

TP3: Simulation et programmation de PIC

Introduction Dans ce TP, vous mettrez en œuvre un microcontrôleur PIC à travers 3 applications. Vous saisissez le programme sous forme d'algorithme via *FlowCode* pour ensuite le compiler en "fichier objet", qui pourra être utilisé par le simulateur pour émuler le fonctionnement du PIC.

1 Afficheur LCD

Saisir le schéma de la fig. 1 sous ISIS et le sauvegarder avec le nom **T:\M1103\TP3\TP3_1.dsn**

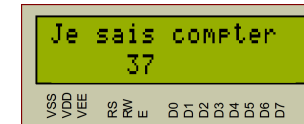
1.1 Utilisation élémentaire de Flowcode

1. Démarrer un projet Flowcode et y insérer une boucle infinie.
2. Insérer le composant « LCDDisplay » : le composant apparaît sur la fenêtre «panneau». Consulter l'aide sur ce composant : menu "Aide" et y chercher «LCD» : la description complète de l'utilisation est donnée.
3. Dans l'algorithme, insérer un bloc «Manipulation caractères» : le sélectionner, propriétés, puis créer une variable **ch1** de type chaîne, et lui donner une valeur : **ch1="Bonjour"**
4. Insérer à la suite deux blocs "Manipulation de routine composant". Dans le premier, faire un appel à la fonction d'initialisation de l'afficheur. Dans le second, appeler la fonction d'affichage de chaîne de caractère et lui dire d'afficher **ch1**.
5. Vérifier le fonctionnement sous Flowcode (bouton "Exécuter", F5).

1.2 Transfert du programme et simulation sous ISIS

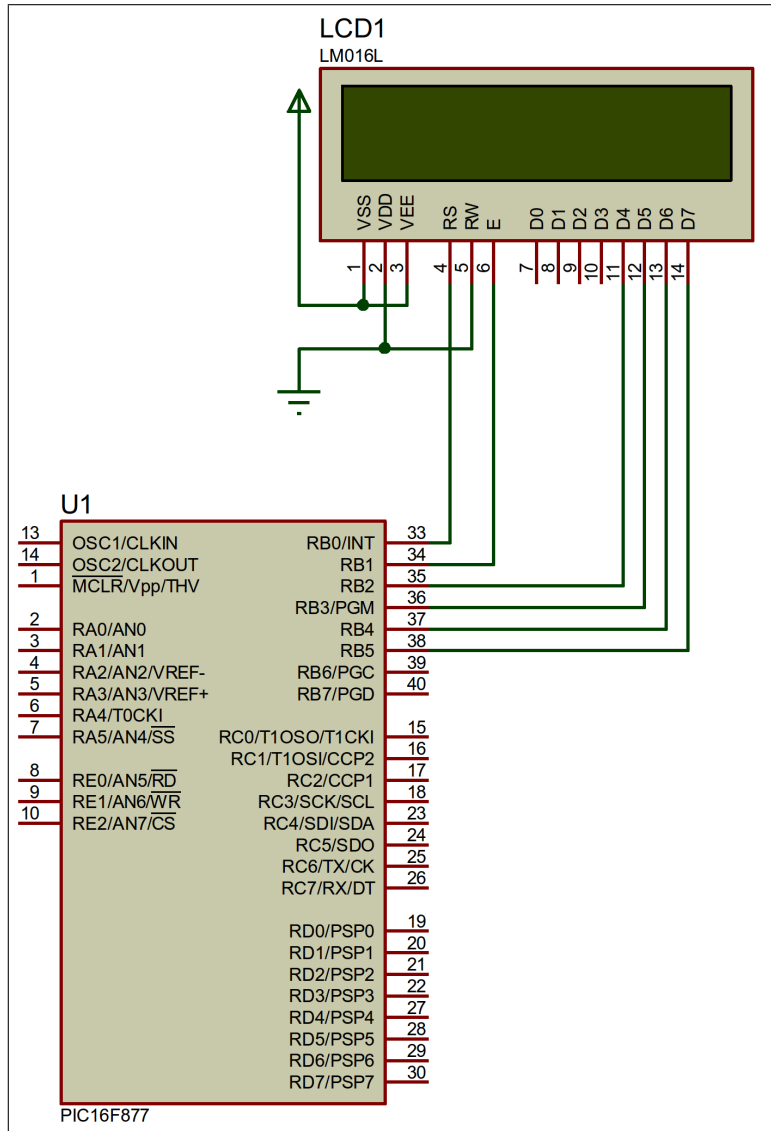
1. Configurer les connexions de façon à respecter le schéma électronique : dans la fenêtre "panneau", clic-droit sur l'afficheur LCD et "Connexions". Vous devez configurer les connexions de broches conformément au schéma (attention : "validation" est nommée "E" sur le schéma).

