**Réalisation d’une journal lumineux par liaison RS-232 à base de Microcontrôleur PIC**

**Réalisé par :**

**Ben Ettaieb Houssemeddine**

**Abidi Abdelhamid**

**&**

**Tlig Mohamed**

Introduction générale

Depuis sa naissance, la micro-électronique a permis le développement de nombreuses technologies domestiques, industrielles et technologies pour les laboratoires et a fait naître une nouvelle discipline aussi dominante qui est l’informatique. Elle a su prendre sa place dans de très nombreuses applications. Cependant, depuis un peu plus d’une dizaine d’années, nous avons percevoir une forte demande concernant les systèmes de petites tailles intégrant des fonctions électrique, mécanique, thermique, magnétique ou encore optique. Il s’agit par exemple de capteurs de petites tailles, de petits robots explorateurs dans le corps humain, etc.

Les progrès technologiques dans le domaine de la micro-électronique permettent de répondre à la demande croissante du marché en circuit intégrés rapides, consommant peu et offrant la possibilité d’intégrer des fonctions électroniques de plus en plus complexes.

Dans ce travail, l'intérêt majeur est d’attribuer à la recherche d’une solution permettant d’automatiser la commande des afficheurs lumineux avec un circuit programmable, simple à utiliser surtout pour le changement du message et avec un prix réduit.

Donc, notre projet consiste à réaliser un prototype d’un journal lumineux géré par un microcontrôleur qui est commandé par un ordinateur à travers d’une liaison série qui facilite la modification du message par une simple interface de commande.

Ce rapport est subdivisé en deux grands chapitres :

Dans le premier chapitre, on va définit le journal lumineux, l’étude théorique du microcontrôleur utilisé et la liaison entre PC et la carte de commande. Le deuxième chapitre sera consacré à l’étude pratique en général.

L’enchaînement des données est représenté par le schéma synoptique si dessous en précisant les circuits intégrés utilisés telle que le PIC16F877A, le registre à décalage (CD4094BE), le démultiplexeur (74hc138), l’amplificateur de courant (74LS541) et le convertisseur (MAX232).

**SCHEMA SYNOPTIQUE**

***PC***

16F877a

**µc**

**Matrices à LED**

**74HC138**

Max232

**CD4094BE**

**RS 232**

**74LS541**

**74LS541**

# 

# Chapitre1

**Généralité sur les journaux lumineux**

* 1. Introduction

Dans ce chapitre, on va présenter d’abord une généralité sur le journal lumineux et l’étude des matrices LEDs. On va étudier aussi le microcontrôleur et ces diverses utilisations, Puis, on va présenter le principe de la liaison série entre le PC et le PIC.

* 1. Journal lumineux
     1. généralité

Les afficheurs électriques sont de plus en plus utilisés dans la vie quotidienne, bien évidemment les journaux lumineux qui peuvent être à base des diodes LED, des afficheurs LCD ou des matrices à LED. Il existe actuellement une variété innombrable d'écritures lumineuses pour vitrines, ou panneaux indicateurs, à contenu variable, ou tout simplement destinées à attirer l'attention. Les appareils se prêtant à cet emploi sont toutefois relativement coûteux et leur réalisation dépasse le niveau du simple bricolage. Le projet d'écriture lumineuse présenté ici, simple à réaliser qu'à utiliser, présente en outre l'avantage d'être peu coûteux.

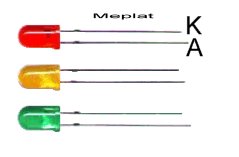
Les afficheurs à [LED](http://www.sonelec-musique.com/electronique_theorie_led.html) sont des afficheurs qui permettent de visualiser un chiffre ou un nombre ou un caractère quelconque. Les sources lumineuses sont internes au composant et permettent l'affichage même en pleine nuit, ce qui n'est pas le cas des afficheurs LCD non rétro-éclairés. C'est quasiment toujours ce type d'afficheur qui est utilisé sur les radioréveils et dans les affiches public. Dans les afficheurs à LED on peut utiliser des LEDs uni couleur ou multi couleurs.

On peut aussi réaliser des nombreux effets lumineux qui se succèdent automatiquement comme par exemple, un défilement de gauche à droite, de droite à gauche ou de haut vers le bas.

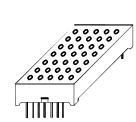
Pour notre projet on a choisi d’utiliser des matrices à LED 7\*5 préfabriquées qui sont plus performants et plus présentables que des matrices construites à la main.

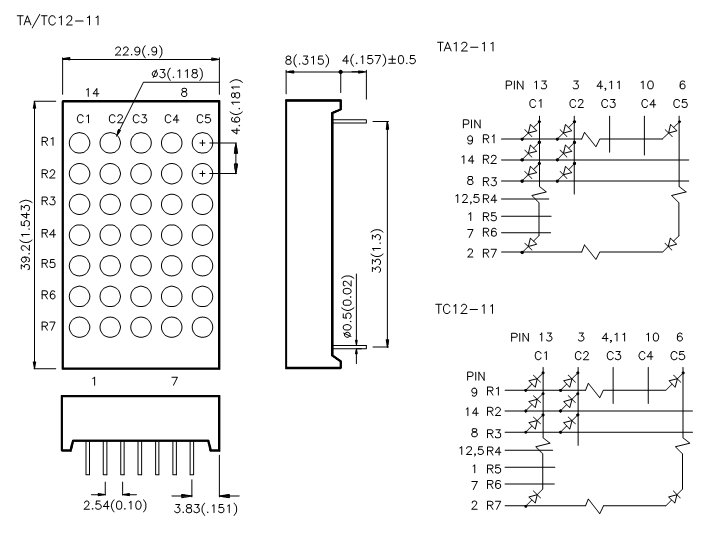
1.2.2 Etudes des matrices à LED

La LED (Light Emitting Diode, diode électroluminescente) est un composant dit passif, de la famille des semi-conducteurs (comme la [diode](http://www.sonelec-musique.com/electronique_theorie_diode.html) et le [transistor](http://www.sonelec-musique.com/electronique_theorie_transistor.html)). Il s'agit d'une diode un peu particulière, qui a la propriété d'émettre de la lumière quand un courant la parcourt (de l'Anode vers la Cathode), et les couleurs que l’on trouve généralement sont : rouge, vert et jaune.

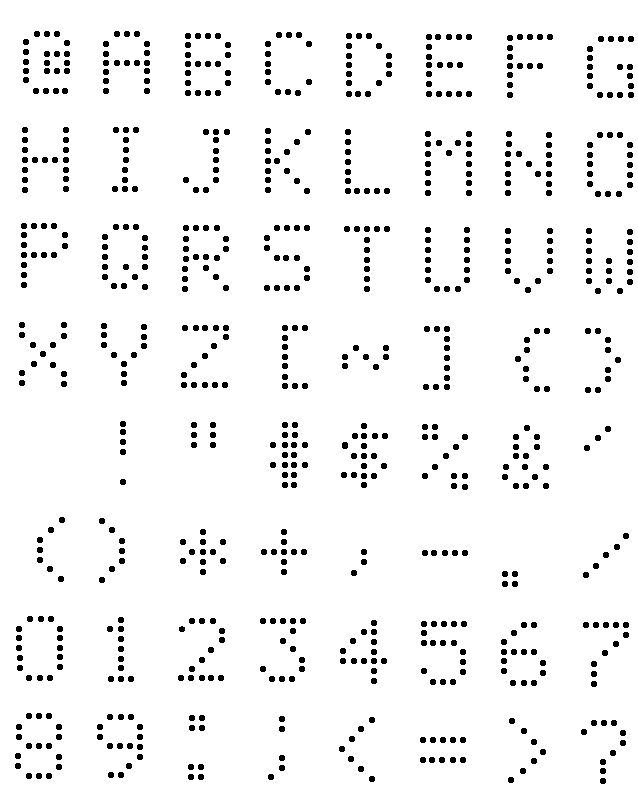


Dans notre cas on a utilisé un afficheur qui emploi 35 LEDs en forme de points disposés selon une matrice à 7 lignes et 5 colonnes à anode commune sur les colonnes.





