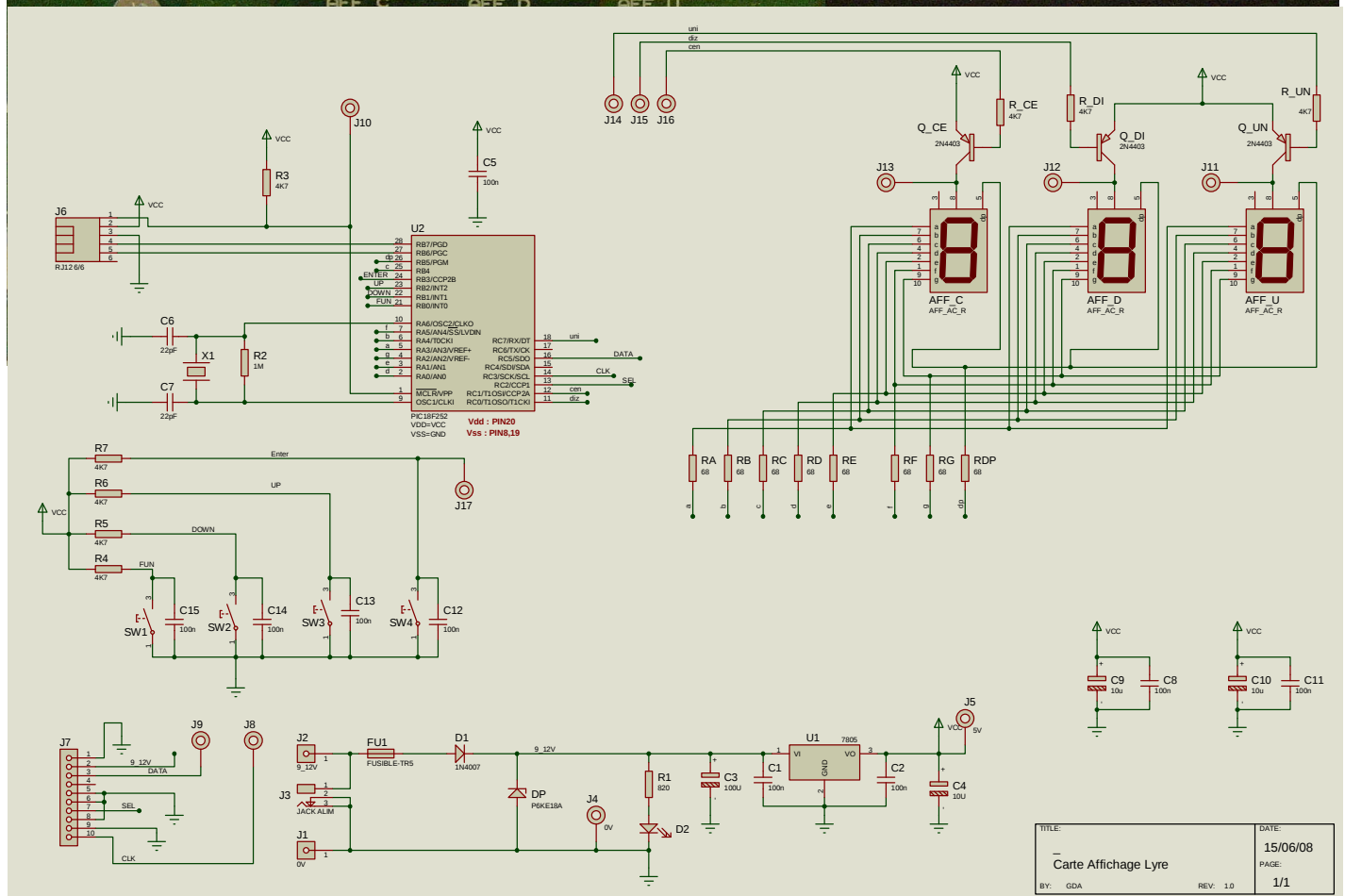


Objectifs :

- Perfectionnement sur l'affichage multiplexé avec afficheur 7 segments.
- Voir le dimensionnement des étages à transistor fonctionnant en tout ou rien.
- Initiation à la transmission série.
- Acquérir la méthode pour dimensionner un dissipateur thermique.
- Continuer à prendre en main le logiciel MPLAB.
- Perfectionnement sur l'initiation à la programmation en C.



