TD : Accès Registre / Firmware

Exercice :

La structure des GPIOs de la famille F4 est différente de celle de la famille F1. Les bits de configuration des 16 pins ne sont pas regroupés ensemble, mais sont séparés dans plusieurs registres (*voir **Annexe 3***).

Ainsi, on trouve pour chaque pin

- 2 bits pour la configuration du mode (Input =00, Output=01, Alternate=10 et Analog=11) dans le registre **GPIOx_MODER**
- 2 bits pour la configuration de la vitesse (low, medium, fast et high) pour chaque pin dans le registre **GPIOx_OPSPEEDR .**

Etc…

On veut développer une couche d'abstraction matérielle (Firmware) qui permet de faciliter la programmation des GPIOs de telle sorte à ce qu'au niveau utilisateur (main.c) la configuration peut se faire ainsi :

```
GPIO_InitTypedef       GPIO_Struct ;

GPIO_Struct.GPIO_Mode= ….;

GPIO_Struct.GPIO_Otype=…. ;

GPIO_Struct.GPIO_OSpeed=…. ;

GPIO_Struct.GPIO_PuPd=…. ;

GPIO_Struct.GPIO_Pin = GPIO_Pin_i | GPIO_Pin_j | ….. ;  // (i,j,k : des valeurs de  0..15)

GPIO_Init ( GPIOx ,  GPIO_Struct) ;  //x pouvant être A, B, C ou D.
```

*Remarque : **GPIO_Init** étant une fonction qui permet de configurer les pins du GPIOx en fonction des paramètres passés au niveau de GPIO_InitStruct et définie aunsi :*

GPIO_Init *(GPIO_Typedef* GPIOx , &GPIO_InitTypedef GPIO_InitStruct)*

Questions :

Q1- Donner la partie à ajouter au fichier stm32f4xx.h qui devrait contenir la partie memory map relative au GPIOs . (Voir ***Annexe1*** pour adresses et offset).

Q2 – Ecrire un programme qui permet de faire clignoter une Led connectée au pin PD12 en utilisant l'accès direct aux registres sachant que la famille **F4** contient uniquement un registre BSRR (pas de BRR comme F1) qui permet d'assurer la mise à 1 et la remise à 0 des 16 bits de ODR. (Voir **Annexe2**).

Rq : sans configurer les pins

Q3- Donner le code à ajouter au fichier stm32f4xx_gpio.h et qui permet de décrire les paramètres de configuration du périphérique GPIO.

Q4 – Donner le code de la fonction **GPIO_Init.**

Q5 – Donner les codes des fonctions ***GPIO_SetBits*** et ***GPIO_ResetBits*** qui permettent de mettre à 1 (resp à 0) un pin ou un ensemble de pins d'un GPIO donné.

Annexe :

Annexe 1 : Adresses et Offsets

GPIOx	A	B	C	D
@	0x4002 0000	0x4002 0400	0x4002 0800	0x4002 0C00

Registre	MODER	OTYPER	OSPEEDR	PUPDR	IDR	ODR	BSRR
Offset	0x00	0x04	0x08	0x0C	0x10	**0x14**	**0x18**

Annexe 2 : Le registre BSRR

GPIO port bit set/reset register (GPIOx_BSRR) (x = A..I/J/K)

Address offset: 0x18

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, half-word or byte mode. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Resets 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)

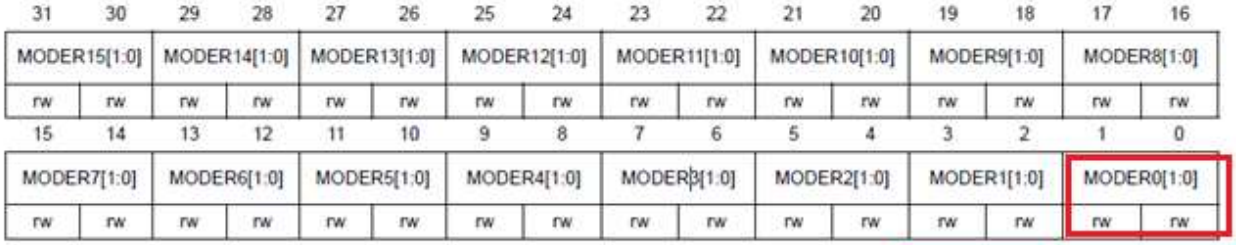
These bits are write-only and can be accessed in word, half-word or byte mode. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Sets the corresponding ODRx bit

Annexe 3 : Les registres de configuration :

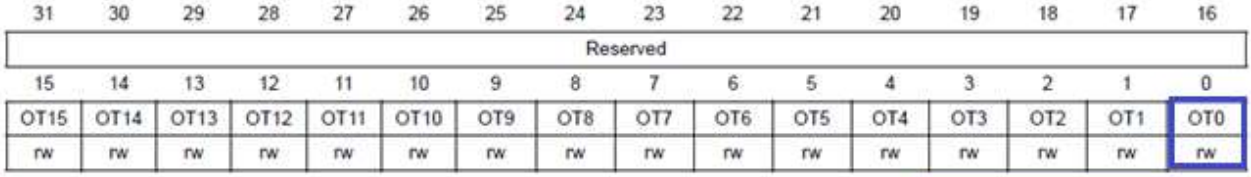
GPIO port mode register (GPIOx_MODER) (x = A..I/J/K)



Bits 2y:2y+1 **MODERy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O direction mode

00: Input (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

GPIO port output type register (GPIOx_OTYPER) (x = A..I/J/K)



Bits 15:0 **OTy**: Port x configuration bits (y = 0..15)
These bits are written by software to configure the output type of the I/O port.

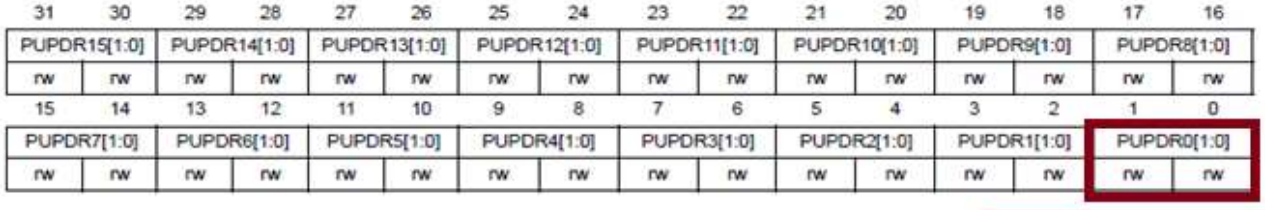
0: Output push-pull (reset state)
1: Output open-drain

GPIO port output speed register (GPIOx_OSPEEDR) (x = A..I/J/K)

Bits 2y:2y+1 **OSPEEDRy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O output speed.

00: Low speed
01: Medium speed
10: Fast speed
11: High speed

GPIO port pull-up/pull-down register (GPIOx_PUPDR) (x = A..I/J/K)



Bits 2y:2y+1 **PUPDRy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O pull-up or pull-down

00: No pull-up, pull-down
01: Pull-up
10: Pull-down