

Dates remise compte rendu :

Groupe A : Mercredi 11 décembre 2024

Groupe B : Jeudi 12 décembre 2024

Quartz du pic : 4Mhz Démonstration : en salle de TP

Il faut traiter chaque question de l'exercice

L'écriture du programme et sa simulation est obligatoire

Pour chaque programme tracer l'organigramme

Projet 1

Un local est surveillé avec 4 capteurs identiques (bouton poussoir BP). Au repos la sortie de BP est à l'état haut. A chaque fermeture de BP une interruption est déclenchée, et la led correspondante est allumée durant 10s. La sortie de chaque BP est mise en forme par une porte.

- 1) Faire le choix de la led et calculer la valeur pratique de la résistance R de protection.
- 2) Calculer la valeur N du compteur de boucle pour la timer TMR0.
- 3) Tracer le schéma à base du pic16f84
- 4) Ecrire le programme assembleur et faire la simulation avec PROTEUS.

Projet 2

Le pic 16f84 commande un moteur à courant continu (cc) 12V à l'aide d'un transistor de puissance. Un commutateur K permet la mise en marche du moteur ou son arrêt. K = 1 moteur tourne et led verte allumée, K = 0 moteur arrêté et led rouge allumée.

- 1) Faire le choix des led et calculer la valeur pratique de la résistance de protection
- 2) Tracer le schéma d'interface du moteur
- 3) Ecrire le programme assembleur et faire la simulation avec PROTEUS
- 4) Proposer une solution pour faire tourner le moteur dans les deux sens.

Projet 3

Deux boutons poussoirs sont utilisés pour incrémenter (0 à 9) ou décrémenter (9 à 0) la variable var. Cette variable est affichée sur un afficheur 7-segments. Le bouton poussoir qui incrémente est Inc, et celui qui décrémente est Dec. Lorsqu'on actionne Inc, la variable est incrémentée, si on appuie sur Dec elle est décrémentée. Lorsque var est incrémentée, elle revient à 0 après la valeur 9, et lorsqu'elle est décrémentée elle revient à 9 après le 0. Il y a mise en forme des boutons poussoirs. Etat initial : var = 0.

- 1) Faire le choix du système d'affichage et déterminer la valeur pratique des résistances de protection
- 2) Tracer le circuit d'interface
- 3) Ecrire le programme assembleur et exécuter la simulation avec PROTEUS.

Projet 4

Un système à base du pic 16f84 simule la gestion d'un parking pour véhicules, de 4 places. Quatre interrupteurs K1 à K4 simule l'état du parking : $K_i = 0$ place libre, $K_i = 1$ place occupée. Pour chaque interrupteur une led verte est allumée si la place est libre, sinon une led rouge est allumée.

- 1) Faire le choix des leds, et calculer la valeur pratique des résistances de protection.
- 2) Tracer le circuit d'interface
- 3) Ecrire le programme assembleur et faire la simulation avec PROTEUS.

Projet 5

Le microcontrôleur 16f84 est utilisé pour le comptage d'évènements externes. Une variable est incrémentée à chaque évènement détecté. Lorsqu'on atteint N, la variable est remise à zéro au prochain évènement. La valeur de N est stockée à l'adresse 0x10. Cette variable est affichée sur deux afficheurs 7-segments à l'aide de décodeur BCD-7segments.

- 1) Faire le choix du décodeur et des afficheurs, déterminer la valeur pratique des résistances de protection.
- 2) Tracer le circuit d'interface
- 3) Ecrire le programme assembleur et simuler l'exécution avec PROTEUS

Projet 6

Deux boutons poussoirs BP4 et BP7 sont connectés respectivement aux lignes de port RB4 et RB7. Une action sur l'un des boutons poussoirs déclenche une interruption, qui allume la led correspondante : verte pour BP4 et rouge pour BP7. L'allumage de la led dure 10s. Le timer0 est utilisé.

- 1) Faire le choix des leds et des valeurs pratiques des résistances de protection.
- 2) Tracer le circuit d'interface avec une mise en forme des impulsions des boutons poussoirs.
- 3) Ecrire le programme assembleur et faire la simulation avec PROTEUS.

Projet 7

La ligne de port RB0 est une entrée d'interruption. Un bouton poussoir déclenche une interruption sur RB0 à chaque action. La routine d'interruption incrémente un compteur qui est affiché sur des afficheurs 7-segments. Après 20 actions le compteur est remis à zéro.

- 1) Faire le choix des afficheurs et déterminer la valeur pratique des résistances de protection.
- 2) Tracer le circuit d'interface avec circuit de mise en forme du bouton poussoir.
- 3) Ecrire le programme assembleur et faire la simulation avec PROTEUS.

Projet 8

Le PORTB commande 8 leds. Un commutateur K commande l'allumage des leds comme suit : $K = 0$ allumage des leds de RB0 à RB7, $K = 1$ allumage des leds de RB7 à RB0. L'allumage d'une led dure 5s. La temporisation est exécutée par le timer du pic 16f84.

- 1) Faire le choix des leds et déterminer la valeur pratique des résistances de protection.
- 2) Tracer le circuit d'interface
- 3) Ecrire le programme assembleur et faire la simulation avec PROTEUS.