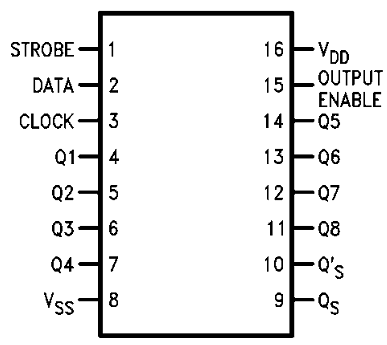
L’affichage est réalisé en allumant en même temps ou à des instants différents et périodiques les point lumineux nécessaires du tableau pour communiquer l’information désirée, en ce sens le tableau d’affichage est constitué de points lumineux couramment appelés pixels (c’est une abréviation anglaise de « Picture élément »). La taille du pixel dépend de la distance de laquelle l’affichage est prévu d’être lu. La distance entre les pixels est imposée par la résolution désirée des caractères à afficher et par les dimensions du tableau. Ce qui rend l’affichage une séquence de données logiques définissant l’extinction de certains pixels et l’allumage d’autres, de ce fait l’utilisation d’un calculateur est indispensable et suivant les taches à accomplir et la diversité des types d’affichage comme dans notre cas qu’on a utilisé le PIC16F877a. Les caractères du message sont affichés de la façon suivante :



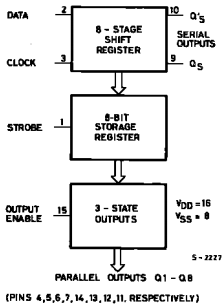
* Le défilement du message est géré par l’intermédiaire de registre à décalage ‘CD4094’ qu’on a utilisé.
* Le registre à décalage :

En électronique et en informatique, un registre à décalage est un registre de taille fixe dans lequel les bits sont décalés à chaque coup d'horloge (dans le cas d'un système synchrone sur l'horloge). Il est en général constitué d'un chaînage de bascules synchronisées sur l'horloge, la sortie d'une bascule étant reliée à l'entrée de la suivante.

Comme critère du choix on a utilisé le registre à décalage **74HC4094** qui est caractérisé par seize pins.



Son principe de fonctionnement est comme suit :

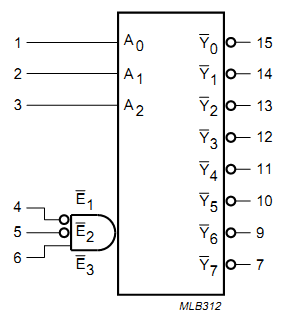
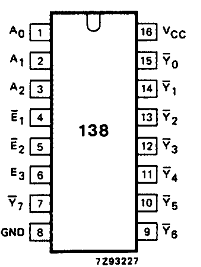


* Le démultiplexeur/décodeur :

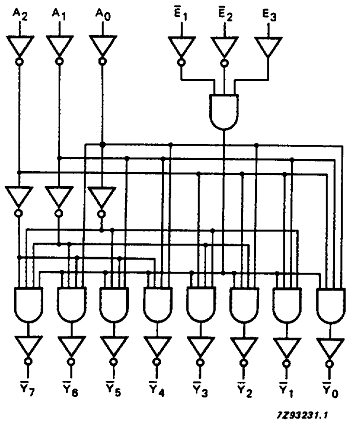
Un démultiplexeur est un circuit combinatoire à N+1 entrées et 2N sorties. N entrées, appelées entrées d'adressage, permettent d'envoyer sur l'une des sorties la dernière entrée, appelée l'entrée donnée.

Un décodeur est un cas particulier de démultiplexeur dans lequel on relie l'entrée donnée à l’état bas. Il est donc un circuit combinatoire à N entrées et 2N sorties. Sélectionner une sortie grâce aux entrées d'adressage la fera passer de l'état 1 à l'état 0.

On a utilisé le démultiplexeur/décodeur **74hc138** accepte trois entrées binaires (A0, A1, A2) et faire sortir huit sorties inversées (de Y0 à Y7).



Son schéma interne est :



1.3Etude de microcontrôleur PIC

1.3.1 Définition

Les microcontrôleurs sont aujourd’hui implantés dans la plupart des applications grand public ou professionnelles, il en existe plusieurs familles.

La société Américaine Microchip Technologie a mis au point dans les années 90 un microcontrôleur CMOS : le PIC (Periphirol Interface Contrôler). Ce composant encore très utilisé à l’heure actuelle, est un compromis entre simplicité d’emploi, rapidité et prix de revient.

Les caractéristiques du PIC sont :

* Séparation des mémoires de programme et de données (architecture Harvard) : On obtient ainsi une meilleure bande passante et des instructions et des données pas forcément codées sur le même nombre de bits.
* Communication avec l'extérieur seulement par des ports : il ne possède pas de bus d'adresses, de bus de données et de bus de contrôle comme la plupart des microprocesseurs.
* Utilisation d'un jeu d'instructions réduit : d'où le nom de son architecture : RISC (Reduced Instructions Set Construction). Les instructions sont ainsi codées sur un nombre réduit de bits, ce qui accélère l'exécution (1 cycle machine par instruction sauf pour les sauts qui requirent 2 cycles). En revanche, leur nombre limité oblige à se restreindre à des instructions basiques, contrairement aux systèmes d'architecture CISC (Complex Instructions Set Construction) qui proposent plus d'instructions donc codées sur plus de bits mais réalisant des traitements plus complexes.

Il existe trois familles de PIC :

* La famille Baseline : les instructions sont codées sur 12 bits.
* La famille Mid-range : les instructions sont codées sur 14 bits.
* La famille High performance : les instructions sont codées sur 16 bits.

Dans notre projet, on a utilisé le pic 16F877A qui appartient à la famille Mid-Range, qui utilise des mots de 14 bits. On a choisi ce microcontrôleur parce qu’il est le mieux performant dans notre cas grâce à ces avantages par rapport aux autres microcontrôleurs.

Il possède :

- Un nombre suffisant d’entrées/sorties

- Une mémoire flash adéquate pour contenir la totalité du programme

- Une fréquence d’horloge convenable

La référence d’un microcontrôleur PIC est de la forme NN LLL XXX, où :

NN : désigne la famille à laquelle appartient le circuit.

LLL : est un ensemble d’une, deux ou trois lettres qui indiquent le type de mémoire de programme contenue dans le circuit et si la plage de tension d’alimentation est normale ou étendue.

XXX : est un ensemble de deux ou trois chiffres constituant la référence du circuit.

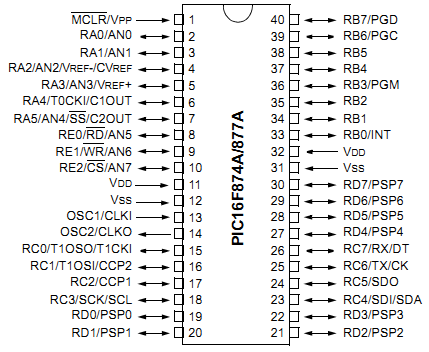
Exemple : 16F877

16 : indique la famille mid-range.

