

GPIO (GENERAL PURPOSE INPUT/OUTPUT)

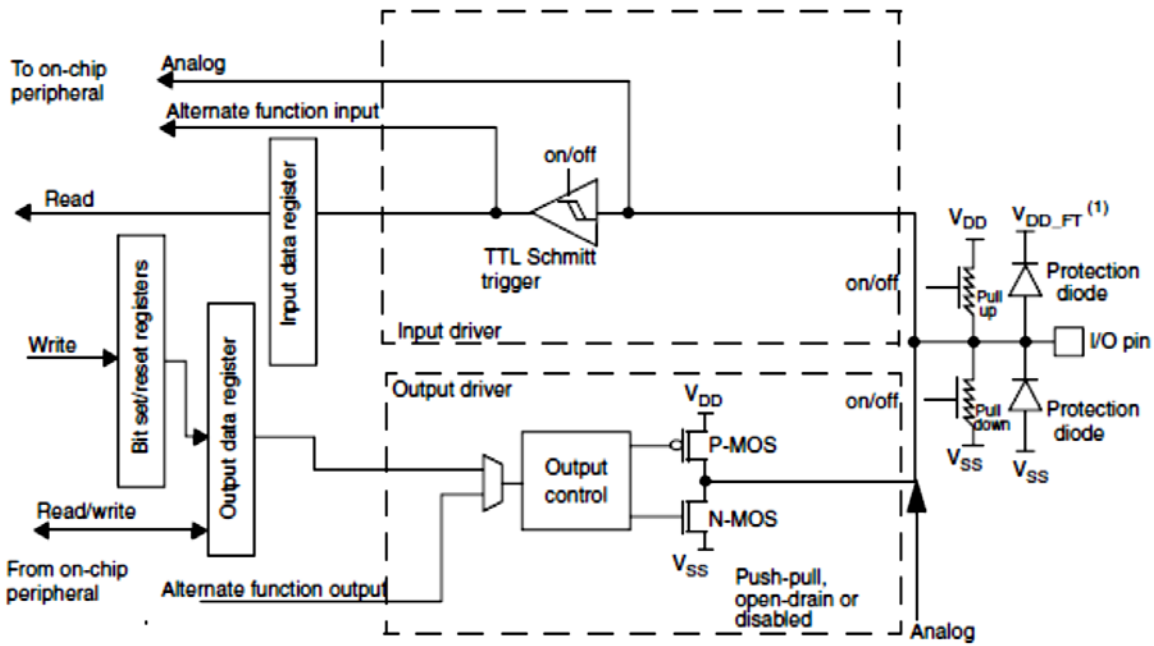
1. Présentation

STM32F407 dispose de 9 ports d'entrées/sorties de 16 bits (dénommés GPIOA à GPIOI), partagés avec d'autres périphériques,. A chaque port I/O est associé quatre registres de configuration 32 bits (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR et GPIOx_PUPDR), deux registres de données 32bits (GPIOx_IDR and GPIOx_ODR), un register de 32bits set/reset (GPIOx_BSRR), un registre 32bits de verrouillage (GPIOx_LCKR) et deux registres 32 bits pour les fonctions secondaire (Alternate function) (GPIOx_AFRH and GPIOx_AFRL).

2. Configuration des broches

Chaque broche du port I/O peut être configurée selon les modes suivants :

- Entrée flottante
- Entrée pull-up (PU)
- Entrée pull-down (PD)
- Analogique
- Sortie drain ouvert (OD) avec résistance pull-up ou pull-down
- Sortie push-pull (PP) avec résistance pull-up ou pull-down
- Alternate Function push-pull avec résistance pull-up ou pull-down
- Alternate Function drain ouvert avec résistance pull-up ou pull-down



La configuration de chaque proche est décrite par le tableau suivant :

