

Programmer les entrées-sorties

L'objectif de ce TP est de pratiquer l'usage des entrées-sorties du MSP430 en agissant sur ses registres.

Nous abandonnerons la librairie Arduino (ou ce qui l'a remplacée). Le programme se présentera sous la forme de la procédure `main()`, suivie des initialisations, suivie de la boucle principale `while(1) {...}`.

Le broches voici les broches utilisées pour les poussoirs et les LED : LedRouge : P1.0, LedOrange : P8.1, LedVerte : P8.2, Poussoir 1: P1.7, Poussoir 2: P2.2, Poussoir 3: P4.5

1) Clignotant

Récrivez complètement le programme qui fait clignoter la LED rouge (P1.0).

Partez du programme *incomplet* **base-clign.c**, disponible sur Moodle !

Vous aurez à compléter les définitions **LedRouge** et **LedRougeOff**.

2) Mise en évidence de l'utilité des résistances de tirage

Ajoutez à votre programme le fait que le clignotement ne se produise que lorsque la poussoir 1 (P1.7) est pressé. Dans la boucle principale vous aurez à ajouter `if (Pous1On) {...}`.

Attention : si c'est bien une valeur 1 qui allume les LED, les poussoirs donnent une valeur 0 lorsqu'ils sont pressés. Vous aurez à utiliser l'opérateur d'inversion « ! » du C dans la définition de `Poussoir1On`.

Mais le programme ne va pas fonctionner correctement... La broche du poussoir 1 est bien programmée en entrée, vu que `P1DIR` est initialisé à 0 au moment du Reset. Lorsqu'on presse sur le poussoir, une valeur 0 est bien transmise à la broche P1.7. Mais lorsque le poussoir n'est pas pressé, la broche P1.7 n'est plus connectée. Elle devient une « antenne » qui capte toutes les petites charges statiques qui se présentent. Elle peut détecter tantôt un état 0, tantôt un état 1.

Ajouter l'initialisation de la résistance de tirage vers le haut (*pull-up resistor*) pour la broche P1.7. Deux instructions seront nécessaire !

Le poussoir devrait alors commander correctement le clignotement.

3) Définitions complètes

Ecrivez les définitions complètes pour les poussoir 1 à 3 et pour les LED rouge, orange et verte, ainsi que les initialisations correspondantes dans `main()`.

Ecrivez un programme de test qui allume la LED verte avec le poussoir 1, la LED orange avec le poussoir 2 et la LED rouge avec le poussoir 3.

Ce programme ne doit pas comporter de délai : on souhaite une réaction immédiate sur les LED à la pression (allumer la LED correspondante) et au relâchement (éteindre la LED) des poussoirs.

4) Toggle

Définissez `LedRougeToggle` au moyen de l'opérateur ou-exclusif du C.

Récrivez un programme de clignotement plus simple, dont la boucle principale sera :

- inverse la LED
- attends une demi-seconde.

Bon TP !