Schéma structurel de la carte :



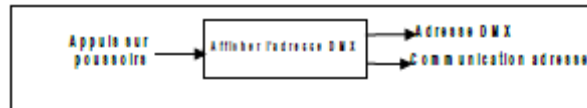
Analyse fonctionnelle

1. Tracez sur le schéma structurel le contour des fonctions principales et secondaires.



13.5 La carte Affichage adresse de la lyre

Rôle : afficher l'adresse DMX de la lyre, la modifier et la mémoriser. Transmettre cette adresse à la carte moteur.

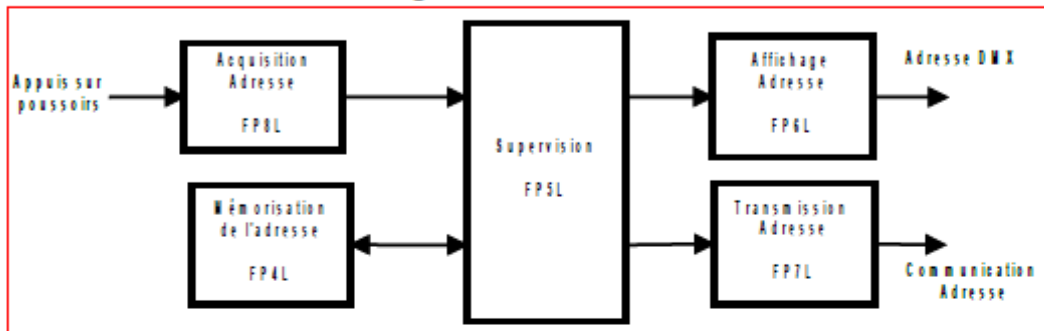


Appuis sur poussoir : 4 poussoirs permettent le changement d'adresse par incrémentation ou décrémentation, ainsi que sa validation.

Adresse DMX : adresse du récepteur DMX sur 3 digits (de 001 à 512)

Communication SPI : les 3 variables sont transmises successivement de manière série synchrone.