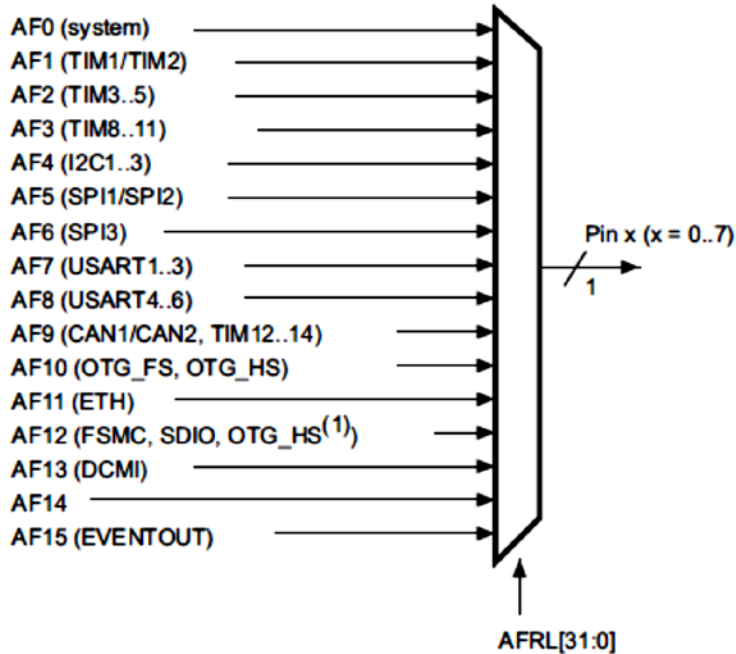
MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		Configuration des I/O	
01	0	SPEED [B:A]				GP output	PP
	0					GP output	PP + PU
	0					GP output	PP + PD
	0					Reservé	
	1					GP output	OD
	1					GP output	OD + PU
	1					GP output	OD + PD
	1					Reservé (GP output OD)	
10	0	SPEED [B:A]				AF	PP
	0					AF	PP + PU
	0					AF	PP + PD
	0					Reservé	
	1					AF	OD
	1					AF	OD + PU
	1					AF	OD + PD
	1					Reservé	
00	x	x	x	0	0	input	Floating
	x	x	x	0	1	input	PU
	x	x	x	1	0	input	PD
	x	x	x	1	1	Reservé (input floating)	
11	x	x	x	0	0	Input / output	Analog
	x	x	x	0	1	Reservé	
	x	x	x	1	0		
	x	x	x	1	1		

GP = general-purpose, PP = push-pull, PU = pull-up, PD = pull-down, OD = open-drain, AF = alternate function

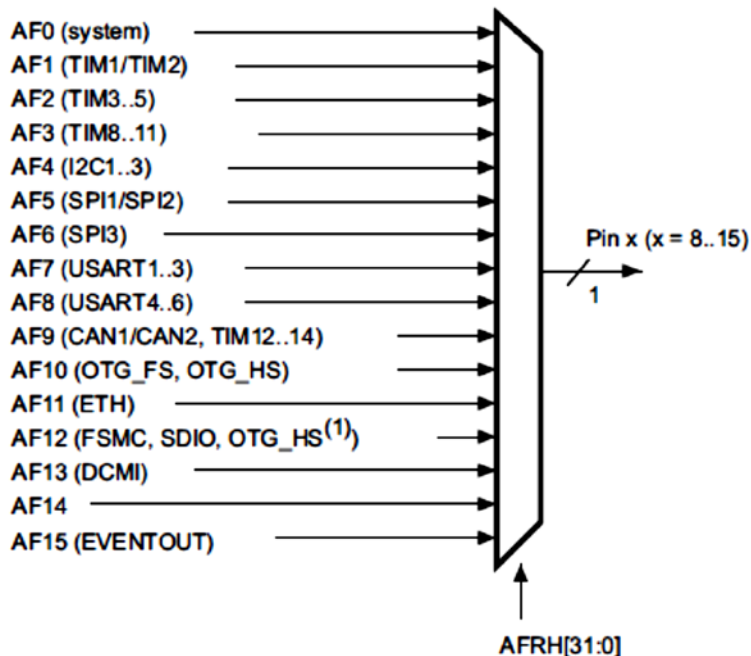
2.1. Multiplexage des broches

Chaque broche I/O peut être reliée à l'une de seize fonctions secondaires (Alternate Function AF0 .. AF15) à travers les registres GPIOx_AFRL et GPIO_AFRH :

For pins 0 to 7, the GPIOx_AFRL[31:0] register selects the dedicated alternate function



For pins 8 to 15, the GPIOx_AFRH[31:0] register selects the dedicated alternate function



2.2. Le fichier entête STM32F4xx.h

Le fichier entête STM32F4xx.h comporte la définition des tous les registres des périphériques ainsi que la définition de toutes les constantes.