Projet 9

On reçoit sur le PORTA (RA0 à RA3) un code sur 4 bits (0 à F). Cette valeur est affichée sur un afficheur 7-segments durant 10s avant l'affichage d'une nouvelle valeur.

- 1) Faire le choix de l'afficheur et calculer N compteur de boucles du timer
- 2) Tracer le circuit d'interface et calculer la valeur pratique des résistances de protection.
- 3) Ecrire le programme assembleur et faire la simulation avec PROTEUS

Projet 10

Le PORTB commande 4 afficheurs 7-segments. On veut réaliser un affichage par défilement. Le chiffre 1 est affiché sur les 4 afficheurs à tour de rôle. La temporisation de 5s de l'affichage est exécutée par le timer0.

- 1) Faire le choix des afficheurs et déterminer la valeur pratique des résistances de protection.
- 2) Tracer le circuit d'interfaçage et calculer le compteur de boucles N du timer.
- 3) Ecrire le programme assembleur et faire la simulation avec PROTEUS

Projet 11

Sur le PORTA (lignes RA0 à RA3) on reçoit 2 variables Xvar et Yvar : de 0 à 9. On additionne ces deux variables et le résultat est affiché en décimale sur deux chiffres : diz. et unité. Le PORTB commande les 2 afficheurs. On utilise un décodage BCD-7 segment

- 1) Faire le choix des afficheurs et déterminer la valeur pratique des résistances de protection
- 2) Tracer le schéma d'interface
- 3) Ecrire le programme assembleur et faire la simulation avec PROTEUS

Projet 12

Des pièces fabriquées passent devant un contact (capteur) qui se ferme à chaque passage d'une pièce, ce capteur est connecté au pic 16f84. Le microcontrôleur détecte le front montant à chaque fermeture du capteur. Le registre TMR0 sert de compteur, et à chaque débordement une interruption est déclenchée, qui incrémente la variable VAR1. En fin de fabrication, le contenu de TMR0 est sauvegardé dans VAR2. Lorsque le microcontrôleur compte les pièces une led verte est allumée, sinon c'est la led rouge qui s'allume. Un bouton poussoir BP lance le comptage.

- 1- Si on appelle A le contenu de VAR1 et B celui de VAR2, donner la formule pour calculer N le nombre de pièces produites.
- 2- Tracer le schéma d'interface
- 3- Ecrire le programme assembleur et faire la simulation avec PROTEUS.

Projet 13

Des variables notées X (0 à 9) sont lues sur le PORTA. Chaque variable X est traité comme suit : $Y = -4 * X + 48$; la valeur Y est stockée en RAM dans un tableau. On donne : adresse tableau : 0x0C, taille tableau : N = 10. Le résultat Y est affiché sur des afficheurs 7-segments.

- 1- Tracer le schéma d'interface des afficheurs et de l'entrée X
- 2- Ecrire le programme assembleur et faire la simulation avec PROTEUS

Projet 14

Le PORTB commande 8 leds qui s'allument à tour de rôle en commençant par RB0. La led s'allume s'il y a action d'un bouton poussoir BP branché sur la ligne RA0. L'allumage de la led est réalisé en envoyant le contenu de la variable output sur le PORTB après sa rotation d'un bit. Les leds ont les anodes portées au +5V.

1-Tracer le schéma d'interface et calculer la valeur des résistances de protection des leds.

2-Ecrire le programme assembleur et faire la simulation avec PROTEUS

Projet 15

Un bouton poussoir BP sert d'entrée de comptage pour le registre TMR0 ; à chaque action sur BP le registre TMR0 est incrémenté. Selon la valeur de ce compteur on exécute l'une des actions :

$0 \leq \text{TMR0} < 101$: led RB5 allumée

$100 < \text{TMR0} < 201$: led RB6 allumée

$200 < \text{TMR0} < 256$: led RB7 allumée

Les leds de couleur différente ont les anodes portées au +5V.

1-A l'impulsion 256 quel est la led qui doit s'allumer ?

2-Les actions sur BP sont remplacées par des impulsions d'interruption sur l'entrée RB0/INT. Expliquer en deux lignes comment on compte ces impulsions sans l'emploi du registre timer TMR0.

3-Tracer le schéma d'interface

4- Ecrire le programme assembleur et simuler ce programme avec PROTEUS

Projet 16

Le registre TMR0 du pic 16f84 est configuré en compteur pour le comptage de 650 évènements.

1- On pose $y = 650$, mettre y sous la forme : $y = a.x + b$ et identifier a, x et b ($a=$, $x=$, $b=$) : ces 3 paramètres sont affichés séparément.

2- Tracer le schéma d'interface avec des afficheurs 7-segments.

3- Ecrire le programme assembleur et simuler ce programme avec PROTEUS.

Projet 17

Le registre TMR0 du pic 16f84 est configuré pour fonctionner en compteur d'évènements. Le système doit compter X pièces ($X < 100$) ; cette valeur X est mémorisée en RAM à l'adresse 0x20. A la fin de ce comptage le programme recommence un autre cycle de comptage.

1- On pose N la valeur de pré-chargement du registre timer TMR0, donner l'expression pour le calcul de N.

1-Ecrire les instructions qui chargent TMR0 avec la valeur N

2-Tracer le circuit d'interface pour l'affichage de X.

3-Ecrire le programme assembleur et simuler ce programme avec PROTEUS