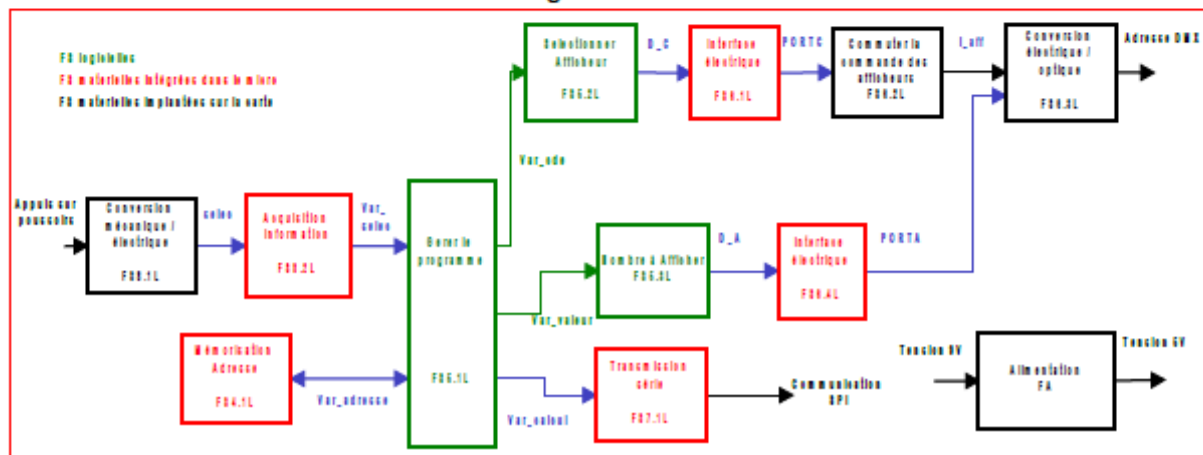
13.5.1 Schéma fonctionnel de degré 1



13.5.2 Description des fonctions principales

- **FP4L - Mémorisation de l'adresse :** permet de stocker en mémoire l'adresse du récepteur. L'adresse doit rester présente après une mise hors tension de l'appareil.
- **FP5L - Supervision :** cette fonction assure la gestion du fonctionnement de la carte « affichage de l'adressage DMX lyre ».
- **FP6L - Affichage adresse :** permet de lire l'adresse DMX dans une situation diurne et nocturne.
- **FP7L - Transmission adresse :** cette fonction assure la transmission de l'adresse DMX à la carte « moteur micro pas ».
- **FP8L - Acquisition adresse :** permet de modifier et de fixer l'adresse DMX du récepteur.

13.5.3 Schéma fonctionnel de degré 2



13.5.4 Description des fonctions secondaires

- **FS4.1L Mémorisation adresse**

Elle stocke l'adresse DMX sur 3 octets. Chaque octet représente une puissance de 10. Cette mémoire est une EEPROM.

Var_adresse : variable contenant l'adresse DMX

- **FS5.1L Gérer le programme**

Cette fonction utilise les nombreuses ressources du microcontrôleur afin de piloter les différentes fonctions FS5.2L, FS5.3L, FS7.1L et FS4.1L en fonction de l'état de FS8.2L.

- **FS5.2L Sélectionner afficheur**

Cette fonction permet de sélectionner 1 afficheur parmi les 3 présents sur la carte. Elle est cadencée par un timer et permet le multiplexage temporel.

Var_cde : variable logicielle à l'image des afficheurs à piloter.

- **FS5.3L Nombre à afficher**

Elle permet de présenter les états binaires sur 7 sorties binaires afin d'afficher les nombres allant de 0 à 9.

Var_Valeur : variable logicielle représentant les segments à allumer pour l'afficheur sélectionné.

- **FS6.1L Interface électrique**

Ports du microcontrôleur permettant d'assurer une compatibilité de niveau électrique en fonction des variables logicielles présentes en entrée.

D_C : commande binaire du PORT C

PORTC : niveaux électriques à l'image de D_C.

- **FS6.2L Commuter la commande des afficheurs**

Organe permettant la commande en courant de l'ensemble des leds composant un afficheur

I_aff : courant de commande des différents afficheurs.

- **FS6.3L Conversion électrique optique**

Ce sont des afficheurs (3) restituant l'adresse DMX allant de 001 à 512

Adresse DMX : adresse du récepteur DMX sur 3 digits (de 001 à 512)

- **FS6.4L Interface électrique**

Ports du microcontrôleur permettant d'assurer une compatibilité de niveau électrique en fonction des variables logicielles présentes en entrée.

D_A : commande binaire du PORT A

PORTA : niveaux électriques à l'image de D_A.

- **FS7.1L Transmission série**

Communication des 3 octets représentant l'adresse DMX sous forme série à la carte moteur.

Var_calcul : variables contenant l'adresse DMX. Chacune de ces variables contient une puissance de 10 de l'adresse DMX.

Communication SPI : Les 3 variables sont transmises successivement de manière série synchrone.

- **FS8.1L Conversion mécanique / électrique :**

Elle permet d'obtenir une information électrique à l'image de l'appui sur un bouton poussoir.

Appuis sur poussoir : 4 poussoirs permettent le changement d'adresse par incrémentation ou décrémentation et sa validation.

Selec : états électriques (1 ou 0) à l'image de l'appui sur l'un des poussoirs.

- **FS8.2L Acquisition information**

Elle permet d'obtenir l'information d'appui accessible par le logiciel.