F : mémoire utilisé de type FLASH.

1. identité.

* Brochage du PIC



1.3.2 Architecture interne du PIC 16F877a

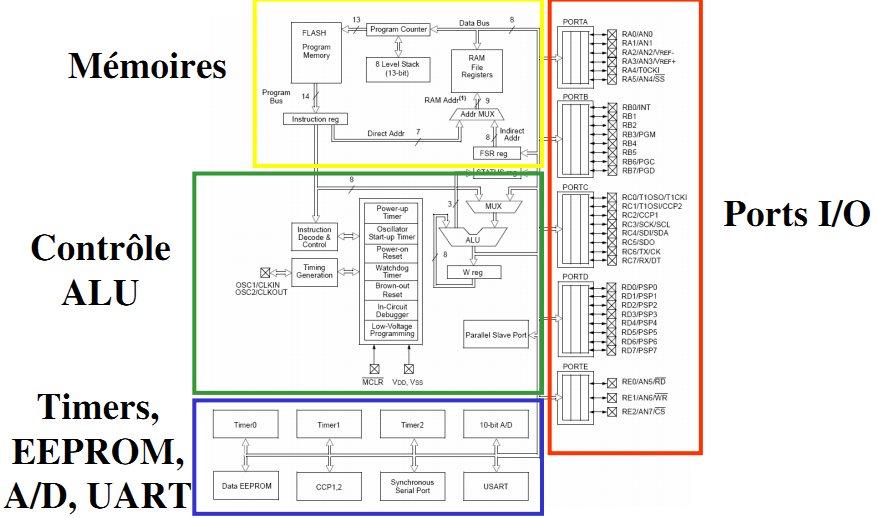
Comme pour tous les circuits intégrés, chacun de ses broches à une ou plusieurs fonctions qui sont résumées par un sigle mnémotechnique.

Ce microcontrôleur présente une architecture Harvard, les données sont placées dans une mémoire de type RAM de 368 bites. La mémoire de programme est constituée de mot de 14 bites et de type FLASH.

Ces ressources sont donc précieuses, en comparaison de celles d’autres composants.

Le 16F877a possède 5 ports (A à E) et 3 temporisateurs (timers).

* Ce diagramme présente les différents blocks du PIC 16f877a :



* La mémoire EEPROM :

La mémoire EEPROM (Electrical Erasable Programmable Read Only Memory) est constituée de 256 octets qui peuvent être lus et/ou écrits à partir du programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi- permanents. Il est donc plus rapide de la lire que d’y écrire.

* La mémoire RAM :

La mémoire RAM est celle que nous allons sans cesse utiliser. Toutes les données qui y sont stockées sont perdues lors d’une coupure de courant. La mémoire RAM est organisée en quatre banques (0,1, 2 et 3) pour le 16F877a. La RAM est subdivisée de plus en deux parties. Dans chacun des banques nous allons trouver des « cases mémoires spéciales » appelées registres spéciaux et des cases mémoires « libres ».

* Les timers / compteurs :

Notre pic possède trois timers qui sont :

* Le timer 0 :

Le timer 0 est un compteur 8 bits qui peut compter (de 0 à 255) :

-soit les impulsions de l'horloge via un pré-diviseur.

-soit des impulsions externes, via la broche RA4.

* Le timer 1 :

Le timer 1 est un compteur 16 bits qui peut compter (de 0 à 65535) :

- soit les impulsions de l’horloge.

- soit les impulsions externes, et en particulier les impulsions d'un quartz externe.

* Le timer 2 :

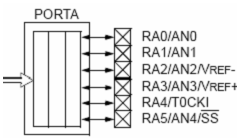
Le timer 2 est un compteur 8 bits couplé au module dit CCP. Nous l'utiliserons essentiellement pour la génération d'impulsions à période ajustable (PWM).

Tous ces timers peuvent déclencher une interruption interne, s’ils ont été autorisés.

* + 1. Les entrées sorties du PIC

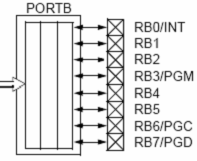
On trouve cinq ports d’entées sorties (A E).

* PORT A :



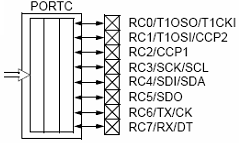
Comme le montre le schéma si dessus, le port A possède six pins d’entrée sortie numérotées de RA0 à RA5. Nous avons donc six bits à utiliser dans le registre Tris A, dans notre cas on a utilisé trois pins qui sont (RA0, RA1 et RA2) pour commander le décodeur 74hc138.

* PORT B :



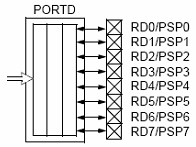
Le port B est formé de huit pins entrées/sorties numérotées de RB0 à RB7, nous avons donc huit bits à utiliser dans le registre Tris B. Il peut être configuré pour générer une interruption sur un changement d’état de broche RB4 commandée par un bouton poussoir.

* PORT C :



Le port C possède huit pins entrées/sorties numérotées de RC0 à RC7, nous avons donc huit bits à utiliser dans le registre Tris C. Il peut être configuré pour gérer la transmission et la réception des données à travers les broches RC6/TX et RC7/RX.

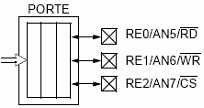
* PORT D :



Le port D possède aussi huit pins entrées/sorties numérotées de RD0 à RD7, donc nous avons huit bits à utiliser dans le registre Tris D. Dans notre cas on a utilisé seulement trois bits pour commander les registres à décalage CD4094 comme suit :

RD0 pour gérer le cycle d’horloge ‘clock’, RD1 pour envoyer les données et RD2 pour le ‘strobe’.

* PORT E :



Ce Port ne comporte que trois pins d’entrées/sorties qui sont RE0, RE1 et RE2. Dans notre cas on ne va pas utiliser ce port.

* MCLR :

Cette broche sert à initialiser le microcontrôleur. Elle peut être simplement reliée à VDD si on n’a pas besoin de RESET externe. Par contre si on souhaite implanter un bouton de remise à zéro, on pourra câbler un simple réseau RC sur la broche MCLR.

## Oscillateur : OSC1 et OSC2 ou CLKIN et CLOUT :

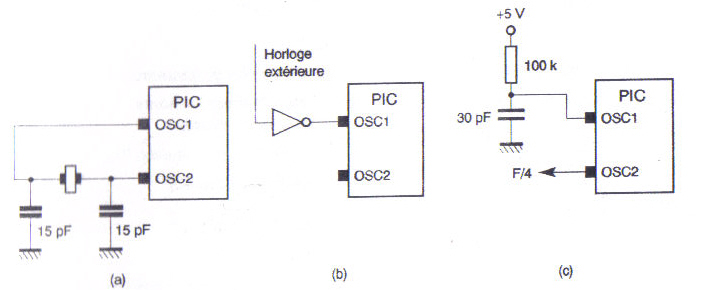
Ces broches permettent de faire fonctionner l’oscillateur interne du PIC.

On peut utiliser 3 types d’oscillateurs :

**-** Un quartz ou résonateur céramique **(a)**

- Un oscillateur externe **(b)**

- Un réseau RC **(c)**

****

* Alimentation : VDD et VSS :

Ce sont les broches d'alimentation du circuit. Les tensions qui peuvent être appliquées vont :

1. - De 4,5V à 6V pour la gamme standard F.
2. - De 2 à 6V pour la gamme étendue LF.

