



Projet de Fin d'Etudes
pour l'obtention du diplôme
d'Ingénieur d'Etat en Electronique

Présenté par :

REDJEL Ilyas

Intitulé :

**Etude et réalisation d'un panneau
routier à messages variables**

Soutenu publiquement le **26/06/2012** devant le jury composé de :

Président :	BERKANI Daoud	Professeur	Ecole Nationale Polytechnique
Rapporteur :	HADDADI Mourad	Professeur	Ecole Nationale Polytechnique
Examineur :	ADNANE Mourad	Docteur	Ecole Nationale Polytechnique

ENP 2012

Résumé :

Les panneaux routiers traditionnels offrent aux utilisateurs de la route des informations figés dans le temps ; la forte demande d'informations en temps réel entraîne la nécessité d'un nouveau type de panneaux, qui s'adapte avec l'état de la route pour afficher des informations décrivant l'état de cette route en temps réel ; pour surmonter ces problèmes on propose d'utiliser des panneaux à messages variables, dont on a réalisé un prototype. Notre prototype reçoit les informations d'un ordinateur ou d'un clavier, et les affiche sur un panneau constitué par des matrices à LED, les matrices sont adressées séparément par l'entremise d'un registre à décalage pour afficher seulement un caractère sur chaque matrice tout en utilisant la caractéristique de persistance rétinienne de l'œil humaine.

Mots clés :

Panneau routier, messages variables, matrice à LED, microcontrôleur, affichage d'information, persistance rétinienne.

ملخص:

إشارات المرور التقليدية تمنح لمستخدم الطريق معلومات جامدة زمنياً، إن الطلب الملحّ على المعلومات في وقتها الحقيقي تتطلب استحداث نوع جديد من إشارات المرور قادرة على التكيف مع حالة الطريق من أجل عرض المعلومات على شكل رسالة تصف حالة هذا الطريق في الوقت الحالي، من أجل التغلب على هذه المشاكل نقتراح استعمال إشارات إعلانية متغيرة والتي قمنا بإنجاز نموذج عنها. النموذج المنجز يستقبل المعلومات عن طريق الحاسوب أو عبر لوحة مفاتيح مستقلة، ويقوم بعرضها على لوحة مؤلفة من مصفوفات ضوئية، هذه المصفوفات يتم بها عن طريق مسجل إزاحة من أجل عرض حرف واحد على كل منها مع الاستفادة من خاصية العين البشرية "استمرارية الرؤية"

كلمات مفتاحية:

إشارات المرور، رسالة متغيرة، مصفوفة ضوئية، متحكم دقيق، عرض المعلومات، استمرارية الرؤية.

Abstract :

Conventional road signs provide the user with time-frozen information; the high demand for real-time information leads to the need of a new type of signs, which fits with the condition of the road to display information in the form of a message describing the road's condition in real time. In order to solve these problems, we suggest using variable message signs from which we have built a prototype that receives information from a computer or an independent keyboard, and then shows it on dot-matrix panel, these matrixes are addressed separately through a shift register to display only one character on each matrix using the characteristic "persistence of vision" of the human eye.

Keywords :

Roadside signs, variable message, dot matrix, microcontroller, information display, persistence of vision.

Remerciements

Au nom de Dieu le Clément et le Miséricordieux.

Louange à Dieu Seigneur des mondes, et que la bénédiction et la paix soient sur le dernier des envoyés de Dieu : Mohamed, sa famille, ses compagnons et tous ceux qui les ont suivi sur la bonne voie jusqu'au jour de la rétribution.