Var_Selec : variables logicielles à l'image de l'appui sur l'un des poussoirs accessibles par FS8.1L.



ANALYSE STRUCTURELLE

Identification des interfaces d'entrées / sorties utilisées sur le PIC.

1. Faire un tableau indiquant pour chacun des bits de PORT utilisé sur le PIC son mode de fonctionnement (*entrée ou sortie*) et sa fonctionnalité sur la carte.
2. Identifier sur le schéma structurel le bit du microcontrôleur qui permet de détecter l'appui sur le bouton poussoir SW1.
3. Quel est le niveau logique lu par le microcontrôleur lorsque SW1 n'est pas enfoncé?
4. Quel est le niveau logique lu par le microcontrôleur lorsque SW1 est enfoncé?
5. Quel est le rôle du condensateur C15 , donner le chronogramme de Vc15 lorsque SW1 est enfoncé et non enfoncé.
6. Quel registre du microcontrôleur permet de placer le Port B en entrée?

Affichage multiplexé : principe, intérêts, analyse

7. Rappelez en quelques mots le principe de fonctionnement d'un affichage simple et d'un affichage multiplexé. Vous pouvez illustrer votre réponse avec un organigramme.
8. Indiquez les avantages de l'affichage multiplexé.
9. Calculez les intensités, maximale et moyenne, dans les segments ainsi que dans les afficheurs. Vérifiez si les transistors sont correctement polarisés et capables de commander les afficheurs. La carte pouvant fonctionner en mode DEBUG (*arrêt possible du multiplexage en mode pas à pas*), justifiez la sous-consommation des afficheurs tels qu'ils sont polarisés.

Communication série avec la carte moteur pas à pas

10. Indiquez la différence entre une transmission synchrone et une transmission asynchrone. De quel type est la transmission entre les cartes « Affichage adresse lyre » et « Commande moteur pas à pas » ? La carte affichage fonctionne-t-elle en maître ou en esclave ?

Analyse des dispositifs de protection du régulateur de tension

11. Quel est le rôle de la diode DP ? Justifiez son dimensionnement.
12. Calculez, ou mesurez, l'intensité fournie par le régulateur, puis vérifiez si son montage sur un dissipateur thermique est indispensable.

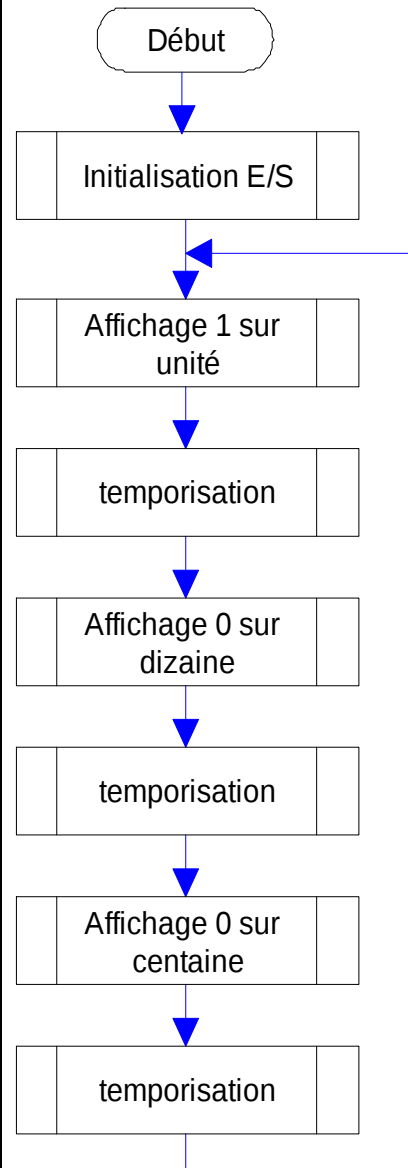


ETUDE DU PROGRAMME → *Aff_3digit_mux.mcp*

Copier tout le dossier de *Aff_3digit_mux* et coller-le dans un répertoire **D:/TGEN/groupe...(N° du groupe)/ votre nom.**

