STM32 INTERRUPTIONS

Objectifs : Mise en œuvre des interruptions. (General Purpose Input Output interrupts).

Matériel : Ce TP utilise une NUCLEO-F411RE, mais n'importe quelle autre carte NUCLEO convient.
Logiciel : MBED

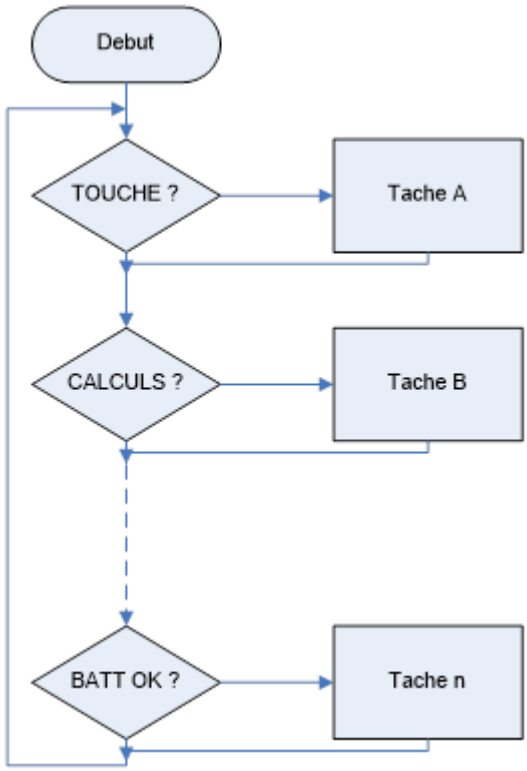
1 Généralités sur les interruptions

Généralement un programme évolue en fonction d'événements physiques. (touche enfoncée, gestion d'un afficheur, dépassement d'une température, réception d'une communication série…)

Prenons l'exemple d'une simple calculette. Le microcontrôleur doit :

- Détecter l'appui sur les touches.
- Effectuer les calculs demandés
- Afficher le résultat
- Surveiller l'état des batteries

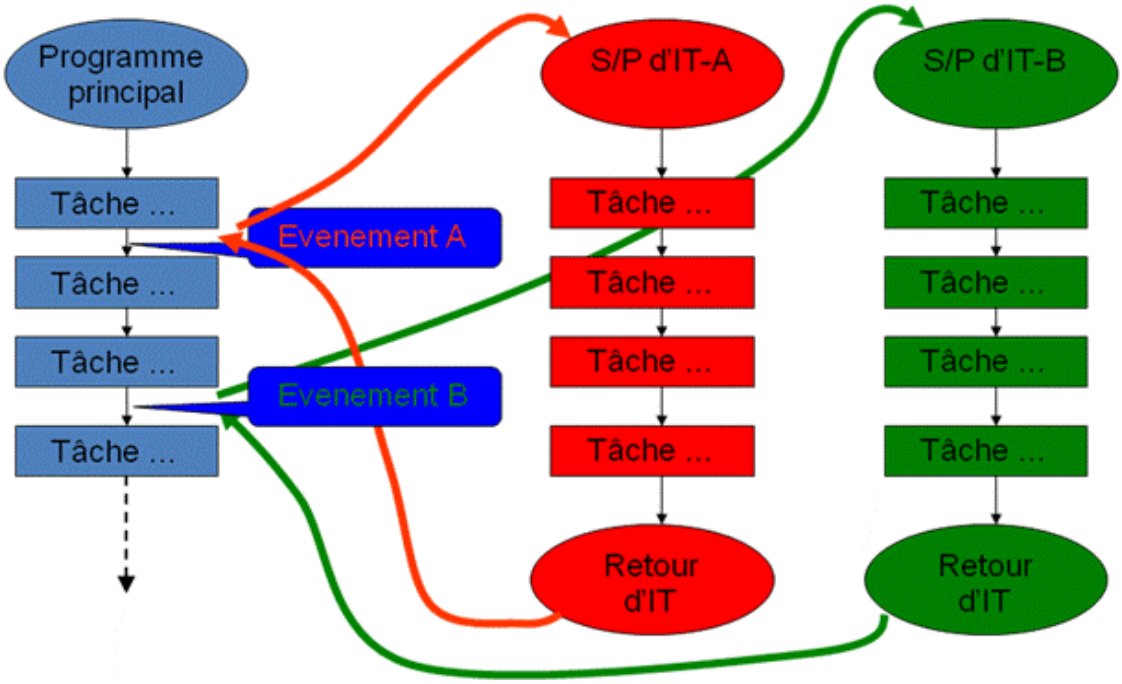
Une première approche consiste à rechercher les tâches à effectuer par scrutation :



