero consumption for every analog value of the I/O pin. The output of the Schmitt Trigger is forced to a constant value (0).
  - The weak pull-up and pull-down resistors are disabled.
  - Read access to the Input Data Register gets the value “0”.



# Peripherals’ GPIO configurations



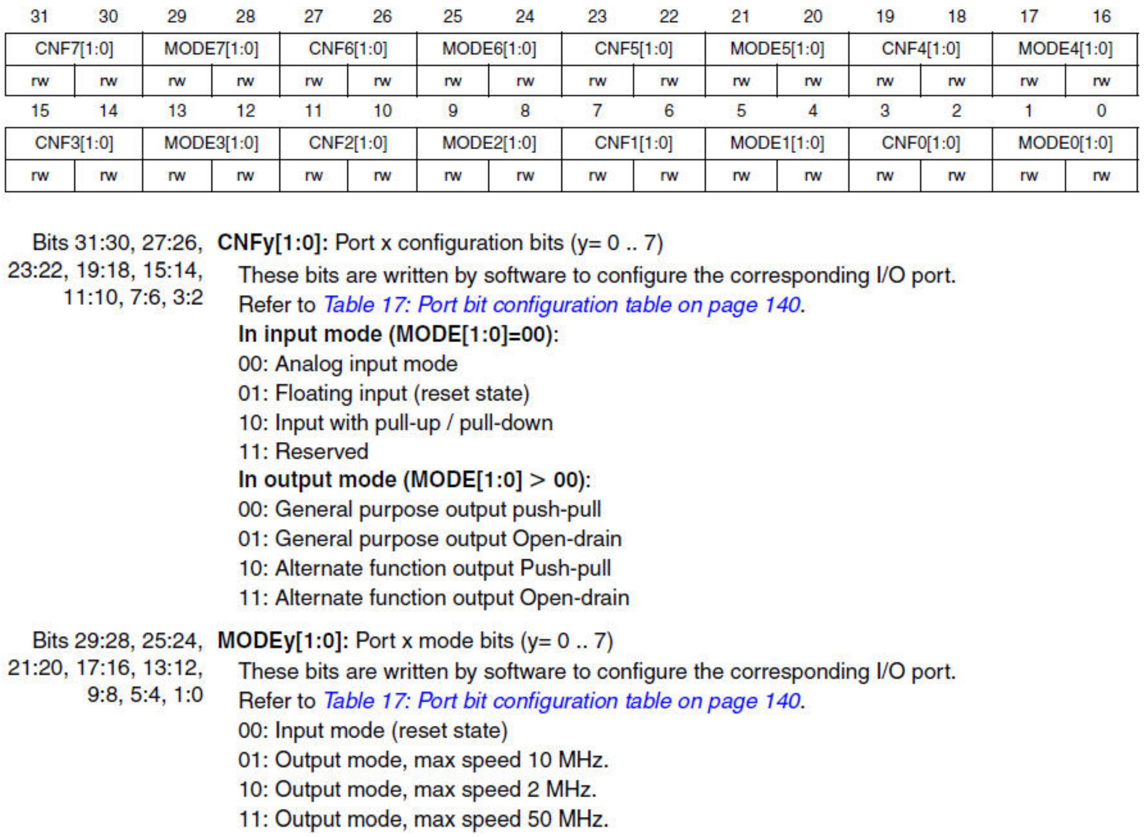
| TIM1/8 pinout | configuration | GPIO configuration |
|---|---|---|
| TIM1/8_CHx | Input capture channel x | Input floating |
| | Output compare channel x | Alternate function push-pull |
| TIM1/8_CHxN | Complementary output channel x | Alternate function push-pull |
| TIM1/8_BKIN | Break input | Input floating |
| TIM1/8_ETR | External trigger timer input | Input floating |



| I2C pinout | Configuration | GPIO configuration |
|---|---|---|
| I2Cx_SCL | I2C clock | Alternate function open drain |
| I2Cx_SDA | I2C Data I/O | Alternate function open drain |

# GPIO registers

- Port configuration register low (GPIOx_CRL) (x=A..G)
  - Address offset: 0x00
  - Reset value: 0x4444 4444



| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CNF7[1:0] | | MODE7[1:0] | | CNF6[1:0] | | MODE6[1:0] | | CNF5[1:0] | | MODE5[1:0] | | CNF4[1:0] | | MODE4[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNF3[1:0] | | MODE3[1:0] | | CNF2[1:0] | | MODE2[1:0] | | CNF1[1:0] | | MODE1[1:0] | | CNF0[1:0] | | MODE0[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2 **CNFy[1:0]:** Port x configuration bits (y= 0 .. 7)

These bits are written by software to configure the corresponding I/O port.
Refer to *Table 17: Port bit configuration table on page 140*.

**In input mode (MODE[1:0]=00):**
00: Analog input mode
01: Floating input (reset state)
10: Input with pull-up / pull-down
11: Reserved

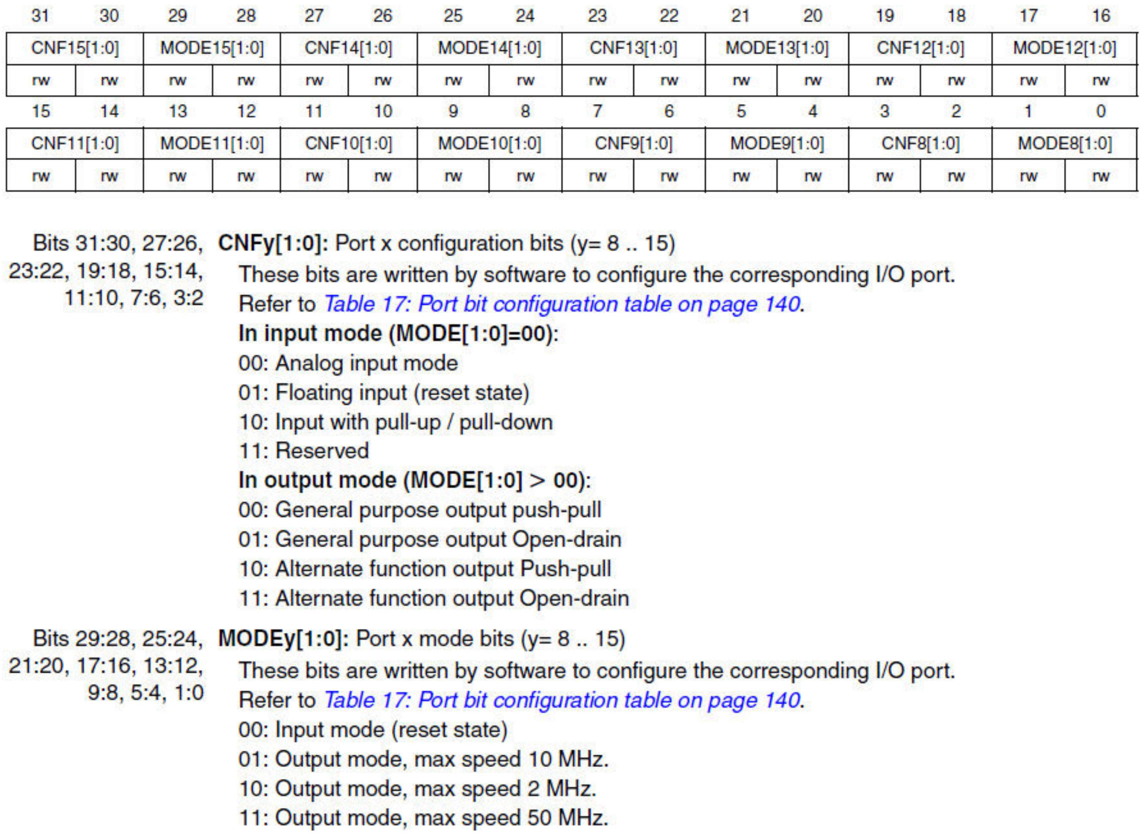
**In output mode (MODE[1:0] > 00):**
00: General purpose output push-pull
01: General purpose output Open-drain
10: Alternate function output Push-pull
11: Alternate function output Open-drain

Bits 29:28, 25:24, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0 **MODEy[1:0]:** Port x mode bits (y= 0 .. 7)

These bits are written by software to configure the corresponding I/O port.
Refer to *Table 17: Port bit configuration table on page 140*.
00: Input mode (reset state)
01: Output mode, max speed 10 MHz.
10: Output mode, max speed 2 MHz.
11: Output mode, max speed 50 MHz.