*Mes remerciements les plus particuliers vont à mon promoteur monsieur **M.HADDADI**, qu'il trouve ici l'expression de ma vive gratitude pour ses précieux conseils, son suivi, sa disponibilité et son aide.*

*Monsieur **D.BERKANI** m'a fait l'honneur de présider le jury de ce mémoire. Qu'il en soit vivement remercié. Je le remercie, encore une fois, en tant que mon enseignant.*

*Mes remerciements vont aussi à Monsieur **M.ADNANE**, qui m'a fait l'honneur d'être l'examineur de ce mémoire. Pour avoir consacré du temps à l'évaluation de ce travail, je lui exprime ma vive reconnaissance.*

*J'exprime mes remerciements à mes **chers parents** et toute la famille pour leur soutien, leurs sacrifices et leurs encouragements.*

Enfin, merci pour tous ceux qui ont contribué de près ou de loin à l'élaboration de ce projet.

Ilyas.

Dédicaces

*Je dédie ce modeste travail à mes **chers parents** à qui j'adresse au Dieu les vœux les plus ardents pour la conservation de leur santé et de leur vie.*

*A mes **chers frères et sœurs** à qui je souhaite beaucoup plus de succès.*

*A mon oncle **Dr. Ammar** qui m'a toujours suivi avec ses précieux conseils.*

*A tous mes **chers amis**.*

*Et à toute la promotion des électroniciens : **2012**.*

Ilyas

Sommaire

Sommaire	I
Liste des figures	IV
LISTE DES TABLEAUX	VI
Liste d'abréviations :	VII

Introduction générale	1
------------------------------	----------

Chapitre I: Généralités sur les panneaux routiers à messages variables

I. Introduction	3
II. Définition	3
III. Historique	3
IV. L'émergence de la signalisation à messages variables	5
V. Types de PMV	5
VI. Spécifications et installation	6
VI.1 Caractéristiques du panneau	6
VI.2 Dimensions	6
VI.3 Police	6
VI.4 Contraste	6
VI.5 Placement	7
VI.6 Distance de lisibilité	7
VI.7 Distance de lisibilité requise	8
VI.8 Distance de lisibilité réelle	8
VII. Les PMV dans le monde	9
VII.1 Une nouvelle tendance, l'affichage sous forme de diagramme	9
VII.2 Panneaux spécifiques	13
VII.3 Différents points intéressants visant l'utilisation des PMV	14
VIII. Conclusion	16

Chapitre II: Etude et présentation de la carte et ses composants

I. Introduction	17
II. Les différents modules	17
II.1 Carte de contrôle	17
II.1.1 Microcontrôleur PIC 18F252	17
II.1.2 Le driver ULN 2804	20
II.1.3 L'afficheur LCD	21
II.1.4 Clavier PS2	21
II.1.5 La liaison RS232 (DB9)	22
II.2 Carte d'affichage	25
II.2.1 Matrice à LED HS2088B	25
II.2.2 Le registre à décalage 74HC595	27
III. Conclusion	29

Chapitre II: Etude et présentation de la carte et ses composants

I. Introduction	30
II. Principes de fonctionnement et routage	30
II.1 La carte de commande	30
II.1.1 Le microcontrôleur	31
II.1.2 Le driver ULN2804	33
II.1.3 L'afficheur LCD	33
II.1.4 Les connecteurs	33
II.2 La carte d'affichage	35
II.2.1 La matrice à LED	35
II.2.2 Le registre à décalage 74HC595	37
III. Réalisation et programmation	39
III.1 Programmation du PIC	39
III.1.1 Le langage C	39
III.1.2 Le compilateur mikroC PRO	39
III.1.3 Le programmeur PICFLASH	40
III.1.4 La carte de développement EASYPIC5	41
III.2 Le fonctionnement du programme	42
III.2.1 Les initialisations	42
III.2.2 Les déclarations	43
III.2.3 Les fonctions	44
IV. Test et validation	53
IV.1 Allumer les matrices	53
IV.2 Commander une matrice par le 74HC595	53

IV.3 Afficher l'alphabets sur une matrice	54
IV.4 Tester l'écran LCD	55
IV.5 Tester les connexions PS/2 et RS232	55
IV.6 Balayer tout le panneau	55
V. Conclusion :	57
Conclusion Générale	58
Bibliographie	59
Annexes:	61