L'intensité du courant consommé peut aller de 1μA à 10mA.

La consommation du microcontrôleur sera fonction de :

- La tension d'alimentation.

- La fréquence interne.

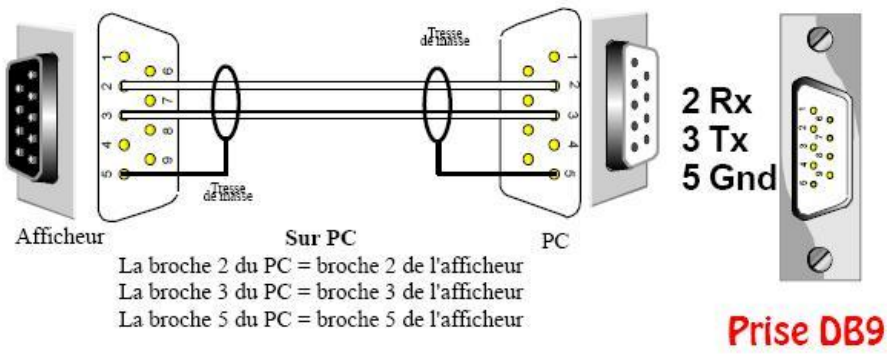
- Le mode de fonctionnement.

* 1. La liaison série entre le PC et le PIC

On a choisie d’utiliser la liaison série RS232 (Recommandé Standard 232). Elle est utilisée dans tous les domaines de l'informatique (ex : port de communication com1 et com2 des PC, permettant la communication avec des périphériques tels que modem / scanner / …), et également dans le domaine des automatismes pour relier des capteurs aux automates, pour programmer un automate par un PC….

* + 1. Connecteur

La norme RS-232 peut être utilisée avec un connecteur Sub-DE9 (généralement appelé Sub-DB9), il s’agit de la norme EIA/TIA 574.



* Configuration des broches :
* **Brochage du connecteur DB9 :**

|  |  |  |
| --- | --- | --- |
| Abréviation | Broche | Nom complet |
| TX | **3** | Transmitted Data |
| RX | **2** | Received Data |
| DTR | **4** | Data Terminal Ready |
| DSR | **6** | Data Set Ready |
| RTS | **7** | Request To Send |
| CTS | **8** | Clear To Send |
| DCD | **1** | Data Carrier Detect |
| RI | **9** | Ring Indicator |
| GND | **5** | **masse** |

- Broche N°1 : DCD (Data carrier détect), cette ligne est une entrée active haute. Elle signale à l’ordinateur qu’une liaison a été établie avec un correspondant.

- Broche N°2 : RX (receive Data), cette ligne est une entrée. C’est ici que transitent les informations du correspondant vers l’ordinateur.

- Broche N°3 : TX (Transmit Data), cette ligne est une sortie qui transmettre les données du PC vers un correspondant.

- Broche N°4 : DTR (Data Terminal Ready) : cette ligne est une sortie active haute. Elle permet à l’ordinateur de signaler au correspondant que le port série a été libéré et qu’il peut être utilisé s’il souhaite.

- Broche N°5 : GND (GROUND), c’est la masse.

- Broche N°6 : DSR (Data Set Ready), cette ligne est une entrée active haute. Elle permet au correspondant de signaler qu’une donnée est prête.

- Broche N°7 : RTS (Request To Send) : cette ligne est une sortie active haute. Elle indique au correspondant que l’ordinateur veut lui transmettre des données.

- Broche N°8 : CTS (Clear To Send), cette ligne est entrée active haute. Elle indique à l’ordinateur que le correspondant est prêt à recevoir des données.

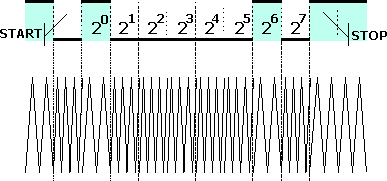
- Broche N°9 : RI (Ring Indicator) : cette ligne est une entrée active haute. Elle permet à l’ordinateur de savoir qu’un correspondant veut initier une communication avec lui.

* Fonctionnement :

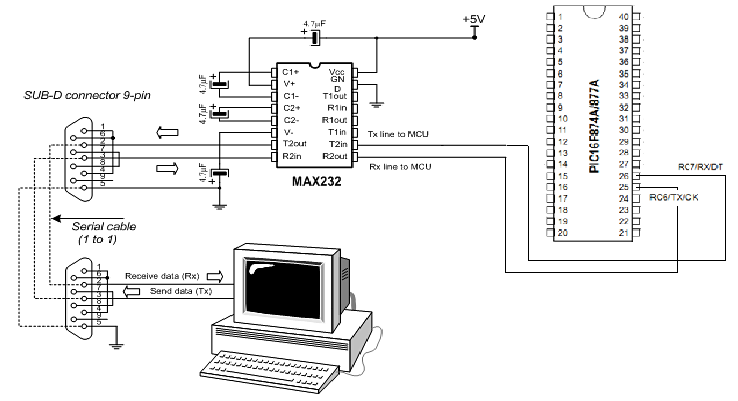
La liaison série est totalement asynchrone. Aucune horloge n’est transmise, il faut donc que les deux systèmes (PC/PIC) soient d’accord sur la vitesse de transfert des bits préalablement à toute transmission.

Donc la solution est de rajouter des bits de synchronisation encadrant chaque jeu de bits transmis : un bit de START et un bit de STOP. La transmission s’effectue donc octet par octet sur un fil.

Exemple : transmission de la lettre ‘A’ ($41)



Le pic 16F877A utilise les niveaux 0v et 5v pour définir respectivement les bits : 0 et 1. La norme RS 232 définit des niveaux de -12v et +12v pour établir ces mêmes niveaux Nous avons donc besoin d’un circuit (driver de bus) chargé de convertir les niveaux des signaux entre PIC et PC alors que la solution c’est le circuit MAX232.

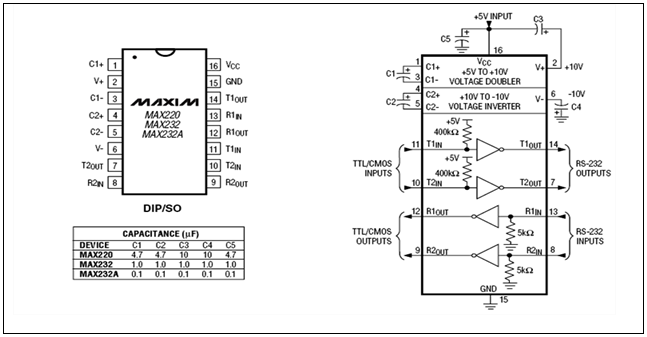


* + 1. Convertisseur MAX232

Le MAX232 est un standard depuis longtemps, il permet de réaliser des liaisons RS232 et des interfaces de communication, il amplifier et met en forme deux entrées et deux sorties TTL/CMOS vers deux entrées et deux sorties RS232, la connexion est réalisée avec un DB9.

Ce circuit dispose de :

* Deux blocs nommés T1 (T1IN et T1OUT) et T2 (T2IN et T2OUT) qui convertissent les signaux d’entrées 0v et 5v en signaux de sorties +12v et -12v.
* Deux blocs nommés R1 (R1IN et R1OUT) et R2 (R2IN et R2OUT) qui convertissent les signaux d’entrées +12v et -12v en signaux de sorties 0v et 5v.



