

Denis Hautot – Lycée Victor-Hugo BESANÇON.

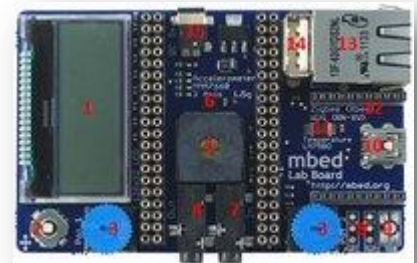
STS SN

Développement à base de cartes de prototypage rapide
mbed

Exercices



Prérequis : langage C et TP1, 2 et 3 mbed.



Ce que vous saurez faire à la fin de ce TP.

Écrire un programme en fonction d'un cahier des charges et utiliser dans ce programme, différentes ressources ajoutées progressivement. (Joystick, LED RGB, CAD, PWM, etc.)

Premier programme.

Nous allons utiliser la LED RGB (Red=p23, Green=p24, Blue=p25) de la carte de test ainsi que le joystick pour la piloter. (Allumer une LED revient à lui appliquer un 0)

Le joystick est équivalent à 5 boutons poussoirs qui seront activés en fonction de sa position. (Up, Down, Left, Right, Center).

Écrivez un programme qui permet de :

Allumer la LED R lorsque le joystick est en position Up.

Allumer la LED G lorsque le joystick est en position Down.

Allumer la LED B lorsque le joystick est en position Left.

Allumer la LED R+G lorsque le joystick est en position Right.

Éteindre les leds lorsque le joystick est relâché.

Vous pourrez utiliser les structures du type : Si condition vraie alors faire instruction

```
If (Up==1)
{
R=0 ;
}
```

Ou bien une structure de choix du type

```
switch(expression)
{
case constante1:
instruction1;
case constante2:
instruction2 ;
instruction2 ;
}
```

Second programme

Nous allons intégrer l'afficheur de la carte de TP dans notre programme afin d'afficher la position du joystick dans notre programme précédent.

Nous allons devoir importer la bibliothèque correspondant au C12832 et réutiliser (copier-coller) certaines déclarations de nos programmes précédents.

Attention : l'affichage ne doit pas scintiller, le passage de l'affichage d'un texte à un autre doit se faire sans que des pixels non désirés ne restent actifs)

Troisième programme

Nous allons cette fois-ci utiliser le potentiomètre pot1 comme actionneur. Ce potentiomètre est connecté sur l'entrée p19 qui est considérée comme entre analogique.

Le programme réalisé devra lire la tension sur p19 (issue de pot1), et afficher après conversion, le résultat sur l'afficheur C12832.

Ce résultat de conversion sera affiché grâce à une ligne d'instruction du type :

```
Lcd.printf("Voici la valeur de i : %d", i);
```

Dans cette instruction, %d sera remplacé par le contenu de la variable i en décimal.

Le caractère de formatage d (%d) peut être remplacé par les caractères de formatage ci-dessous en fonction de l'affichage désiré.

	Type de donnée à afficher	Caractère de formatage
Numériques	Entier décimal signé	d
	Entier décimal non signé	u ou i
	Entier octal non signé	o
	Entier octal non signé	x (avec les caractères 'a' à 'f') ou X (avec les caractères 'A' à 'F')
	Flottants de type double	f, e, g, E ou G
Caractères	Caractère isolé	c
	Chaîne de caractères	s
Pointeurs	Pointeur	p

Quatrième programme

Nous allons maintenant utiliser la fonction PWM.

Repérez sur quelles sorties le PWM peut être utilisé.

Créez un nouveau projet et indiquez que la LED R doit être pilotée en PWM. (Allez regarder dans les classes Mbed pour trouver comment faire).

Recherchez la fonction PWM et paramétrez la fréquence à 1KHz et faites plusieurs essais avec des rapports cycliques (duty cycle) variant de 0.1 à 0.9. (Vous pouvez mesurer l'allure du signal sur p23 afin de vérifier fréquence et rapport cyclique).

Que remarquez-vous ?

Cinquième programme

Nous allons fusionner (3 et 4) nos programmes afin de faire varier la luminosité de la LED R en faisant varier la position de pot1.