On voit que le programme passe beaucoup de temps à surveiller s'il se passe quelque chose, le délai entre l'événement et son traitement n'est pas connu. D'autre part si un événement apparaît durant l'exécution d'une tâche, son traitement en sera d'autant plus retardé et peut même être ignoré.

Une deuxième approche consiste à utiliser le processus d'interruption équipant tous les microcontrôleurs.

Les événements ne sont plus scrutés par programme, mais leur source est connectée à un dispositif **matériel** (interne au microcontrôleur) qui détectera l'événement et interrompra le programme en cours pour effectuer la tâche logicielle correspondante.

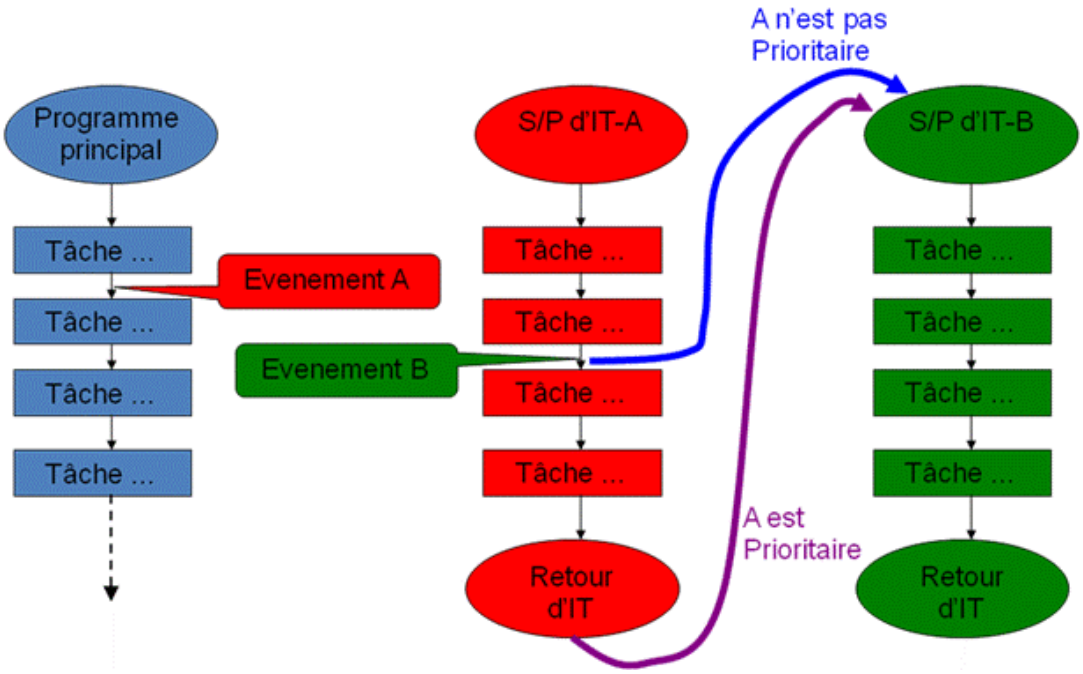


Les avantages de cette technique sont évidents. Le microcontrôleur peut passer son temps à ne rien faire ou traiter des tâches permanentes et non prioritaires, il est sollicité automatiquement lorsqu'une nouvelle tâche est nécessaire. Le temps de réponse à un événement est beaucoup plus rapide et surtout il est connu.