1.5 Conclusion

Partant d’une présentation générale sur les afficheurs lumineux, nous avons ensuite défini les microcontrôleurs et ses différents familles, plus particulièrement le 16F877A, puis on a parlé des matrices des LEDs et des composants qui l’entourent et arrivons à la liaison série entre le PC et le la carte de commande.

En conclusion dans ce chapitre nous pouvons dire que le microcontrôleur 16F877A peut bien jouer le rôle d'une unité de contrôle pour notre système et qu’on a besoin d’une interface de communication PC/PIC.

Maintenant, nous pouvons passer à l’étude de la conception de notre carte puisque les composants les plus importants sont déjà familiers.

# Chapitre1

**Réalisation de la carte**

2.1 Introduction

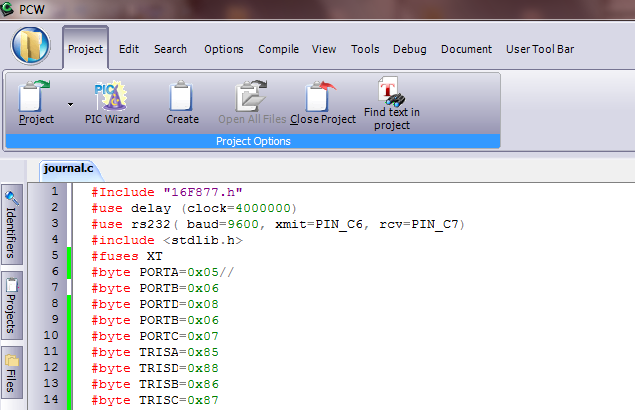
Dans ce chapitre on va présenter une description détaillée de la solution adoptée pour répondre aux besoins de notre projet et en abordant la conception détaillée de chaque partie du système afin d’obtenir une schématisation complète et précise.

* 1. Le passage des données

L’affichage de message sur les matrices doit passer par les étapes suivantes :

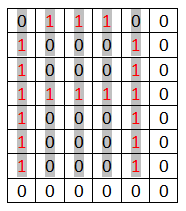
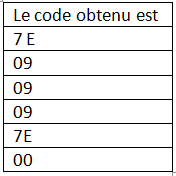
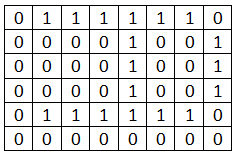
* L’envoi de message du PC vers le port série RS232
* Réception du message à afficher bit par bit sur RS232.
* Conversion des données par le convertisseur MAX232 qui est connecté au pin 6 et pin 7 de PORTC du PIC (RC6/TX; RC7/RX).
* Le message va être identifié caractère par caractère
* Chaque caractère est représenté par 6 tampons en fonction de numéro de la colonne.
* Le journal est constitué de 7 lignes, chaque ligne possède 45 diodes, donc une ligne correspondant à un tampon de 45 bits.
* Les bits de tampon sont envoyés bit par bit par le pin RD1 du port D vers les registres.
* Lorsque la première ligne est chargée dans tous les registres le pin RD2 est activé (STB) pour faire l’affichage.
* Le démultiplexeur va changer vers la ligne 2 jusqu’à 7 dans un boucle infini.
  1. Programme du journal

Il existe plusieurs langages de programmation qu’on peut l’utiliser pour programmer notre PIC et dans ce projet on a choisi d’utiliser le langage C parce qu’il est le plus facile, sur tout pour la configuration de la liaison RS232, et on a utilisé comme compilateur le CCS compiler qui est spécifié pour la programmation des PIC.



Codage des caractères : (exemple lettre A)

Le principe de codage des caractères est très facile, on écrit la lettre sous forme des ‘1’ et des ‘0’ dans un tableau de huit colonnes et six lignes puis on l’inverse et on prélève le code sur huit bits :

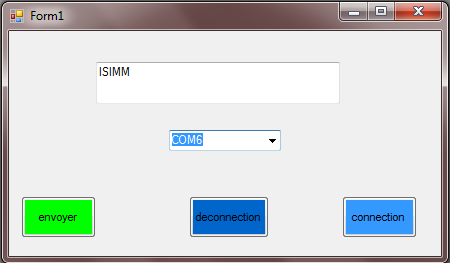
 

Remarque

On a ajouté la sixième colonne pour espacer les lettres.

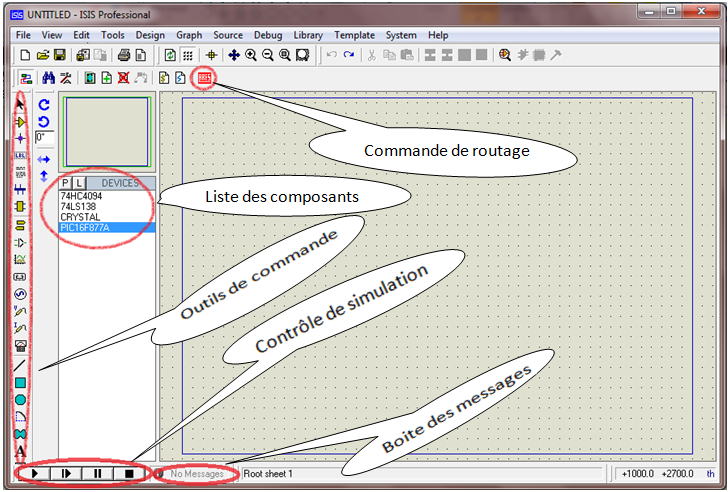
* 1. Interface de communication

Pour assurer la liaison série entre le PC et le PIC on a créé une simple interface **VB (visuel basic)** qui permet d’envoyer le message à afficher au PIC sans aucune perte des données.

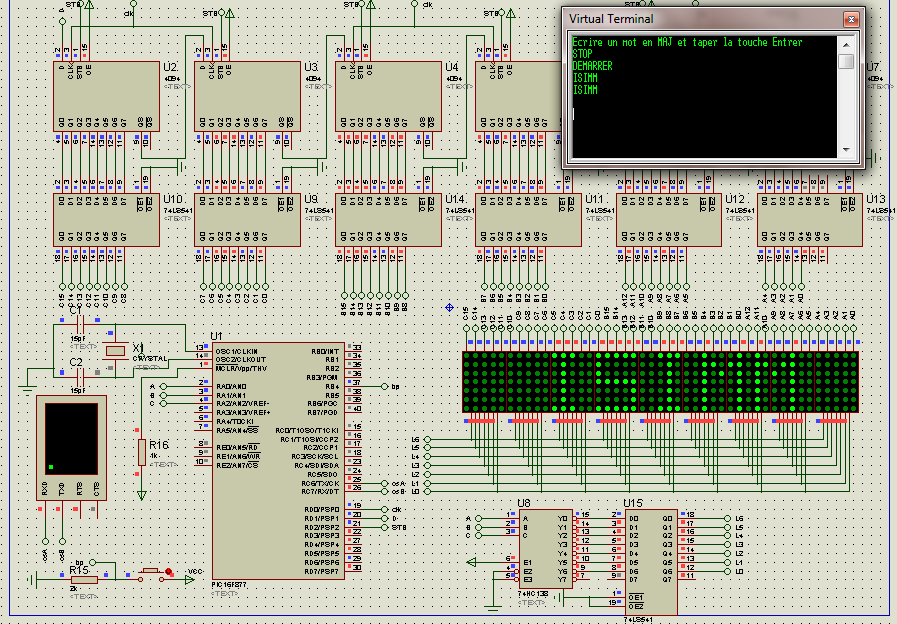
****

* 1. Montage du circuit
     1. Simulateur

Pour la réalisation du montage on a recours au logiciel ISIS (Intelligent Schematic Input System) qui est dédié pour la CAO (Conception Assistée par Ordinateur) et qu’il a comme objectif la saisie d’un schéma électronique, puis procéder à une animation en temps réel du montage et exploiter les résultats de la simulation. Ce logiciel facilite beaucoup nos travail et si la simulation démarre correctement, il nous garanti beaucoup que notre projet fonctionne dans le plan réel.