# GPIO registers

- Port configuration register high (GPIOx_CRH) (x=A..G)
  - Address offset: 0x04
  - Reset value: 0x4444 4444



| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CNF15[1:0] | | MODE15[1:0] | | CNF14[1:0] | | MODE14[1:0] | | CNF13[1:0] | | MODE13[1:0] | | CNF12[1:0] | | MODE12[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNF11[1:0] | | MODE11[1:0] | | CNF10[1:0] | | MODE10[1:0] | | CNF9[1:0] | | MODE9[1:0] | | CNF8[1:0] | | MODE8[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2 **CNFy[1:0]:** Port x configuration bits (y= 8 .. 15)

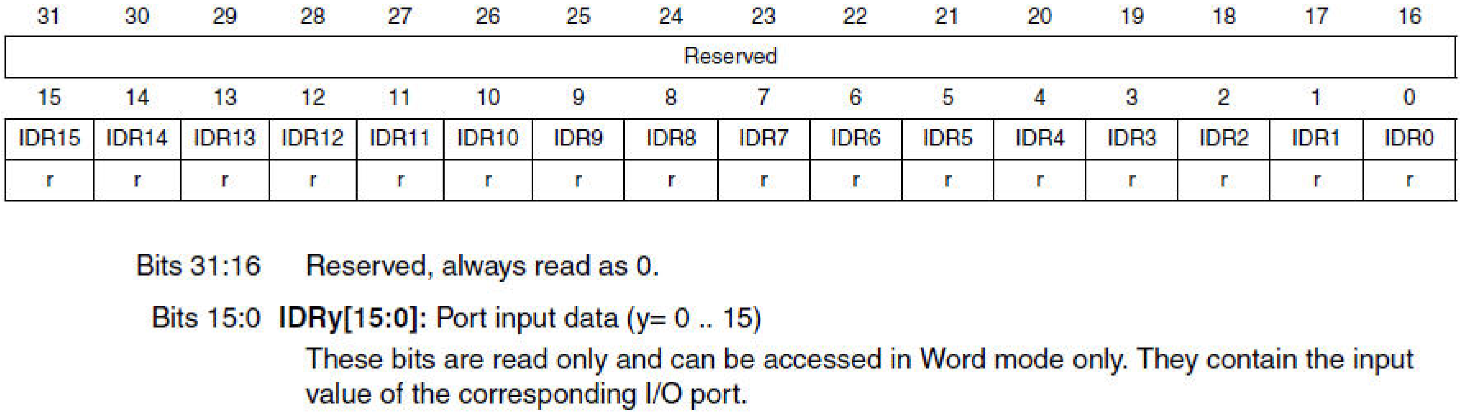
These bits are written by software to configure the corresponding I/O port.
Refer to *Table 17: Port bit configuration table on page 140*.
**In input mode (MODE[1:0]=00):**
00: Analog input mode
01: Floating input (reset state)
10: Input with pull-up / pull-down
11: Reserved
**In output mode (MODE[1:0] > 00):**
00: General purpose output push-pull
01: General purpose output Open-drain
10: Alternate function output Push-pull
11: Alternate function output Open-drain

Bits 29:28, 25:24, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0 **MODEy[1:0]:** Port x mode bits (y= 8 .. 15)

These bits are written by software to configure the corresponding I/O port.
Refer to *Table 17: Port bit configuration table on page 140*.
00: Input mode (reset state)
01: Output mode, max speed 10 MHz.
10: Output mode, max speed 2 MHz.
11: Output mode, max speed 50 MHz.

# GPIO registers

- Port input data register (GPIOx_IDR) (x=A..G)
  - Address offset: 0x08h
  - Reset value: 0x0000 XXXX



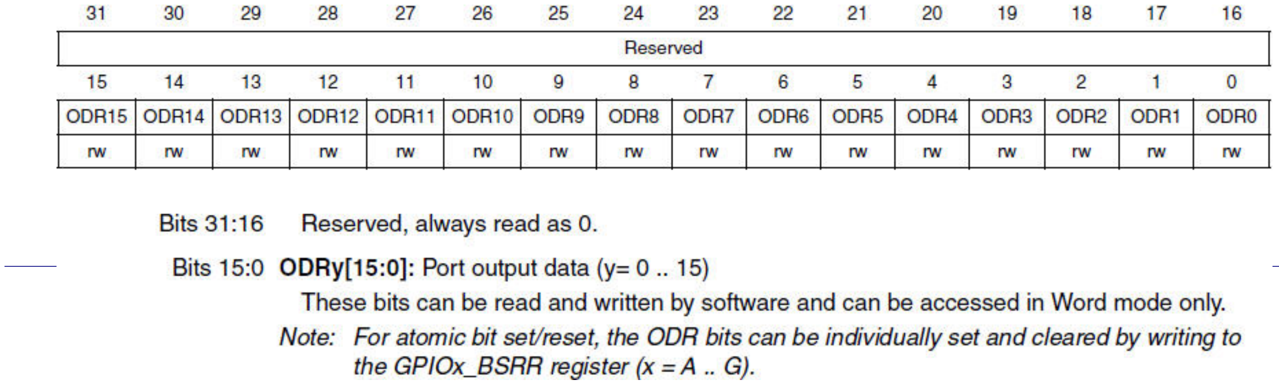
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IDR15 | IDR14 | IDR13 | IDR12 | IDR11 | IDR10 | IDR9 | IDR8 | IDR7 | IDR6 | IDR5 | IDR4 | IDR3 | IDR2 | IDR1 | IDR0 |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bits 31:16 Reserved, always read as 0.

Bits 15:0 **IDRy[15:0]:** Port input data (y= 0 .. 15)

These bits are read only and can be accessed in Word mode only. They contain the input value of the corresponding I/O port.

- Port output data register (GPIOx_ODR) (x=A..G)
  - Address offset: 0x0C
  - Reset value: 0x0000 0000



| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ODR15 | ODR14 | ODR13 | ODR12 | ODR11 | ODR10 | ODR9 | ODR8 | ODR7 | ODR6 | ODR5 | ODR4 | ODR3 | ODR2 | ODR1 | ODR0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 31:16 Reserved, always read as 0.