Liste des figures

Figure	Titre	Page
Chapitre I		
Figure 1.1	Distance de lisibilité	8
Figure 1.2 (a)	PMV spécial au Japon	9
Figure 1.2 (b)	PMV spécial au Corée du sud	9
Figure 1.3	Panneau graphique en chine	10
Figure 1.4 (a)	Panneau graphique au Pays-Bas	10
Figure 1.4 (b)	Panneau graphique en Espagne	10
Figure 1.5 (a)	Affichage numérique au Danemark	11
Figure 1.5 (b)	Affichage numérique en France	11
Figure 1.6 (a)	Panneau multimodal en Allemagne	12
Figure 1.6 (b)	Panneau multimodal au Japon	12
Figure 1.7	Panneau à image vidéo en Corée du Sud	13
Figure 1.8	Pictogramme avec texte en Italie	14
Figure 1.9	Utilisation combinée de textes et de pictogrammes	15
Figure 1.10	Utilisation de symboles graphiques	15
Figure 1.11	PMV librement programmables	16
Chapitre II		
Figure 2.1	Schéma des broches du PIC 18F252	19
Figure 2.2	Schéma des broches du ULN2803	20
Figure 2.3	schéma illustratif d'un bloc de l'ULN2804	20
Figure 2.4 2	Une entrée PS/2	22
Figure 2.5	Une sortie RS232	23
Figure 2.6	Schéma des broches du MAX232	24
Figure 2.7	Schéma du montage d'un MAX232	24
Figure 2.8	connexion RS232 utilisant le MAX232	25
Figure 2.9	Schéma interne de matrice HS2088B	25
Figure 2.10.(a)	Dimensions de la matrice HS2088B	26
Figure 2.10.(b)	Dimensions de la matrice HS2088B	26
Figure 2.11	Les caractéristiques électriques de matrice HS2088B	27
Figure 2.12	Schéma des broches du 74HC595	28
Figure 2.13	chronogramme élémentaire du 74HC595	29
Chapitre III		
Figure 3.1	schéma bloc de la carte globale	31
Figure 3.2	Le PIC 18F252	32
Figure 3.3	Oscillateur quartz de 20 MHz	32
Figure 3.4	Driver ULN2804	33

Figure 3.5	Afficheur LCD	33
Figure 3.6	Connecteur PS/2	34
Figure 3.7	Connecteur coudé	34
Figure 3.8	La carte de commande	35
Figure 3.9	Matrice à LED (Dot Matrix)	35
Figure 3.10	Alimentation d'une LED	36
Figure 3.11	Registre à décalage 74HC595	37
Figure 3.12	La carte d'affichage	38
Figure 3.13	Fenêtre du compilateur MikroC PRO pour PIC.	40
Figure 3.14	Fenêtre de PICFLASH	41
Figure 3.15	La carte de développement EASYPIC5	42
Figure 3.16	Code pour le caractère A	44
Figure 3.17.(a)	Formation de la lettre 'A'	46
Figure 3.17.(b)	Formation de la lettre 'A'	46
Figure 3.17.(c)	Formation de la lettre 'A'	47
Figure 3.18	Résultat de formation de la lettre 'A'	47
Figure 3.19	Organigramme d'affichage d'un caractère HC595()	48
Figure 3.20.(a)	Codes du balayage d'un clavier PS/2	49
Figure 3.20.(b)	Codes du balayage d'un clavier PS/2	49
Figure 3.21	Organigramme de fonction PS2	50
Figure 3.22	Organigramme de fonction RS232	51
Figure 3.23	Organigramme de fonction lcd	52
Figure 3.24	Balayage par ligne	53
Figure 3.25	Affichage d'alphabets	54
Figure 3.26	Test d'écran LCD	55
Figure 3.27	Balayage du panneau complet	56
Figure 3.28	Test global	56