****

* + 1. Schéma électronique

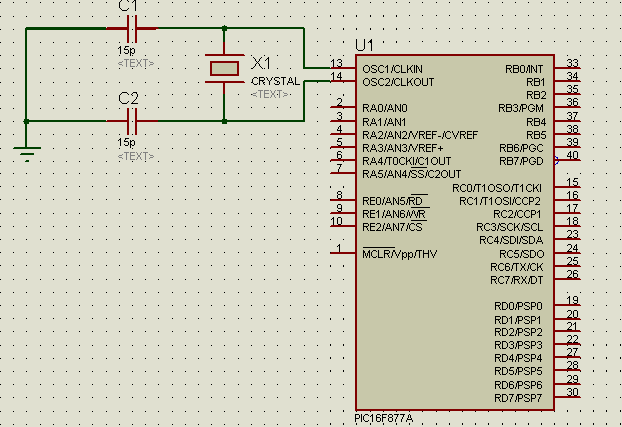


* Principe de fonctionnement :
* Chargement du fichier (.hex) dans le PIC et régler la fréquence du PIC à 4Mhz.
* Appuier sur le bouton ‘play’ pour la simulation.
* Appuier sur le bouton poussoir deux fois successives, puis taper le message à affiché et appuier sur la touche entrer.

2.6 Description des différents blocks

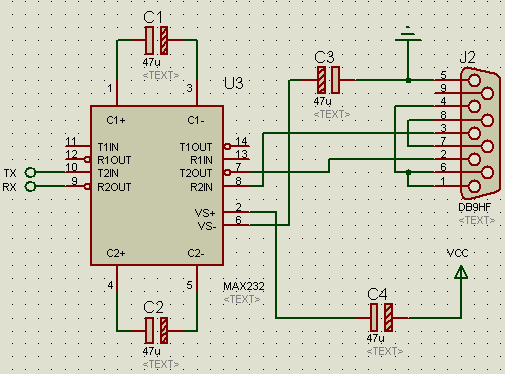
* Circuit oscillateur :

L’horloge est un système qui peut être réalisée soit avec un QUARTZ, soit avec une horloge extérieur, soit avec un circuit RC,  dans ce dernier la stabilité du montage est limitée. Pour réaliser cette fonction on a utilisé un QUARTZ de 4 MHZ relié de ces bornes vers les broches osc1 et osc2 du PIC et deux capacités de 15pF chacune reliées à la masse.



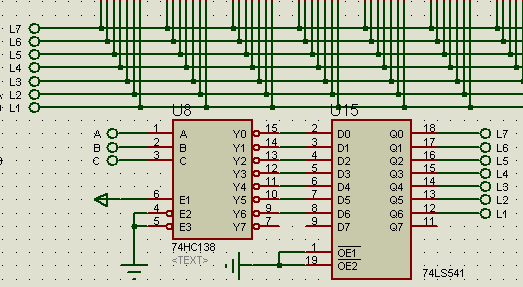
* Le convertisseur MAX232 :

Ce schéma montre que le MAX232 relier avec le DB9 (utilisé les pines T2OUT et R2IN) et le Pic 16F877a (utilisé les pines T2IN et R2OUT) et d’autre part avec les broches (RC6/TX et RC7/RX) du PIC.



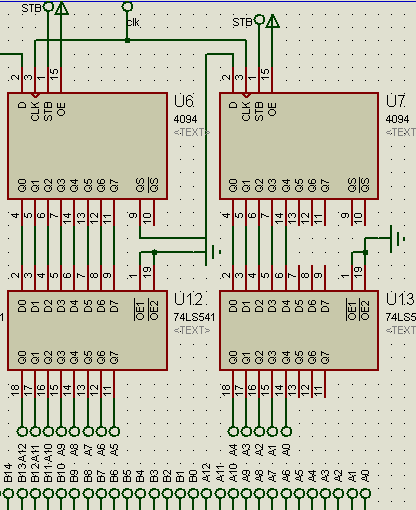
* Le démultiplexeur 74hc138 :

On a utilisé un système de démultiplexage commandé par trois sorties du PIC à base de circuit 74hc138 pour pouvoir contrôler huit sorties qui génèrent les lignes des matrices en utilisant un amplificateur de courant pour la bonne visibilité d’éclairage .



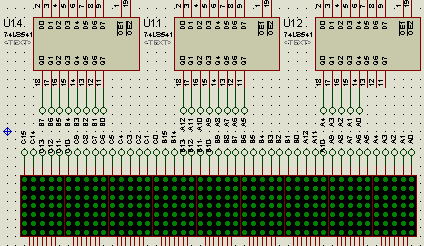
* Les registres à décalage :

On a utilisé le registre à décalage Cd4094 pour commander les colonnes des matrices et assurer le défilement du message de droite à gauche en utilisant trois Broche du PIC pour son fonctionnement, on le branche aussi à un amplificateur de courant et son branchement sur la carte est comme suit :



* Les amplificateurs de courant :

Le pilotage des lignes et des colonnes est confié à des amplificateurs 74LS541 pour amplifier le courant d’entré aux matrices car le décodeur 74HC138 et les registres à décalage ne sont pas suffisants pour fournir les 20mA pilotant chaque LED, alors qu’on a 45 LED dans chaque ligne et 7 LED dans chaque colonne.



* Le bouton poussoir :

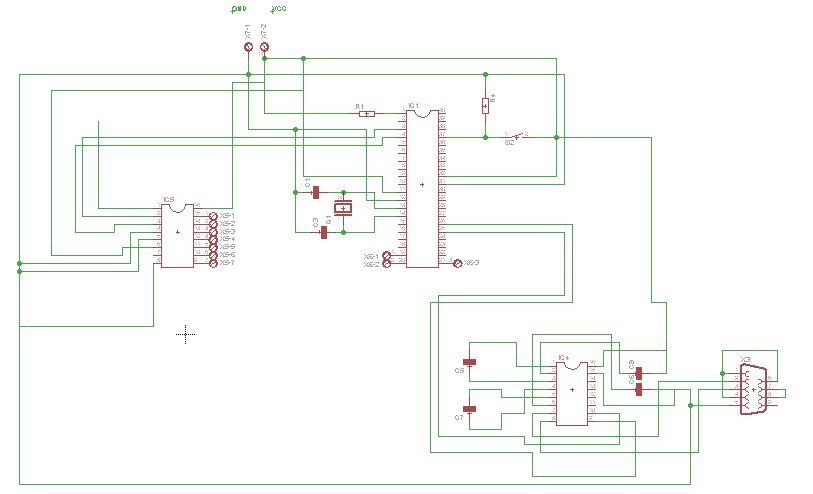
Le bouton poussoir est utilisé pour afficher ou n’est pas afficher le message, il est commander par le PIC à traves la broche RB4. Il correspond au bouton de REZET du PIC dans son branchement.