Les contraintes :

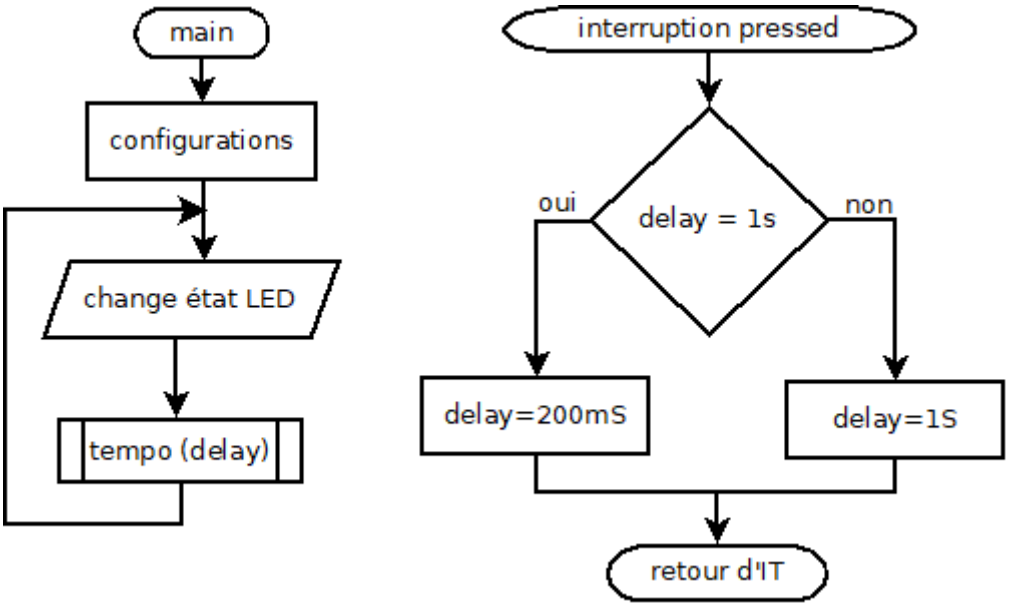
- Gestion automatique des adresses des sous-programmes d'interruption (utilisation de vecteurs)
- Le programme en cours peut être interrompu n'important quand. (un événement physique est asynchrone)
- Les données en cours de traitement avant l'interruption, peuvent être perdues.
- Que se passe t il s'il y a un interruption durant le traitement d'une autre ?

Notion de priorités, soit la tâche en cours est prioritaire sur la nouvelle, soit elle ne l'est pas :



2 Interruptions sur GPIO

Le programme ci dessous réalise en tâche de fond le clignotement d'une LED.
La période est changée lors de l'appui sur un bouton. Il n'y a pas de scrutation du bouton, son appui déclenche l'appel du sous programme d'interruption dans lequel la période est changée.



a) Tester le programme ci-dessous

```
// demo IT sur STM32
#include "mbed.h"

InterruptIn boutonBleu(USER_BUTTON);
DigitalOut maled(LED1);

double delay = 1.0; // 1 sec

void pressed()
{
    if (delay == 1.0)
        delay = 0.2; // 200 ms
    else
        delay = 1.0; // 1 sec
}

int main()
{
    boutonBleu.fall(&pressed);
    while (1)
    {
        maled = !maled;
        wait(delay);
    }
}
```

b) A l'aide de la documentation MBED, commenter la méthode "fall".
c) Que représente l'esperluette dans `boutonBleu.fall(&pressed);`
d) Modifier le programme de manière à changer la période lors du relâchement du bouton
e) Modifier le programme de manière à changer la période lors du relâchement et de l'appui sur le bouton