

TD 4 STM32

Exercice 1. Temporisation

Soit le sous-programme assembleur de temporisation logicielle suivant :

```
.section .data

CPT:
    .word 349500

.text
5    ldr r0,=CPT
5    ldr r0,[r0]
7boucleattente:
3    subs r0,r0,#1
3    bne boucleattente
```

1. Etablir l'organigramme de ce sous-programme.
2. Calculer la durée de la temporisation logicielle sachant que l'instruction subs s'exécute en 1 cycle d'horloge, le branchement conditionnel en 2 cycles d'horloge et que cette dernière est cadencée à 2,097MHz.
(le nombre de cycles pour chaque opération a été mesuré de manière empirique)
3. Quelle est la temporisation maximale réalisable avec cette temporisation ? Quelle valeur faut-il donner à CPT pour une temporisation de 10s ?
4. Quelle est la longueur de la temporisation si CPT = 0 ?

Exercice 2. Mémoire, tableau, adressage indexé

1. Que signifie GPIO ? Expliquer la fonction et le fonctionnement des registres suivants :
 - Registre GPIOx_MODER
 - Registre GPIOx_OTYPER
 - Registre GPIOx_PUPDR
 - Registre GPIOx_IDR
 - Registre GPIOx_ODR
2. Que signifie RCC ? Expliquer la fonction et le fonctionnement du registre RCC_AHB1ENR

Exercice 3.

- 1) Expliquer l'exécution du programme ci-dessous

```

ldr r0, =0x40020000
ldr r1, [r0]
ldr r2, = 0xFFF3F3FF
and r1, r1,r2
str r1,[r0]

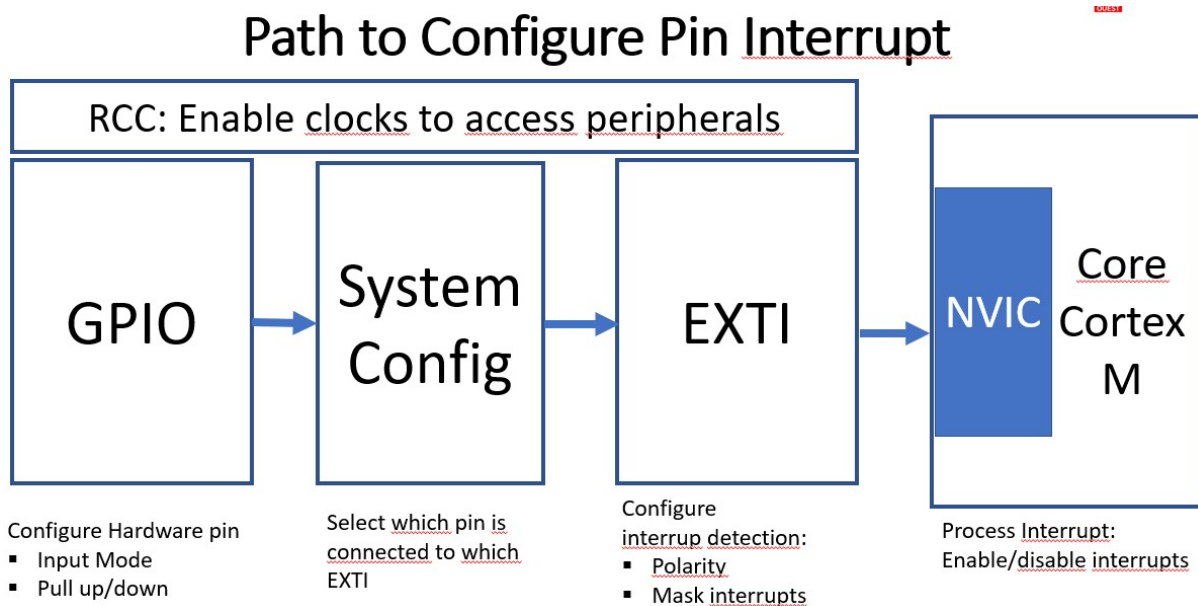
```

- 2) Réécrire le programme en utilisant la fonction bfc qui est expliquée dans le programming manual

Exercice 4. initialisation ports I/O

Un bouton poussoir est relié à la broche PC13 en entrée pour détecter une interruption. Une LED est reliée à la broche PA5 en sortie PUSH-PULL. Configurer ces broches pour gérer ces signaux.

Pour rappel, voici tous les périphériques qui doivent être configurés.



```

/* Allumer la LED verte */

/* Enable port A */
ldr r0, =0x4002381C
ldr r1, =0x00000001
ldr r2, [r0]
orrs r1,r2
str r1, [r0]

/* configure pin in output mode */
ldr r0, =0x40020000
ldr r1, =0x00000400
ldr r2, [r0]
orrs r1,r2
str r1, [r0]

ldr r1, =0xFFFF7FF
ldr r2, [r0]
ands r1,r2
str r1, [r0]

/* configure output pin */
ldr r0, =0x40020014
ldr r1, =0x00000020
ldr r2, [r0]
orrs r1,r2
str r1, [r0]

```

Indication : le ‘memory map’ sert à obtenir les adresses de bases sont dans la datasheet et les offsets dans le reference manual