Définition des PORTS

```
#define APB1PERIPH_BASE      PERIPH_BASE

#define GPIOA_BASE          (AHB1PERIPH_BASE + 0x0000)
#define GPIOB_BASE          (AHB1PERIPH_BASE + 0x0400)
#define GPIOC_BASE          (AHB1PERIPH_BASE + 0x0800)
#define GPIOD_BASE          (AHB1PERIPH_BASE + 0x0C00)
#define GPIOE_BASE          (AHB1PERIPH_BASE + 0x1000)
#define GPIOF_BASE          (AHB1PERIPH_BASE + 0x1400)
#define GPIOG_BASE          (AHB1PERIPH_BASE + 0x1800)
#define GPIOH_BASE          (AHB1PERIPH_BASE + 0x1C00)
#define GPIOI_BASE          (AHB1PERIPH_BASE + 0x2000)

#define GPIOA                ((GPIO_TypeDef *) GPIOA_BASE)
#define GPIOB                ((GPIO_TypeDef *) GPIOB_BASE)
#define GPIOC                ((GPIO_TypeDef *) GPIOC_BASE)
#define GPIOD                ((GPIO_TypeDef *) GPIOD_BASE)
#define GPIOE                ((GPIO_TypeDef *) GPIOE_BASE)
#define GPIOF                ((GPIO_TypeDef *) GPIOF_BASE)
#define GPIOG                ((GPIO_TypeDef *) GPIOG_BASE)
#define GPIOH                ((GPIO_TypeDef *) GPIOH_BASE)
#define GPIOI                ((GPIO_TypeDef *) GPIOI_BASE)

typedef struct
{
    __IO uint32_t MODER;      /* GPIO port mode register,           Address offset: 0x00*/
    __IO uint32_t OTYPER;     /* GPIO port output type register,     Address offset: 0x04*/
    __IO uint32_t OSPEEDR;    /* GPIO port output speed register,    Address offset: 0x08*/
    __IO uint32_t PUPDR;     /* GPIO port pull-up/pull-down register, Address offset: 0x0C*/
    __IO uint32_t IDR;        /* GPIO port input data register,      Address offset: 0x10*/
    __IO uint32_t ODR;        /* GPIO port output data register,     Address offset: 0x14*/
    __IO uint16_t BSRRL;      /* GPIO port bit set/reset register,   Address offset: 0x18*/
    __IO uint32_t LCKR;       /* GPIO port configuration lock register, Address offset: 0x1C*/
    __IO uint32_t AFR[2];    /* GPIO alternate function registers, Address offset: 0x20-0x24*/
} GPIO_TypeDef;
```

Structure de configuration d'une broche (fichier STM32F4xx_hal_gpio.h)

```
typedef struct
{
    uint32_t Pin;
    uint32_t Mode;
    uint32_t Pull;
    uint32_t Speed;
    uint32_t Alternate;
}GPIO_InitTypeDef;
```

Accès aux ports à partir des registres

Registre en lecture

`Vu16 = GPIOx → IDR ; // x : A à I`

Écriture sur un port

`GPIOx → ODR = X16 ;`

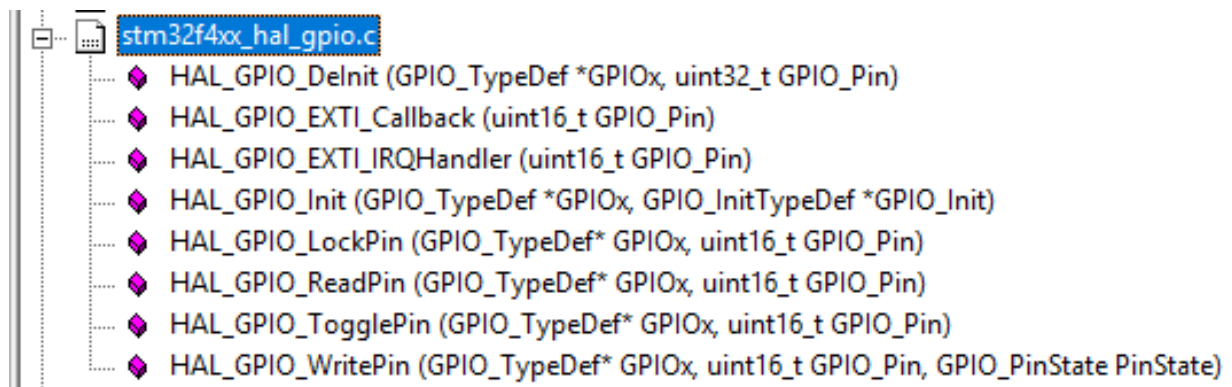
Mettre un bit à 1 ou 0

`GPIOx → BSRR = X32 ; // partie haute pour RESET et partie basse pour SET`

`GPIOx → BSRRLL = X16 ; // partie basse pour SET`

`GPIOx → BSRRH = X16 ; // partie haute pour RESET`

Accès aux ports à travers la librairie (CMSIS)



Exemple : `HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_SET) ;`