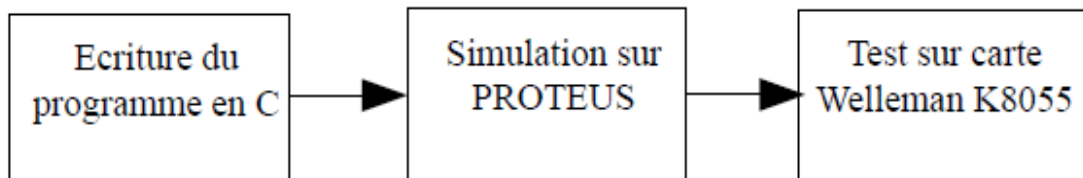


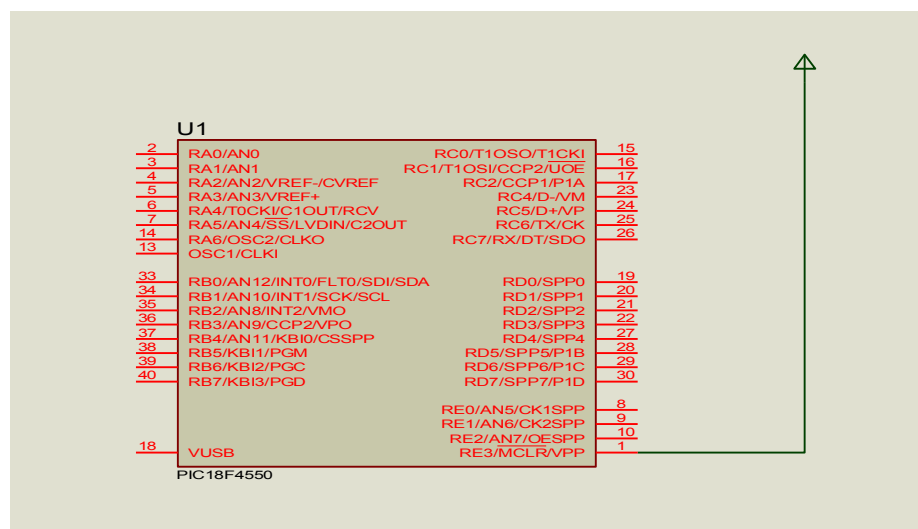
Lors des Tds et Tps de systèmes embarqués, vous allez apprendre à développer des applications sur microprocesseurs.

La chaîne de développement est la suivante :



Le microcontrôleur utilisé est le PIC 18F4550 de la société MICROCHIP.

**Q1 :** Test d'un programme. Ouvrir le projet TD1 sous MPLAB et le construire (build). Le fichier obtenu pour les test est un fichier du type « HEX ». Ouvrir PROTEUS et saisir le schéma suivant :



Dans les propriétés du microcontrôleur :

- désactiver le « chien de garde » (Enable WatchDog timer : No)
- fixer la fréquence à 4 MHz
- charger le programme de test

Lancer la simulation et décrire le fonctionnement du programme.

**Q2:** Modifier le programme pour avoir une fréquence de fonctionnement de 1s précise. Étudier la documentation de la fonction Delay10KTCYx() (voir dans C:\MCC18\doc l'aide hlpC18Lib). Pour le réglage précis, on doit faire une simulation directement sous MPLAB :

Debugger → Select Tool → MPLAB SIM

Debugger → Settings... → 4 MHz

Debugger → StopWatch

A l'aide du fonctionnement Pas à Pas, on est capable d'ajuster le temps d'exécution avec la fenêtre StopWatch

A quoi sert l'instruction « TRISB »

**Q3 :** Tester maintenant votre programme sur la carte VELLEMAN. Pour faire cela :

- brancher la carte VELLEMAN sur un port USB du PC
- démarrer HIDBootLoader (dans le répertoire APPLI)
- ouvrir le fichier HEX du projet et programmer la carte
- démarrer l'application sur la carte (débrancher le port USB et rebrancher en maintenant le bouton **inpl** enfoncé).

**Q4 :** En utilisant les fonctions de gestions du temps étudiées en Q2, écrire la fonction :

void Delayms(int time) ou on passe le temps d'attente en ms.

Tester cette fonction

**Q5:** Changer le fichier source du projet (td1-p2.c). Le logiciel Proteus nous permet de réaliser une co-simulation matériel/logiciel. Pour ceci, changer l'outil de simulation dans MPLAB (Proteus VSM vs MPLAB SIM) . Une fenêtre avec votre schéma apparaît (après l'avoir chargé!). Mettre un point d'arrêt dans le main et lancer la simulation. Vous pouvez déboguer le programme en pas à pas

**Q6 :** On souhaite modifier l'application pour faire varier le sens de défilement du chenillard en fonction d'un bouton poussoir. Modifier le schéma de PROTEUS (tenir compte de la compatibilité avec la carte VELLEMAN). Écrire le programme, mettre au point sur PROTEUS, et tester au final sur la carte. On utilisera le composant LOGICSTATE pour simuler le bouton poussoir.

**Q7 :** Application complète. On souhaite utiliser deux boutons poussoir :

- un pour changer le sens de défilement
- un pour changer la vitesse de défilement (deux possibles).

**Q8 :** Changer le fichier source du projet (td1-p3.c) et utiliser le schéma td1-p3.dsn sous PROTEUS pour la simulation. A l'aide du source et du débogage, expliquer le fonctionnement de la gestion d'un clavier matricé.

**Q9 :** On souhaite utiliser un clavier avec 4 colonnes (KEYPAD-SMALLCALC). Faire le câblage et modifier la fonction `gest_clav()` en conséquence.