LISTE DES TABLEAUX

Table	Titre	Page
Tableau 1.1	Le contraste d'un PMV	6
Tableau 1.2	distance de lisibilité réelle	9
Tableau 2.1	Caractéristiques générales du PIC 18F252	19
Tableau 2.2	Caractéristiques générales de l'ULN2804	20
Tableau 2.3	Caractéristiques générales du PVC200401PTN	21
Tableau 2.4	Les caractéristiques électriques de matrice HS2088B	27
Tableau 2.5	Broches du 74HC595	28

Liste d'abréviations :

I_F :	Courant direct
I_R :	Longueur d'onde
I_V :	Intensité lumineuse
V_F :	Tension directe
λ_p :	Longueur d'onde
ASCII :	American Standard Code for Information Interchange
CAN:	convertisseur analogique numérique
CEMT :	Conférence européenne des Ministres des Transports
CMOS :	Complementary metal-oxide-semiconductor
CMS:	composant monté en surface
DIP:	Dual in-line package
EIA :	Electronic Industries Alliance
ESD:	Electrostatic discharge
FIT:	Forum International des Transports
FIVE :	Framework on Implementation of VMS in Europe
GND :	Ground
I/O :	input/output
I2C:	Inter-Integrated Circuit
LCD :	Liquid crystal display
MCLR :	Master Clear
MLI:	Modulation de Largeur d'Impulsion
NMOS :	N-channel MOSFET
PC :	Personal Computer
PIC :	Peripheral Interface Controller
PMOS :	P-channel MOSFET

- PMV :** Panneau à messages variables
- PS2 :** Personal System/2
- SFR :** Special function register
- TIA :** Telecommunications Industry Association
- TTL :** Transistor–transistor logic
- UART:** Universal asynchronous receiver/transmitter
- USB :** Universal Serial Bus

Introduction générale

Le concept de la route «auto-explicative» englobe de nombreux éléments différents touchant à la conception, à l'aménagement et à l'exploitation du réseau routier. Toutefois, ce terme illustre fort bien la nécessité de créer un environnement qui puisse être facilement compris et utilisé en toute sécurité par l'ensemble de ses usagers.

Pour examiner cette question essentielle on doit penser à créer une signalisation routière qui satisfasse aux exigences de la motorisation en croissance constante. Les niveaux de motorisation augmentent de manière particulièrement rapide dans notre pays, où les usagers de la route, qu'ils soient motorisés ou non, continuent de se partager la chaussée, souvent au plus grand détriment de la sécurité routière.

Cette signalisation doit suivre le rythme du développement routier et autoroutier de notre pays car les panneaux traditionnels offrent aux utilisateurs de la route des informations figés dans le temps et l'espace. Les informations faisant état d'une route coupée, de travaux, d'un accident, des bouchons, et d'autres informations liés à l'utilisation de la route en temps réel doivent être validées et mises en service.

L'utilisation des panneaux routiers à messages variables devrait être la solution pour ce problème ; un panneau routier à messages variables s'adapte avec l'état de la route pour afficher des informations sous forme d'un message décrivant l'état de cette route en temps réel, soit par l'utilisation de capteurs, soit par l'envoi d'un message approprié.

Un panneau à messages variables routiers doit être visible pour tous les usagers de la route, et doit afficher des messages clairs et lisibles à une distance confortable, et par conséquent un tel panneau doit avoir une dimension de quelques mètres pour répondre à ces exigences.

Dans le cadre d'un vaste projet qui s'occupe de la modernisation du réseau routier en Algérie, notre tâche était la conception et la réalisation d'un prototype du panneau routier.

Notre travail consiste à étudier les panneaux routiers à messages variables et les normes qui devront être appliqués pour ce type de panneaux. Afin de proposer et réaliser un

prototype répondant à ces normes, on équipe notre prototype avec différents accessoires assurant sa gestion complète. Le prototype reçoit le message à afficher soit d'un ordinateur via une connexion série, soit par un clavier de type PS/2 intégré qui remplace la liaison série avec l'ordinateur.

Dans notre cahier de charge, le prototype devra être compatible avec l'installation d'un émetteur/récepteur sans fil pour permettre l'affichage du message par voie hertzienne en temps réel.