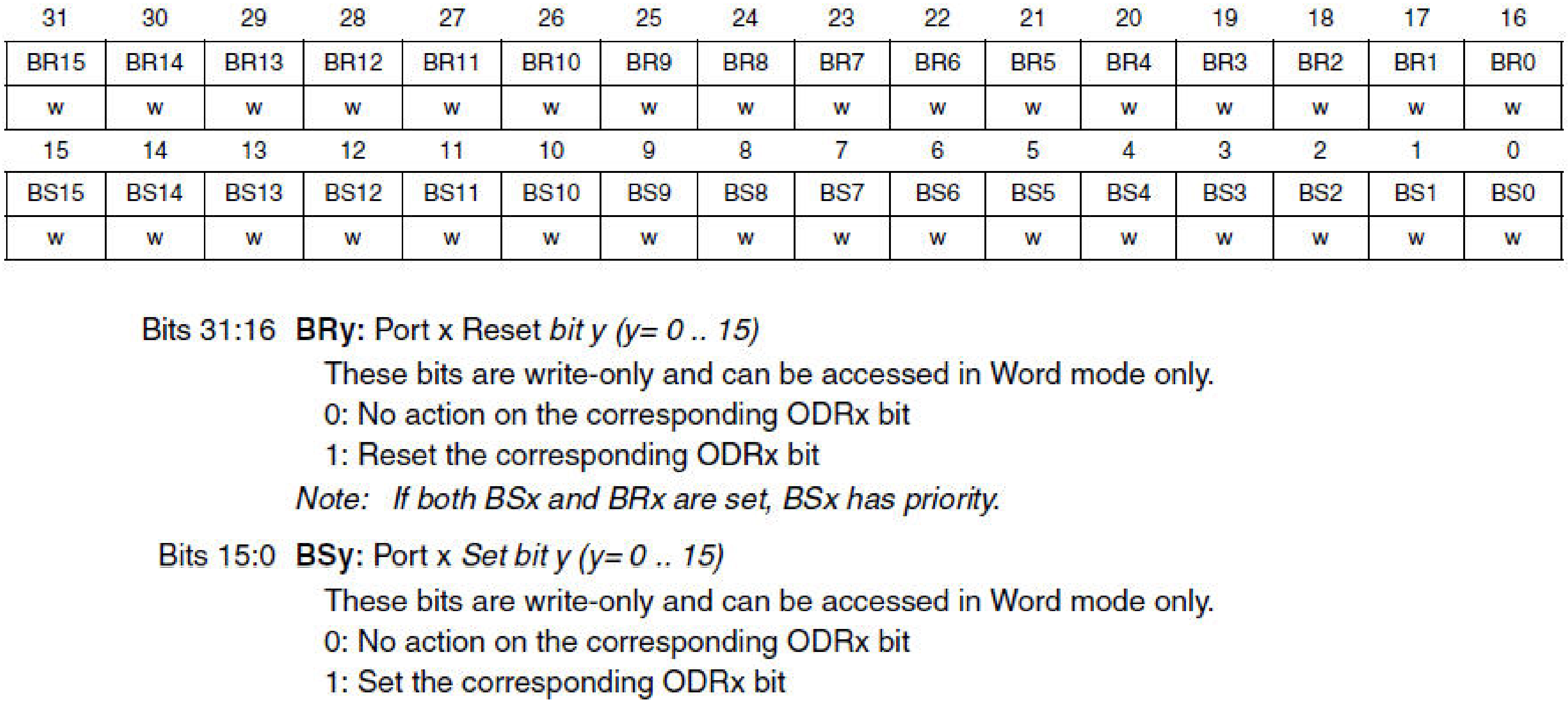
Bits 15:0 **ODRy[15:0]:** Port output data (y= 0 .. 15)

These bits can be read and written by software and can be accessed in Word mode only.

*Note: For atomic bit set/reset, the ODR bits can be individually set and cleared by writing to the GPIOx_BSRR register (x = A .. G).*

# GPIO registers

- Port bit set/reset register (GPIOx_BSRR) (x=A..G)
  - Address offset: 0x10
  - Reset value: 0x0000 0000



| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BS15 | BS14 | BS13 | BS12 | BS11 | BS10 | BS9 | BS8 | BS7 | BS6 | BS5 | BS4 | BS3 | BS2 | BS1 | BS0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:16 **BRy:** Port x Reset *bit y (y= 0 .. 15)*

These bits are write-only and can be accessed in Word mode only.
0: No action on the corresponding ODRx bit
1: Reset the corresponding ODRx bit

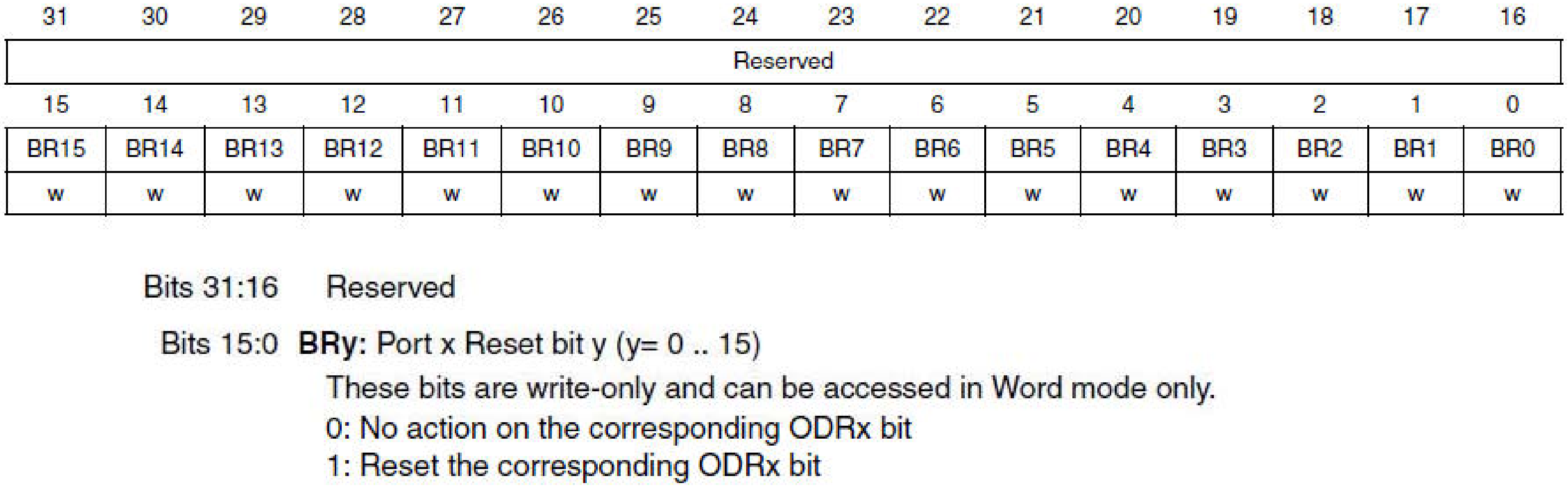
*Note: If both BSx and BRx are set, BSx has priority.*

Bits 15:0 **BSy:** Port x *Set bit y (y= 0 .. 15)*

These bits are write-only and can be accessed in Word mode only.
0: No action on the corresponding ODRx bit
1: Set the corresponding ODRx bit

# GPIO registers

- Port bit reset register (GPIOx_BRR) (x=A..G)
  - Address offset: 0x14
  - Reset value: 0x0000 0000



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:16 Reserved

Bits 15:0 **BRy:** Port x Reset bit y (y= 0 .. 15)

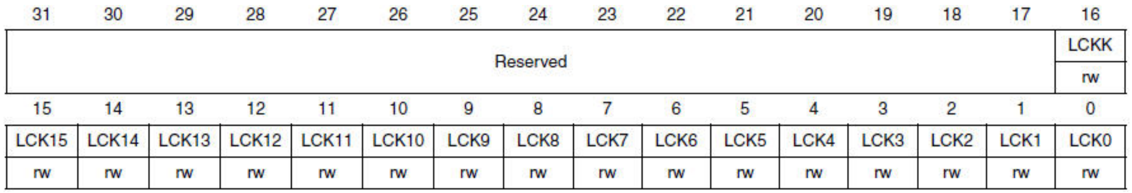
These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

