

Formation Automatique et Informatique Industrielle

Master 1 S2

Matière : Systèmes Embarqués et Systèmes
Temps Réel SE-STR

Par : ATOUI Hamza

Plan de TD-TP

- LAB1 (rappel de cours).
- LAB2 (rappel de cours).
- LAB3 (rappel de cours).
- LAB4 (rappel de cours).
- LAB5.
- LAB6.

LAB1 (rappel de cours)

- LAB1 : rappel de cours sur l'ADC.
- Par votre conception, faire le montage et écrire le programme qui assure la lecture de l'état d'un potentiomètre relié à une des entrées analogiques et afficher la valeur convertie en numérique par l'ADC sur un des ports de 8 bits (ignorer les deux bits du poids faible de résultat de l'ADC) par une fréquence de scrutation de 10Hz.

LAB2 (rappel de cours)

- LAB2 : rappel de cours sur l'ADC et le TIMER2.
- Garder le même montage du LAB1 et changer le programme par le suivant:
 - Version1: la scrutation se fait par le TIMER2 (10Hz).
 - Version2: la scrutation se fait par le TIMER2 (10Hz) et TIMER2 l'ADC, les deux fonctionnent sous interruption.

LAB3 (rappel de cours)

- LAB3 : rappel de cours sur le module CCP en mode PWM.
- Par votre conception, faire le montage et écrire le programme qui assure la commande de variation de la vitesse d'un moteur DC par le biais de signal PWM et un clavier composé de 3 boutons {SPEED_UP, SPEED_DOWN, STOP}.
 - La présence d'une impulsion sur le bouton SPEED_UP augmente la vitesse de 5%.
 - La présence d'une impulsion sur le bouton SPEED_DOWN diminue la vitesse de 5%.
 - La présence d'une impulsion sur le bouton STOP provoque l'arrêt immédiat du moteur.

LAB4 (rappel de cours)

- LAB4: rappel de cours sur le module USART.
- Dans ce LAB on essaye de réaliser un ECHO232 qui présente un système renvoi la valeur transmit par un Terminal vers le Terminal lui-même à travers le port série COMPORT.
- Le débit : 300 baud.
- Format : 8 bits DATA, 1 bit STOP.
- Utiliser MATLAB pour réaliser la partie soft de Terminal.

LAB5.

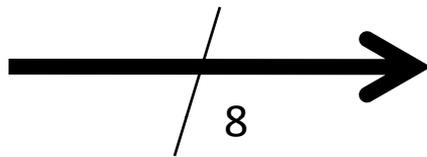
- Configurer L'USART en **300 baud, 8 bits data et 1 bit stop.**
- Envoyer au Terminal le message « **WELCOME** »
- Envoyer au Terminal votre **nom** et **prénom.**
- Boucler infiniment sur le traitement suivant:
 - Si le Terminal envoi le caractère '**L**' la **LED (RB5)** est à l'état **OFF.**
 - Si le Terminal envoi le caractère '**H**' la **LED (RB5)** est à l'état **ON.**
- Utiliser MATLAB pour réaliser la partie soft de Terminal.

LAB6

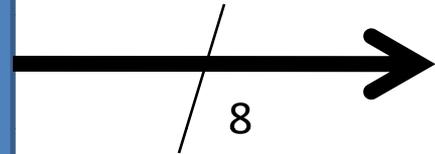
- Dans ce LAB, on va transformer une carte à base de PIC16F877A à une carte d'acquisition et de contrôle-commande.
- La figure suivante présente la carte qu'on va réaliser :
 - 8 entrées numériques.
 - 8 sorties numériques.
 - 8 entrées analogiques (précision 8 bits).
 - 2 sorties PWM (PWM1 et PWM2) [$F_{\text{PWM}} = 4\text{KHz}$].
 - Communication RS232 avec un Terminal (8 DATA, 1 STOP, 9600 baud).
 - Quartz = 4MHz.

LAB6

8 entrées analogiques



8 sorties numériques



PWM1

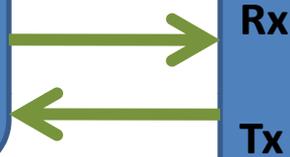
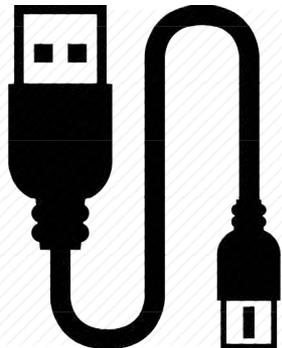
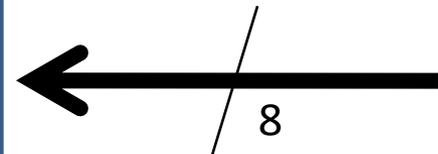


PWM2



PIC16F877A

8 entrées numériques



Rx

Tx

LAB6

- Pour assurer la communication avec cette carte, on propose de réaliser le protocole suivant:
- Pour lire une entrée numérique :
 - Commande : **['I', pin, '\$']** avec pin : {'0', '1',... '7'}.
 - Réponse : l'état de la pin soit 'L' ou 'H' [L \Leftrightarrow 0, H \Leftrightarrow 1]
- Pour écrire une sortie numérique :
 - Commande: **['O', pin, value, '\$']** avec pin : {'0', '1',... '7'} et value {'L', 'H'}.
 - Réponse : rien.
- Pour lire une entrée analogique :
 - Commande: **['A', pin, '\$']** avec pin : {'0', '1',... '7'}.
 - Réponse : valeur numérique sur 8 bits.
- Pour écrire une sortie PWM :
 - Commande: **['P', pin, DC, '\$']** avec pin : {'1', '2'}, DC entre 0 et 100.
 - Réponse : rien.

LAB6

- Ecrire le programme qui assure le bon fonctionnement du protocole de communication entre la carte à PIC et le Terminal.
- Utiliser MATLAB pour réaliser la partie soft de Terminal.
- Utiliser MATLAB pour réaliser une interface graphique pour communiquer avec la carte.