

TP2: Application d'affichage

1. Introduction:

L'objectif de ce second TP est de bien maîtriser les notions d'alarme, date absolue/relative et les événements OSEK/VDX tout en utilisant les fonctions prédéfinies dans l'API OSEK/VDX tel que :**SetRelAlarm**, **SetAbsAlarm**, **CancelAlarm**, **WaitEvent**, **SetEvent**.

2. Questions:

Question 1.1:

Nous avons commencée par la déclaration des deux taches ainsi que, l'alarme dans le fichier .oil en affectant une priorité plus important à Task1 que Task0, et en activant la Task1 dès le déclenchement d alarme. En utilisant la fonctions prédéfinies dans l'API OSEK/VDX SetAbsAlarm qui déclenchera l'alarme à chaque seconde et systick_get_ms qui affichera le temps systèmes.

```
TASK task0
{
    PRIORITY = 1;
    AUTOSTART = TRUE { APPMODE = std; };
};

TASK task1
{
    PRIORITY = 2;
    AUTOSTART = FALSE;
};

ALARM aT1
{
    //le compteur doit avoir été défini
    COUNTER = SystemCounter;
    ACTION = ACTIVATETASK
    {
        TASK = task1; //la tâche doit avoir été définie
    };
    AUTOSTART = FALSE;
};
```

Question 1.2:

Pour que le déclenchement se fait automatiquement sans avoir besoin de setAbsAlarm il suffit juste ajouter ce bout de code dans la partie ALARM du fichier .oil.

```
AUTOSTART = TRUE
{
    ALARMTIME = 100;
    CYCLETIME = 1000;
```

```
    APPMODE = std;
};
```

Question 1.3:

Vue que le compteur de cycle a une valeur maximale qui ne peut pas dépasser, nous avons pensé à ajouter une variable (next = 0) qui va nous résoudre le problème. Nous avons opté pour la fonction SatAbsAlarm car elle utilise la valeur absolue du compteur.

```
next = next + 1000 % 2000;
SetAbsAlarm(aT1, next , 0);
```

Question 2:

Nous avons commencé par donner la même priorité pour les trois taches qu'on va créer qui vont afficher le temps système (chaque seconde) en utilisant la fonction systick_get_ms(), la valeur du capteur de pression (chaque 3 seconde) en utilisant la fonction ecrobot_get_sound_sensor(1) et la valeur du détecteur de distance (chaque 2 secondes) en utilisant la fonction ecrobot_get_sonar_sensor(2).

```
/****** Declaration d'alarm *****/
ALARM aT1
{
    COUNTER = SystemCounter; // le compteur doit avoir été défini
    /****** Activate Task *****/
    ACTION = ACTIVATETASK
    {
        TASK = task0;
    };
    /****** Started automaticly *****/
    AUTOSTART = TRUE
    {
        ALARMTIME = 100;
        CYCLETIME = 1000;
        APPMODE = std;
    };
};
```

Question 3:

Nous avons commencé par la déclaration de l'événement dans le fichier .oil et après nous avons créé Task3 qui va attendre l'événement déclencher par l'appuie sur le bouton gauche.

```
/****** EVENT *****/
EVENT ev_shutdown{
    MASK = 0x01;
};
TASK task3 {
```

```
.....  
EVENT = ev_shutdown;  
};
```