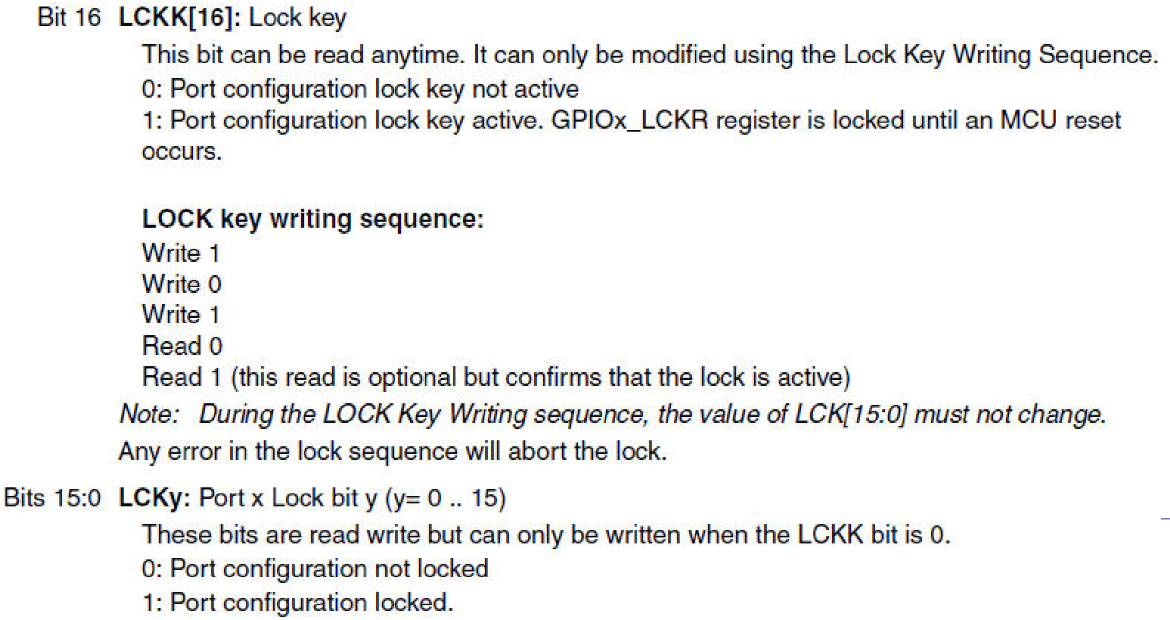
1: Reset the corresponding ODRx bit

# GPIO registers

- Port configuration lock register (GPIOx_LCKR) (x=A..G)
  - This register is used to lock the configuration of the port bits when a correct write sequence is applied to bit 16 (LCKK). The value of bits [15:0] is used to lock the configuration of the GPIO. During the write sequence, the value of LCKR[15:0] must not change. When the LOCK sequence has been applied on a port bit it is no longer possible to modify the value of the port bit until the next reset.
  - Each lock bit freezes the corresponding 4 bits of the control register (CRL, CRH).
  - Address offset: 0x18
  - Reset value: 0x0000 0000



| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LCKK |
| | | | | | | | | | | | | | | | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LCK15 | LCK14 | LCK13 | LCK12 | LCK11 | LCK10 | LCK9 | LCK8 | LCK7 | LCK6 | LCK5 | LCK4 | LCK3 | LCK2 | LCK1 | LCK0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |



Bit 16 **LCKK[16]:** Lock key

This bit can be read anytime. It can only be modified using the Lock Key Writing Sequence.
0: Port configuration lock key not active
1: Port configuration lock key active. GPIOx_LCKR register is locked until an MCU reset occurs.

**LOCK key writing sequence:**
Write 1
Write 0
Write 1
Read 0
Read 1 (this read is optional but confirms that the lock is active)

*Note: During the LOCK Key Writing sequence, the value of LCK[15:0] must not change.*

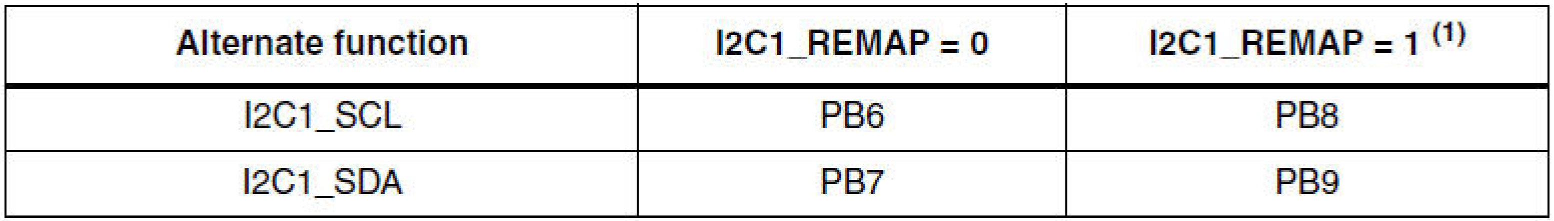
Any error in the lock sequence will abort the lock.

Bits 15:0 **LCKy:** Port x Lock bit y (y= 0 .. 15)

These bits are read write but can only be written when the LCKK bit is 0.
0: Port configuration not locked
1: Port configuration locked.

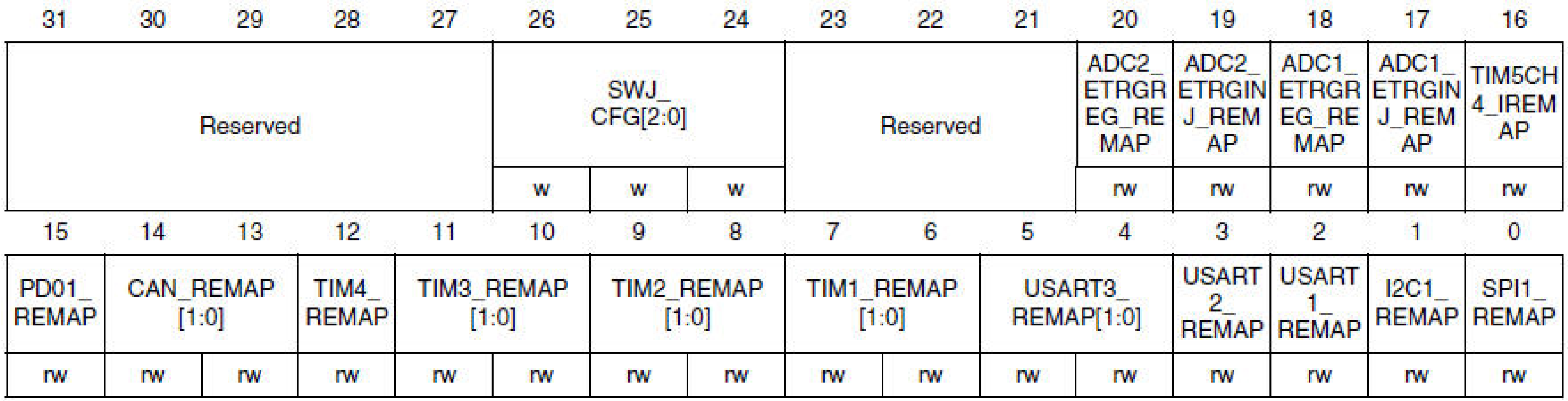
# AF remap and debug I/O configuration register

- EX: I2C1 alternate function remapping



| Alternate function | I2C1_REMAP = 0 | I2C1_REMAP = 1 (1) |
|---|---|---|
| I2C1_SCL | PB6 | PB8 |
| I2C1_SDA | PB7 | PB9 |

- AF remap and debug I/O configuration register (AFIO_MAPR)
  - Address offset: 0x04
  - Reset value: 0x0000 0000