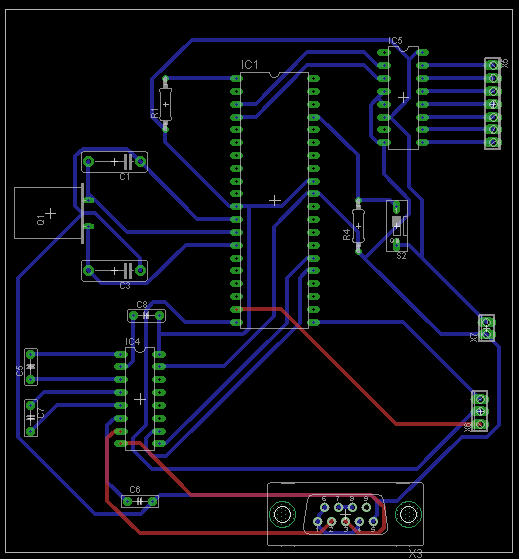
* 1. Routage

Pour réaliser le routage des nos cartes (carte de commande et carte de l’afficheur), on a utilisé l’EAGLE qui est très utile pour ce type d’application à cause de ces divers commandes pour modifier les moindres détails de routage tel que l’épaisseur des pistes, la distance entre eux et la largeur des drills s’il y on a, etc.…

* + 1. Carte de commande
       1. schéma de carte de la commande

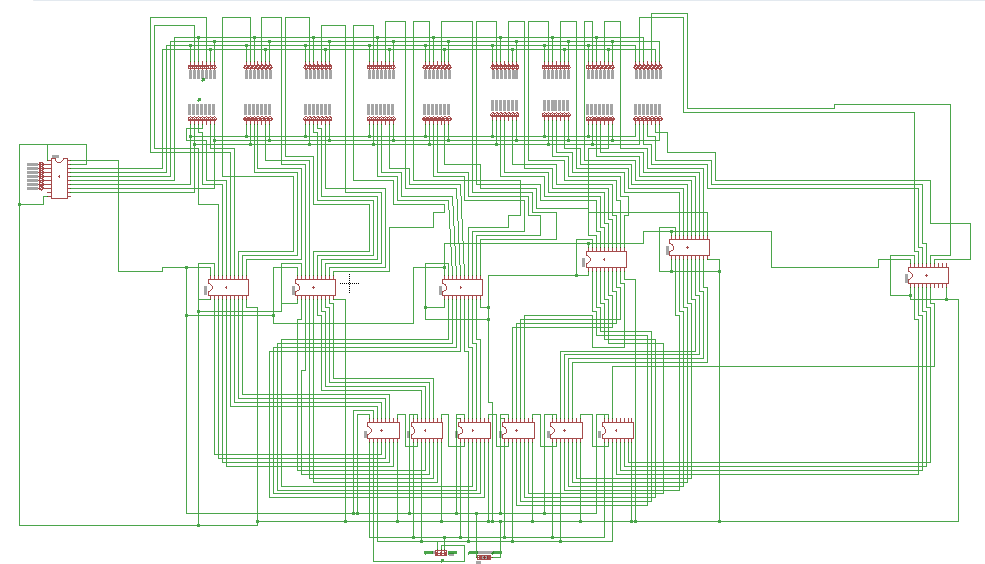
****

* + - 1. **Routage de la carte de commande**

****

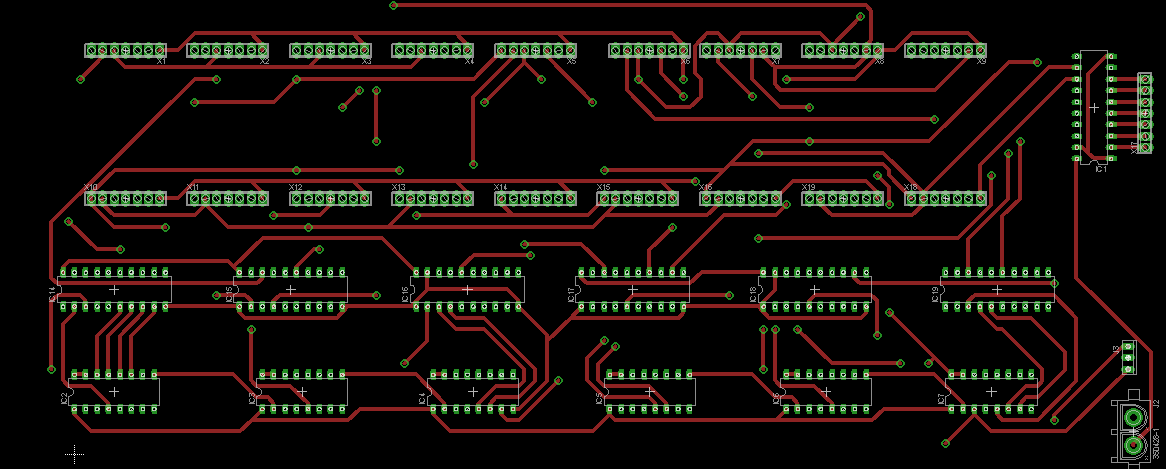
* + 1. Carte de l’afficheur

2.7.2.1 Schémas de l’afficheur

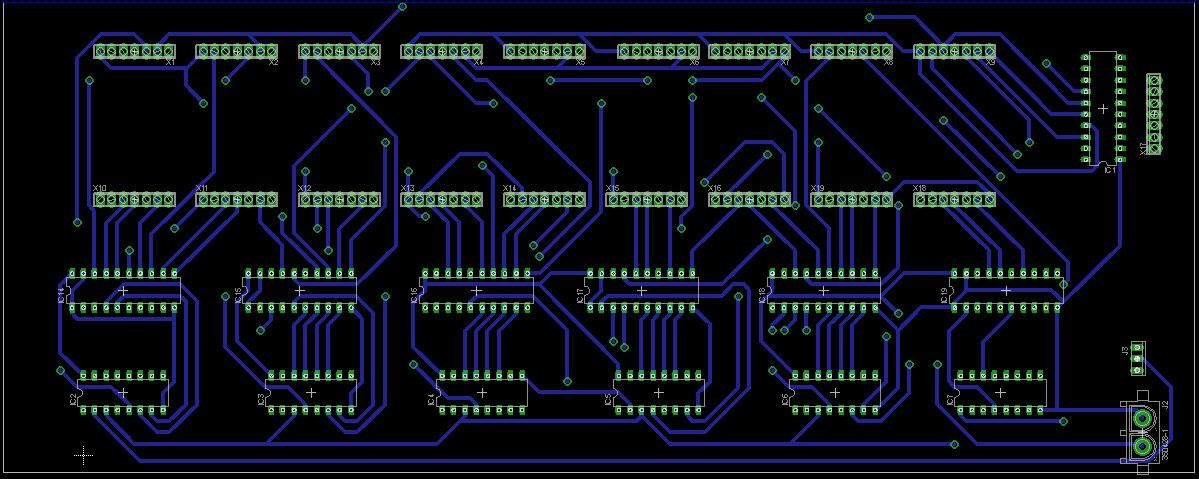


2.7.2.2 Routage de l’afficheur

* Face avant (face de composant):



* Face arrière (face de soudure) :



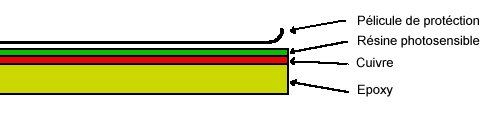
* 1. Réalisation pratique

Pour la réalisation d’une carte électronique on doit passer par plusieurs étapes qui sont :

