

TP Moteur Logidule 2 : rangées de LED, PWM et encodeur

1) Mettez en oeuvre le programme **Va-et-vient.c** . Prenez la version du corrigé, pour avoir les bonnes définitions.

2) **Double rangée de LED** Deux registres 74HC595 sont disponibles sur cette carte. Ils forment un registre de 16 bits, pour commander 16 LED. En voici les définitions :

```
#define HC595Out P3OUT
#define HC595Dir P3DIR
#define bHC595Data 7
#define bHC595Ck 5
#define bHC595Load 6
```

Écrivez d'abord une procédure qui envoie les 16 bits au registre, sous forme de deux nombres de 8 bits (uint8_t). Programmez ensuite une animation du type « chenillard » sur un ces LED.

2) Ajoutez dans la boucle principale while(1) des instructions pour produire le **PWM**.

Indication : mettez dans cette boucle une seule attente de 40us. Donnez au cycle du PWM 256 fois la durée de la boucle.

Récrivez le programme **va-et-vient** à vitesse 50%.

3) Ajoutez la lecture de la position avec **l'encodeur**. Programmez l'algorithme :

```
« si un des bits a changé
  si X est égal à ancienY on a avancé
  sinon on a reculé »
```

Affichez la variable Position sur une des rangées de LED de la carte. Les tests peuvent être faits à vitesse nulle, en tournant le moteur à la main et en observant les LED.

4) Ecrivez un programme qui effectue les **mouvements** suivants lorsqu'on presse sur le bouton-poussoir :

- une avance d'une seconde à vitesse maximale pour un éventuel dégagement de la fin de course
- un recul à vitesse 25% pour la recherche de la fin de course
- une avance d'un tour de la roue dentée à vitesse 50%
- une oscillation de 5 dents dans un sens puis dans l'autre en permanence
- reprise de tout le programme lorsqu'on presse sur le poussoir.

Conseil : pour rendre votre programme plus lisible, placez les instructions qui gèrent le PWM dans une procédure GerePWM() et les instructions qui gèrent la lecture de l'encodeur dans GerePosition().