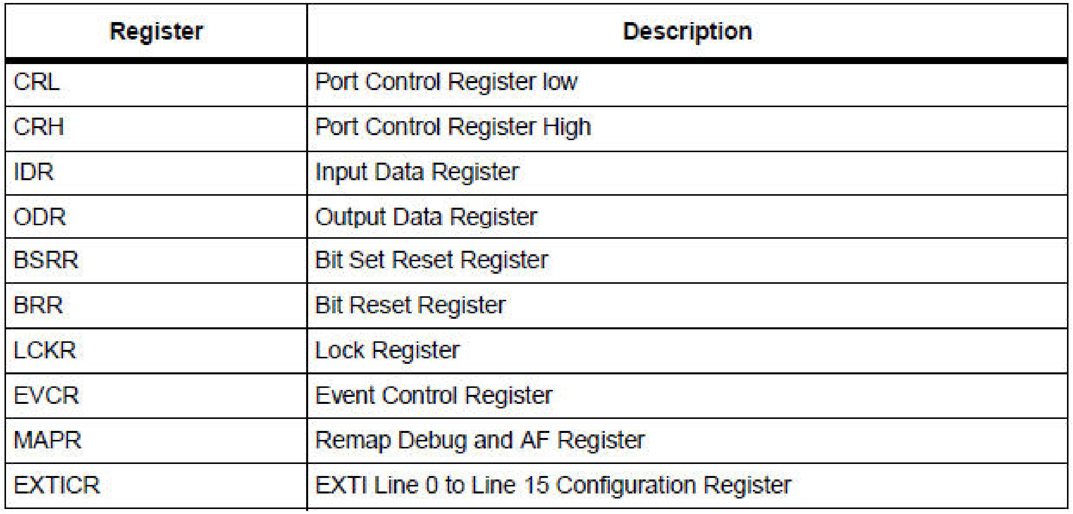
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | SWJ_CFG[2:0] | | | Reserved | | | ADC2_ETRGREG_REMAP | ADC2_ETRGINJ_REMAP | ADC1_ETRGREG_REMAP | ADC1_ETRGINJ_REMAP | TIM5CH4_IREMAP |
| | | | | | w | w | w | | | | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PD01_REMAP | CAN_REMAP[1:0] | | TIM4_REMAP | TIM3_REMAP[1:0] | | TIM2_REMAP[1:0] | | TIM1_REMAP[1:0] | | USART3_REMAP[1:0] | | USART2_REMAP | USART1_REMAP | I2C1_REMAP | SPI1_REMAP |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

# STM32F10x Standard Library for GPIO

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

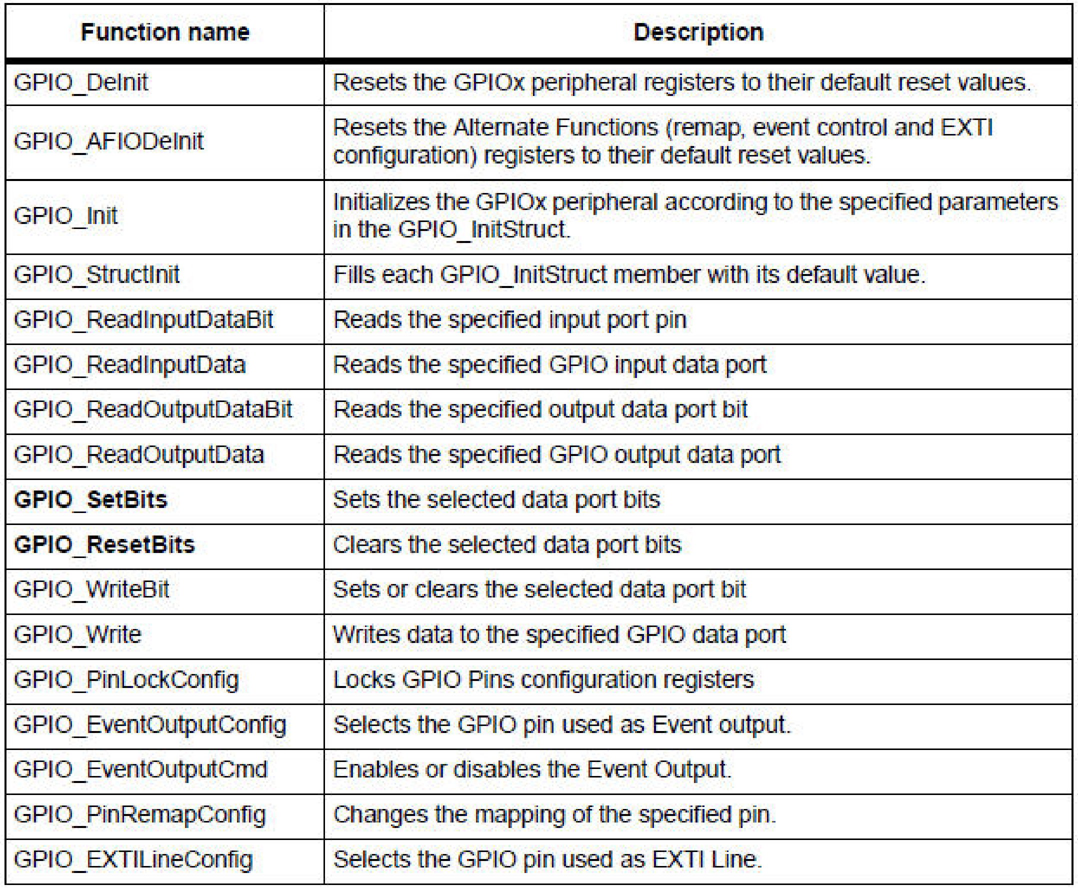
```
typedef struct
{
vu32 CRL;
vu32 CRH;
vu32 IDR;
vu32 ODR;
vu32 BSRR;
vu32 BRR;
vu32 LCKR;
} GPIO_TypeDef;
typedef struct
{
vu32 EVCR;
vu32 MAPR;
vu32 EXTICR[4];
} AFIO_TypeDef;
...
#define PERIPH_BASE ((u32)0x40000000)
#define APB1PERIPH_BASE PERIPH_BASE
#define APB2PERIPH_BASE (PERIPH_BASE + 0x10000)
#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 GPIOF_BASE (APB2PERIPH_BASE + 0x1C00)
#define GPIOG_BASE (APB2PERIPH_BASE + 0x2000)
```