2. Compiler le projet vers le format "HEX" en **vérifiant** qu'il n'y a pas d'erreur (menu "Puce → compiler en .HEX").
Vérifier via l'explorateur Windows que le fichier correspondant est bien créé dans votre dossier de travail. Ouvrir ce fichier avec Notepad++ : vous observez les **codes opératoires** (exprimés en hexadécimal) du programme compilé.
3. Sous Isis, associer au microcontrôleur PIC le fichier programme généré, et lancer la simulation : vous devez observer la chaîne sur l'afficheur.
4. De façon similaire, créer un nouveau projet sous Flowcode qui devra produire l'affichage suivant sous ISIS, avec la valeur qui s'incrémente en continu. Faire valider par l'enseignant.



5. Ajouter un LOGICSTATE connecté sur l'entrée **RA0** du PIC et modifier le projet pour que la valeur affichée s'incrémente ou se décrémente suivant la valeur de cette entrée.
Vous utiliserez un bloc "Entrée". Il faut le configurer pour lire le bit 0 du PortA, puis affecter sa valeur à une variable de type "entier". On peut ensuite tester la valeur de cette variable avec un bloc "Décision", générant les deux branches, l'une où on va incrémenter ($v = v + 1$) et l'autre où on va décrémente ($v = v - 1$).
Tout le reste du code (placement du curseur, affichage de la chaîne) **reste** dans la partie commune de la boucle infinie.
6. Implémenter la fonctionnalité suivante : Quand on atteint la valeur 100, décrémente le compteur de façon à arriver à 0.

FIGURE 1 – Schéma de mise en œuvre du LCD

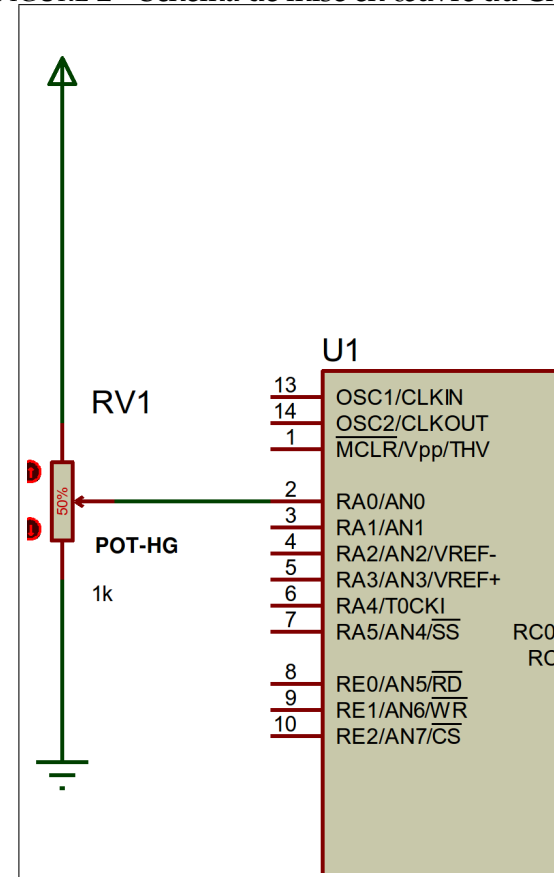


2 Convertisseur Analogique Numérique (CAN)

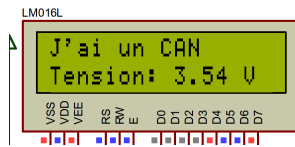
Copier le schéma précédent avec le nom **T:\M1103\TP3_2.dsn** et y ajouter un potentiomètre connecté sur une des entrées analogiques, conformément au schéma de la fig. 2. Ceci permettra avec le simulateur de faire varier la tension d'entrée entre 0 et 5V.

Le but de ce TP est de réaliser un Voltmètre (très sommaire) à l'aide du CAN intégré sur le PIC et de notre afficheur LCD.

FIGURE 2 – Schéma de mise en œuvre du CAN



1. Sous FlowCode, insérer le composant « ADC » (consulter l'aide sur ce composant).
2. Configurer les connexions de façon à respecter le schéma de la fig. 2
3. Réaliser et simuler un programme permettant l'affichage de la valeur de la tension (on utilisera une variable de type chaîne de façon à simplifier les conversions).



3 Communication RS232

Dans cette partie, on mettra en œuvre une transmission de données en série depuis un microcontrôleur PIC vers un « terminal », qui est un simple récepteur d'octet : il affiche sur l'écran le code ASCII de chaque octet reçu sur son entrée RX (la broche TX est la sortie, non utilisée ici). La communication avec un tel terminal se fait via une **liaison série**, ou les bits de chaque octet sont envoyés en séquence.

D'un point de vue électrique, les niveaux de tensions et le débit binaire de telles liaisons sont normalisés. Le terminal attend des niveaux de tensions à la norme « RS232 » ('0' : -12V, '1' : +12V). Le microcontrôleur générant sur ses sorties des niveaux 0/5V, on doit insérer un composant adaptateur (MAX232) qui réalise cette conversion de niveau.

Le terminal sera simulé sous ISIS par le composant « Virtual terminal » (barre d'outil → "mode instruments virtuels").