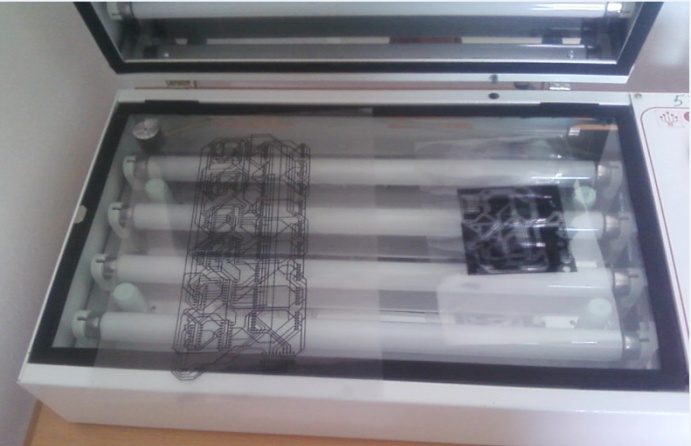
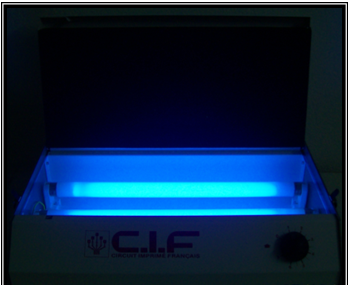
* + 1. Insolation

L’insolation de plaque époxy est tout simplement de faire exposer la plaque aux rayonnements Ultra Violet (UV) de l’insoleuse après avoir retiré le film protecteur, les étapes d’insolation sont :

* Retirer la pellicule protectrice de la plaque (au dernier moment).
* Poser cette plaque sur le typon dans l'insoleuse et fermer le capot.
* Insoler environ 1 à 5 minutes puis retirer la plaque.

**

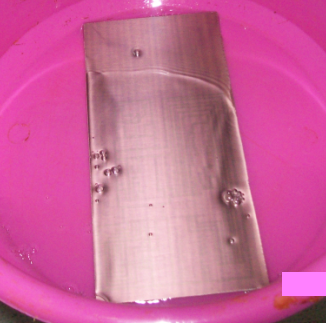
Les couches de plaque époxy



Insoleuse simple face Insoleuse double face

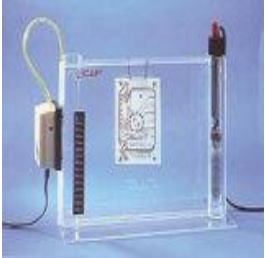
* + 1. Révélation

L’étape de révélation consiste à dissoudre les zones de résine qui ont été détruite pendant l’insolation, pour cette étape on utilise un liquide qui contient du sodium et qu’il s’appelle ‘produit révélateur’.



* + 1. Gravure

La gravure consiste à plonger le circuit dans un bain d'acide (Perchlorure de fer) qui est soumis à une température pour accélérer la gravure. Le cuivre mis à nu lors de la révélation sera dissous. Seules les pistes protégées par la résine qui vont restées.



A la suite de gravure chimique on frotte un peut la plaque affin d’apparaitre les piste à cuivre, puis on la plonge dans un autre liquide ‘l’acétone’ pour changer la couleur des pistes.



Maintenant et après toutes ses étapes notre carte est prête pour la percer et la mise en place des composants.

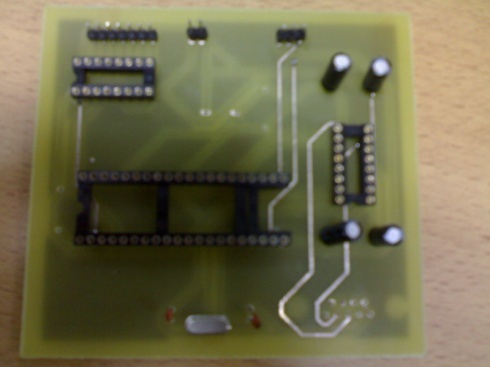
* + 1. Perçage

Avant de la mise en place des composants on doit tout d’abord percer ces pastilles sur la plaque, cette étape est réalisée à l’aide d’une mini perceuse fixe à son support pour la bonne précision dans l’emplacement de foret.

****

* + 1. Soudure

A présent on doit placer les composants sur la plaque, pour cela on a besoin d’un faire à souder et de l’étain parce qu’il est facile à manipuler grâce à sa basse température de fusion. Dans la soudure on commence par souder les composants neutres comme les supports des circuits intégrés et les connecteurs etc.

Face de soudure face des composants

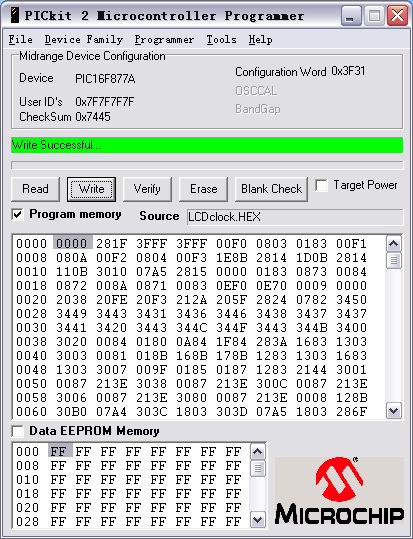
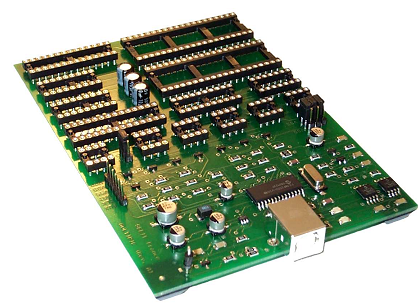
Après la soudure et le test de la continuité des pistes et le court-circuit, nos cartes sont maintenant prêtes à êtres employées.

Maintenant il nous reste que de transférer le programme vers le PIC et de percevoir le résultat.

* 1. Programmation du PIC

Pour implémenter le programme dans le PIC on a utilisé un MULTIPIC qui permet la programmation de plus de 350 microcontrôleurs PIC de chez Microchip, (séries PIC10Fxxx, PIC12Fxxx, PIC16Fxxx, PIC18Fxxx, PIC24Fxxx, dsPIC30Fxxx et dsPIC33Fxxx) grâce à ses supports tulipes pour les composants en boîtiers DIP 8, 14, 18, 20, 28 et 40 broches et son connecteur SPI pour les composants soudés sur circuit ou l commandé par le logiciel Pickit2. Les étapes de transfert du programme sont :

* Mettre le PIC dans l’emplacement convenable sur le multi-pic
* Brancher le MULTIPIC au PC
* Lancer le logiciel Pickit2
* Il faut qu’il détecte automatiquement le pic et il indique son type.
* Importer le fichier (.hex) puis cliquer sur ‘écrire’ ou en anglais ‘write’.

Logiciel de programmation MULTIPIC

* 1. Conclusion

La réalisation d’un circuit imprimé prend autant du temps et d’expérience, elle passe par plusieurs étapes depuis la simulation et le routage jusqu’à l’implémentation des composants et la programmation du pic.

Conclusion générale

L’élaboration de ce travail dans le cadre du projet de la fin d’étude (projet tutoriel), nous a permis d’approfondir nos connaissances théoriques en électronique et d’acquérir une bonne expérience au niveau de la réalisation pratique.

Lors de cette manipulation, on a essayé de fournir l’automatisation de l’éclairage public.

**????????**

………………………………..

……………………………….