| 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 |

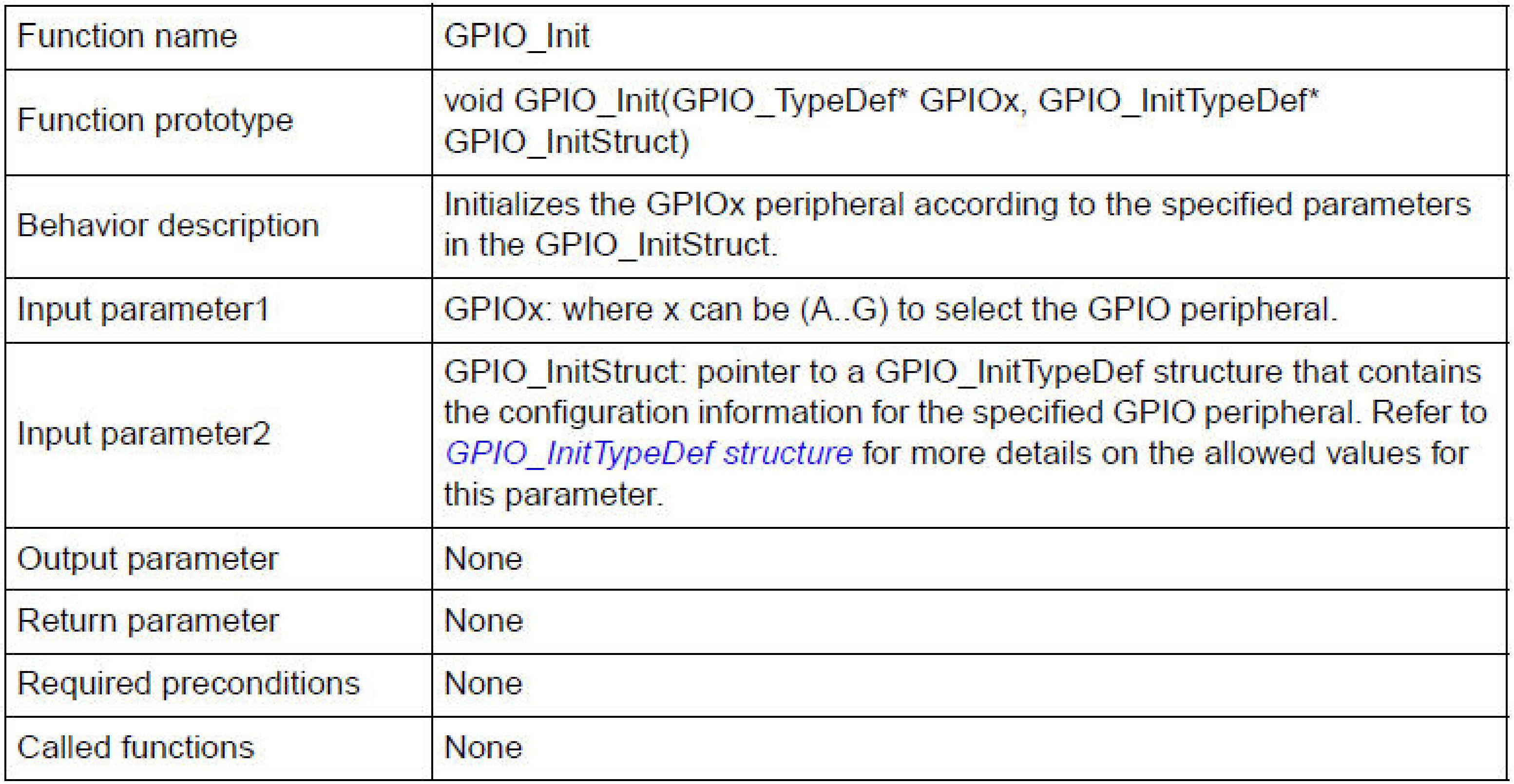
# GPIO firmware library functions

- 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. |

# 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..G) 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 *GPIO_InitTypeDef structure* for more details on the allowed values for this parameter. |
| Output parameter | None |
| Return parameter | None |
| Required preconditions | None |
| Called functions | None |

- **Example:**

  /* Configure all the GPIOF in Output PP mode */

  GPIO_Init(GPIOF, &GPIO_InitStructure);

# GPIO_InitTypeDef structure

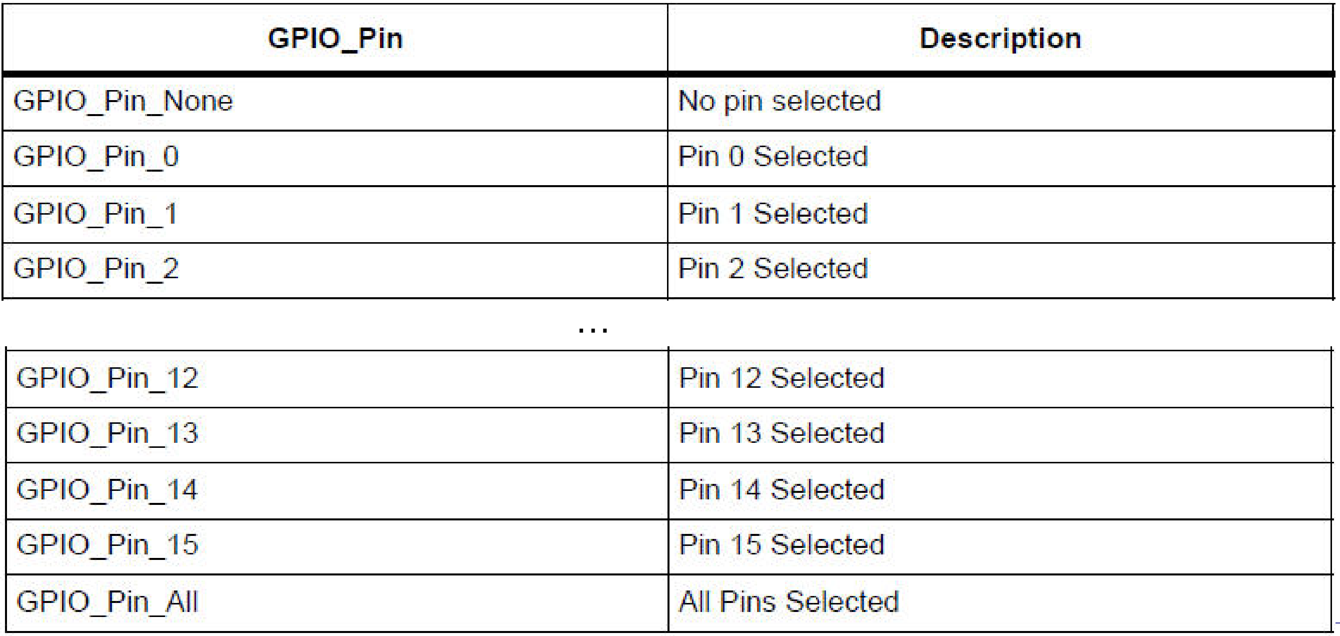
□ The GPIO_InitTypeDef structure is defined in the stm32f10x_gpio.h file:

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

□ **Example:**

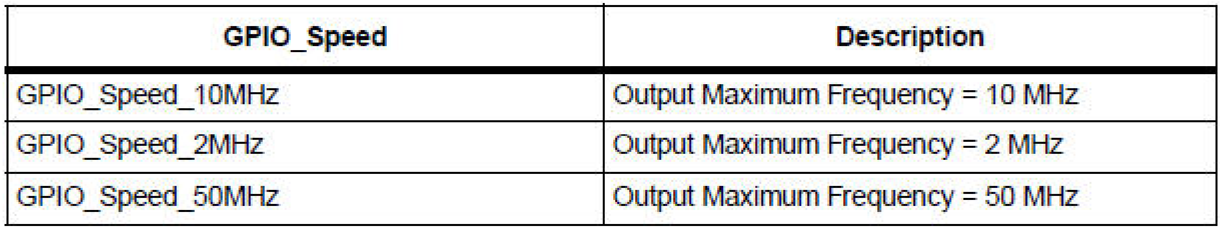
```
/* Configure all the GPIOA in Input Floating mode
        */
GPIO_InitTypeDef GPIO_InitStructure;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11;
```