Le but de ce TP est de transmettre une trame série RS232 représentant la valeur d'une tension mesurée via le convertisseur analogique-numérique (ADC) du PIC. La tension sera générée par un simple potentiomètre. Il faudrait faire afficher la valeur de la tension sur le terminal.

1. Sous ISIS, saisissez le schéma de la figure 3 et configurer correctement le terminal. Réglez l'horloge μC à 20 MHz.

2. Sous Flowcode, saisissez l'algorithme correspondant au pseudo-code ci-dessous. Il faut ajouter les composants «ADC» et la liaison RS232 sur le panneau, ainsi que les appels de fonction correspondants.
3. Le compiler en .hex et importer le programme dans le PIC sous ISIS. Validez le fonctionnement.

```

Debut
Envoyer les caracteres "TEST LIAISON RS232"
Tant que 1 faire
    Envoyer le caractere 13
    Envoyer le caractere '>'
    Convertir la valeur du CAN dans une chaîne de
    caractères
    Envoyer la chaîne de caractère
Fin
  
```

Note : l'octet de valeur 13 correspond au code ASCII CR (retour chariot)

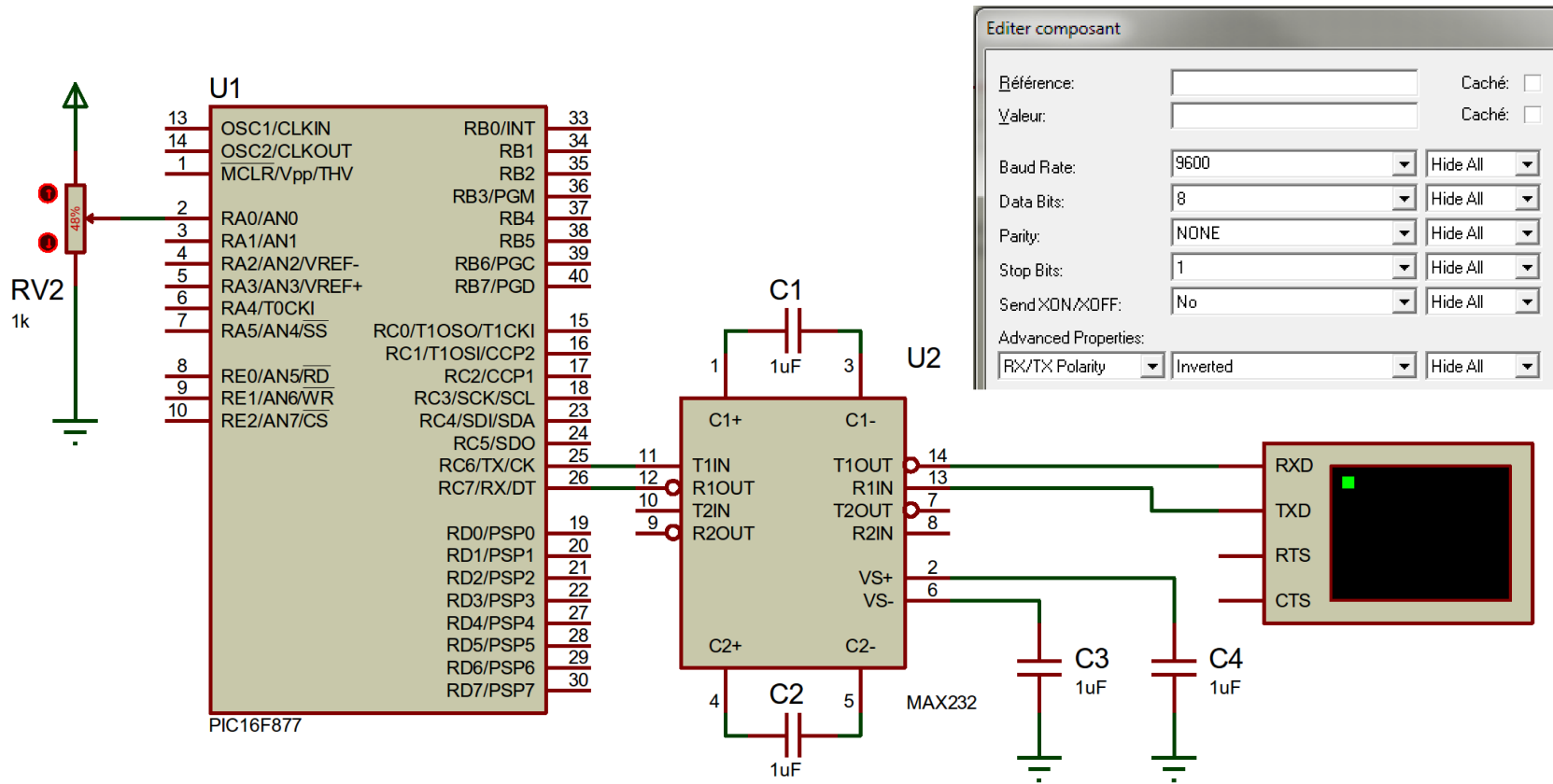


FIGURE 3 – Liaison RS232