MULTILEXAGE :

Le programme principal répond à l'organigramme suivant



1. Ouvrez le projet *Aff_3digit_mux.mcp* dans le répertoire contenant les programmes.
2. Compilez et essayez le programme. Assurez vous que l'organigramme ci-contre est bien effectué.
3. Comment se nomme ce type d'affichage ?
4. Diminuez la temporisation, en changeant le paramètre dans la fonction tempo, jusqu'à ce qu'il n'y ait plus de clignotement et de scintillement.
5. Mettez un point d'arrêt devant une des fonctions tempo(NN) puis un point d'arrêt juste après cette fonction. (Dans le menu **Debugger** sélectionnez **Select tool** puis **MPLABSIM**. Dans le menu **Debugger** sélectionnez **STOPWATCH**).
6. Assurez vous que le quartz est bien à 20MHZ.
7. Lancez une simulation. Le programme s'arrête sur le premier point d'arrêt.
8. Cliquez sur **ZERO** dans la fenêtre **STOPWATCH**. Puis relancez la simulation. L'arrêt sur le deuxième point d'arrêt donne dans la fenêtre **watch** le temps écoulé depuis le dernier **ZERO**. Indiquez la durée de cette temporisation.
9. Sortez du mode simulation et passez en mode **PROGRAMMER**.
10. Faites les modifications nécessaires pour afficher l'adresse 157.



TEMPS DE MULTILEXAGE :

11. Relevez les signaux de commande des afficheurs sur J15 et J16. Identifiez sur le chronogramme le moment où l'afficheur est allumé. Mesurez sa durée et comparez avec la durée simulée. Mesurez la fréquence et le rapport cyclique du signal.
12. Dans le cas où il y aurait 4 afficheurs à commander par cette méthode, quel serait le rapport cyclique ?
13. Relevez le chronogramme de commande de l'un des afficheurs. Mesurez la période et le temps de commande de celui-ci.

*La temporisation se fait grâce à un timer LE **TIMER 0** dont voici l'initialisation :*

T0CON=0xC4;

La commande de l'afficheur se fait sur une interruption à chaque débordement du timer 0.

14. Calculez le temps de débordement (On s'aidera de la documentation sur le **TIMER0** du registre **T0CON**, et du **TIMER0 BLOC DIAGRAM**) comparez avec le temps de commande d'un afficheur.

Modification de l'affichage :

15. Modifier le programme pour faire clignoter les 3 afficheurs. Faire constater par le prof.
16. Modifier le programme pour faire allumer les afficheurs successivement de la gauche vers la droite. Faire constater par le prof.
17. Modifier le programme pour faire allumer les afficheurs successivement de la droite vers la gauche. Faire constater par le prof.
18. Modifier le programme pour faire allumer les afficheurs du centre vers l'extérieur. Faire constater par le prof.
19. Modifier le programme pour faire allumer les afficheurs de l'extérieur vers le centre. Faire constater par le prof.



Description fonctions affiche0() à affiche9()		
<pre>void affiche0(void) { PORTA=0b000100; PORTB=0b100000; }</pre>	<pre>void affiche1(void) { PORTA=0b PORTB=0b }</pre>	<pre>void affiche2(void) { PORTA=0b PORTB=0b }</pre>
<pre>void affiche3(void) { PORTA=0b PORTB=0b }</pre>	<pre>void affiche4(void) { PORTA=0b PORTB=0b }</pre>	<pre>void affiche5(void) { PORTA=0b PORTB=0b }</pre>
<pre>void affiche6(void) { PORTA=0b PORTB=0b }</pre>	<pre>void affiche7(void) { PORTA=0b PORTB=0b }</pre>	<pre>void affiche8(void) { PORTA=0b PORTB=0b }</pre>
<pre>void affiche9(void) { PORTA=0b PORTB=0b }</pre>		