□ GPIO_Pin



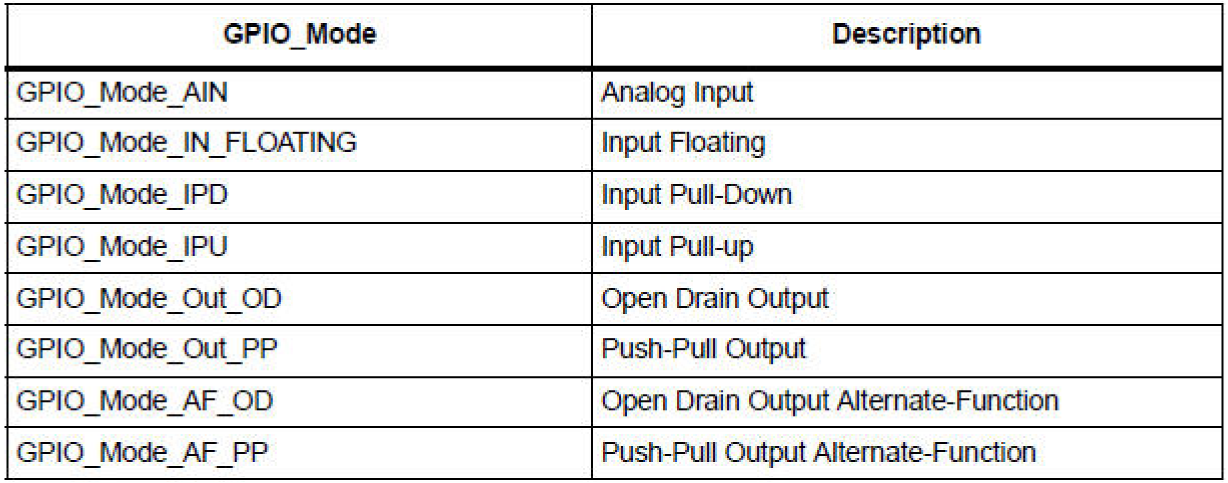
| GPIO_Pin | Description |
|---|---|
| GPIO_Pin_None | No pin selected |
| GPIO_Pin_0 | Pin 0 Selected |
| GPIO_Pin_1 | Pin 1 Selected |
| GPIO_Pin_2 | Pin 2 Selected |
| … | |
| GPIO_Pin_12 | Pin 12 Selected |
| GPIO_Pin_13 | Pin 13 Selected |
| GPIO_Pin_14 | Pin 14 Selected |
| GPIO_Pin_15 | Pin 15 Selected |
| GPIO_Pin_All | All Pins Selected |

## □ GPIO_Speed



| GPIO_Speed | 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 | 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 |

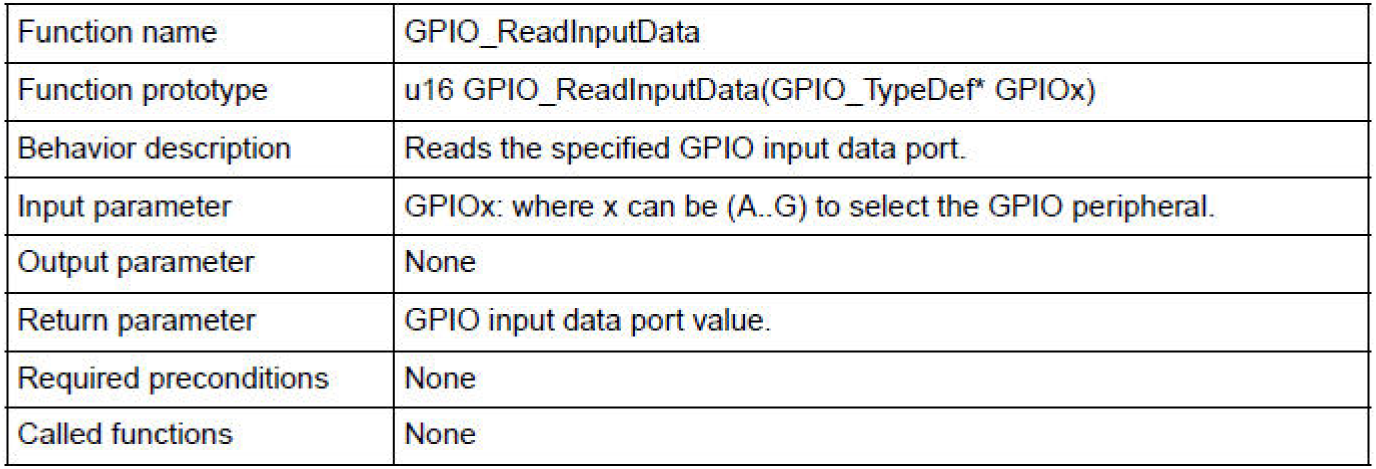
**Example:**

```
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
```

- **Example:**

```
/* Configure all the GPIOA in Input Floating mode */
GPIO_InitTypeDef GPIO_InitStructure;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

# GPIO_ReadInputData function

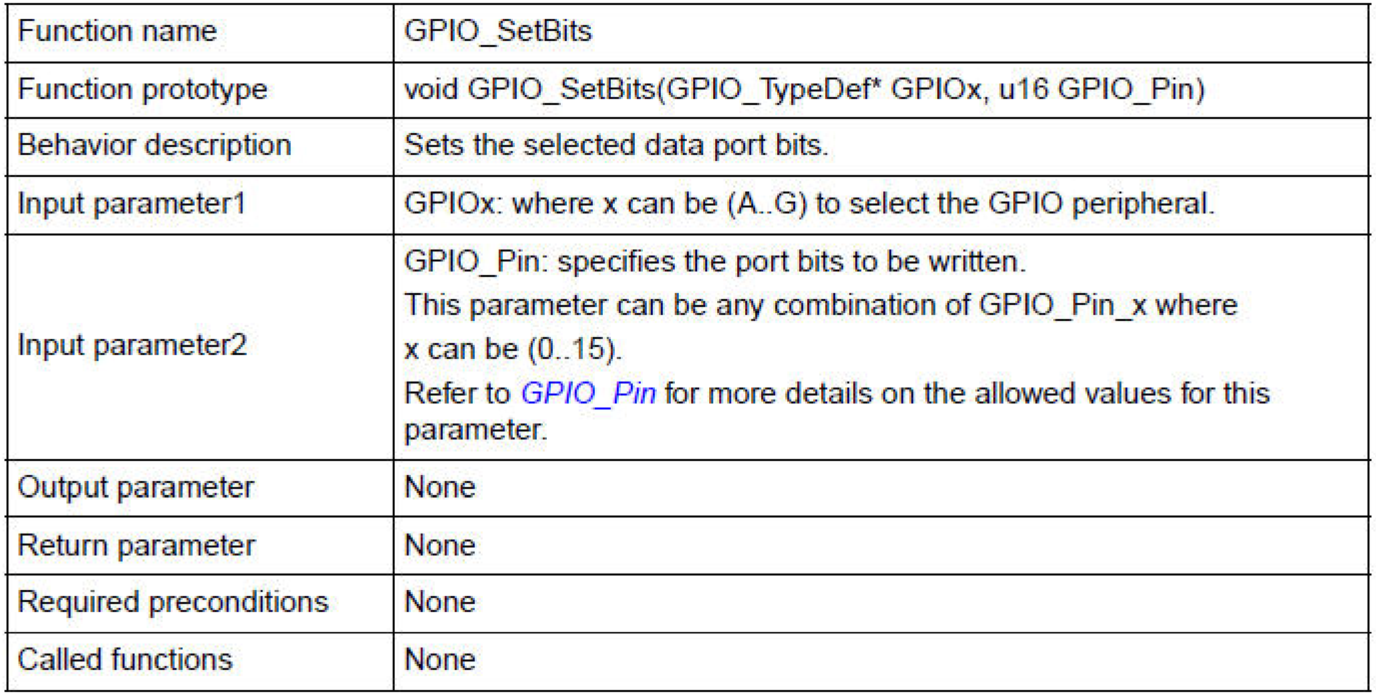


| | |
|---|---|
| Function name | GPIO_ReadInputData |
| Function prototype | u16 GPIO_ReadInputData(GPIO_TypeDef* GPIOx) |
| Behavior description | Reads the specified GPIO input data port. |
| Input parameter | GPIOx: where x can be (A..G) to select the GPIO peripheral. |
| Output parameter | None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

Example:

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

# GPIO_SetBits

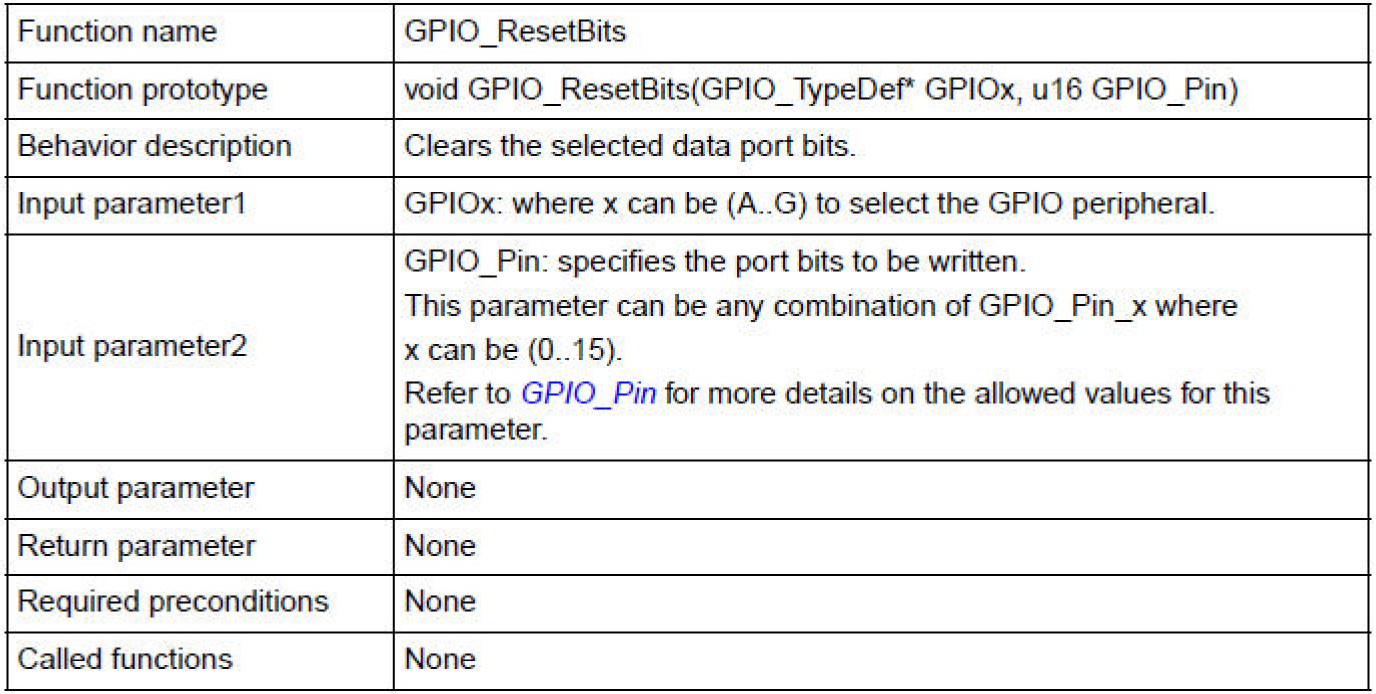


| | |
|---|---|
| Function name | GPIO_SetBits |
| Function prototype | void GPIO_SetBits(GPIO_TypeDef* GPIOx, u16 GPIO_Pin) |
| Behavior description | Sets the selected data port bits. |
| Input parameter1 | GPIOx: where x can be (A..G) to select the GPIO peripheral. |
| Input parameter2 | GPIO_Pin: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

**Example:**

```
/* Set the GPIOA port pin 11 */
GPIO_SetBits(GPIOA, GPIO_Pin_11);
```

# □ GPIO_ResetBits

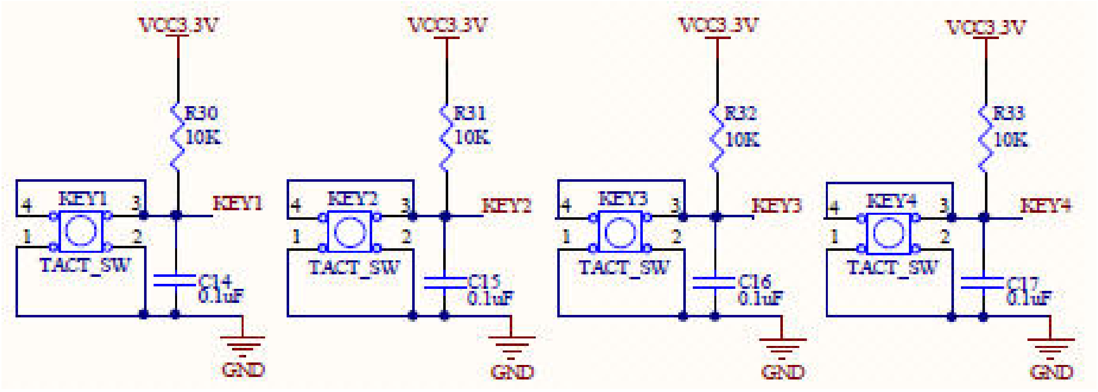


| | |
|---|---|
| Function name | GPIO_ResetBits |
| Function prototype | void GPIO_ResetBits(GPIO_TypeDef* GPIOx, u16 GPIO_Pin) |
| Behavior description | Clears the selected data port bits. |
| Input parameter1 | GPIOx: where x can be (A..G) to select the GPIO peripheral. |
| Input parameter2 | GPIO_Pin: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

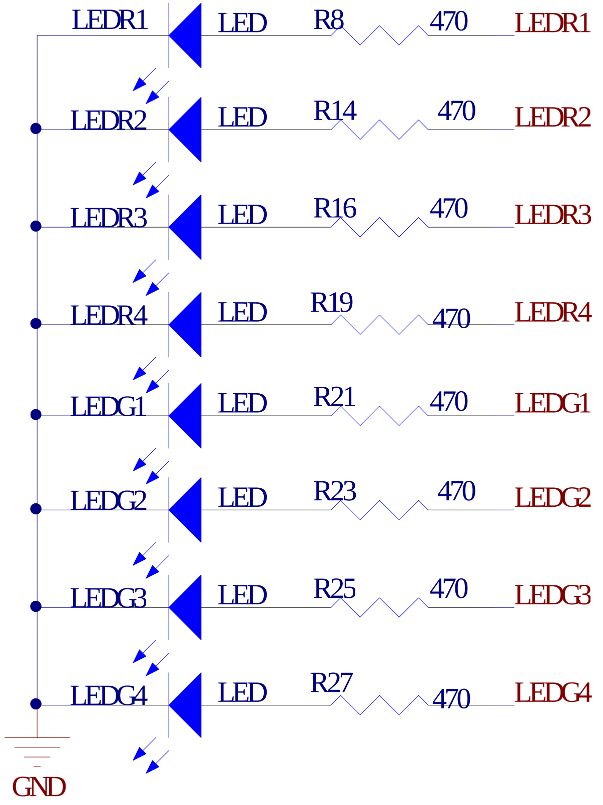
```
/* Clears the GPIOF port pin 11*/
GPIO_ResetBits(GPIOF, GPIO_Pin_11);
```

# 上機實作練習

- □ 試使用STM32F10X之GPIO標準函式庫，寫一段程式其動作當按下KEY1時，LED全亮，按下KEY2時， LED全滅，按下KEY3，LED輪流的由右至左點亮一次， 按下KEY4，LED輪流的由左至右點亮一次。



LED



# Q & A

**_WU-YANG_**

**_Technology Co., Ltd._**