Un écran de type LCD servant à afficher les différents messages d'aide et les commentaires sur le fonctionnement du prototype est installé au niveau du dispositif pour afficher les messages de contrôle et les messages émis.

Notre travail est décrit dans les trois chapitres suivants :

Chapitre 1 : Une étude générale sur les panneaux routiers est présentée dans ce chapitre, ainsi que les différentes normes de taille, d'emplacement et de police.

Chapitre 2 : Une présentation de différents composants constituant le prototype, ainsi qu'une étude sur chacun d'eux pour s'assurer de leur compatibilité et leurs capacités d'exécution des différentes tâches qu'on leur a attribuées.

Chapitre 3 : Une validation des résultats, ainsi qu'une présentation finale du prototype, une explication détaillée sur le fonctionnement du programme du microcontrôleur avec des figures et des organigrammes, et des tests exécutés pour la vérification du bon déroulement du programme écrit.

Chapitre I : Généralités sur les panneaux routiers à messages variables

Introduction

Les panneaux d'information à messages variables sont devenus aujourd'hui un facteur clé dans la publicité des entreprises pour transmettre des informations concernant leurs produits en permanence, et dans la signalisation routière pour informer les usagers de la route en temps réel sur l'état des réseaux routiers et les conditions de circulation (climatiques, densité de trafic, état de la chaussée...etc.), et permettent de gérer l'écoulement du trafic et leur diffusion.

I. Définition

Les **Panneaux à Messages Variables** -PMV- (ou bien : Variable Message Signs – VMS) sont des dispositifs conçus pour afficher un message parmi plusieurs qui peuvent être changés en fonction des besoins. Ils constituent un support important de l'information routière. Ils sont l'élément de communication entre le gestionnaire et les usagers de la route.

On les rencontre de plus en plus souvent et leur importance va croissant. Ils affichent des messages générés électroniquement qui peuvent être modifiés pour afficher des informations texte prédéfinies ou libres, pictogrammes et symboles.

Les PMV sont conçus pour influencer sur le comportement des automobilistes (et donc l'amélioration de la fluidité de la circulation) en fournissant en temps réel des informations concernant le trafic routier [1].

II. Historique

La signalisation routière n'est apparue pour de bon qu'avec l'avènement de l'automobile à la fin du dix-neuvième siècle. Bien que différents systèmes aient été appliqués à partir de 1890, il avait fallu attendre 1909 pour que soit signée la première convention internationale.

Une conférence des Nations-Unies sur les grandes routes et les transports routiers avait été organisée à Genève en août et septembre 1949. Des délégations en provenance de 28 pays y participèrent. Cette conférence se référait au projet de convention (préparé par le Comité des transports au sein de la Commission économique des Nations Unies pour l'Europe et à la Convention interaméricaine de 1943 sur la réglementation du trafic automobile interaméricain.

Les résultats de cette conférence furent premièrement de décider l'élaboration d'un protocole sur la signalisation routière et deuxièmement le constat qu'il était impossible de s'entendre sur un système uniforme de signalisation routière qui serait universellement accepté par les pays intéressés.

Les géométries et les coloris des panneaux des deux systèmes continuent de différer : pour les panneaux de danger, le liséré rouge prévaut dans les pays européens (système utilisé en Europe et dans de nombreux pays asiatiques -la Chine en premier lieu- et des pays africains - y compris l'Algérie -), le fond jaune et le symbole noir du système américain étant utilisés aux États-Unis et, assortis de quelques variantes, en Irlande, au Canada, au Mexique, certains pays d'Amérique centrale et du Sud, au Japon, etc..

Conscients de la nécessité d'harmoniser leurs règlements, les pays européens ont pris depuis 1949 l'habitude de se réunir régulièrement, sous l'égide de la Commission Européenne, des Nations-Unies à Genève et de la Conférence européenne des ministres des transports (CEMT), récemment réorganisée en FIT (Forum International des Transports).

Après toute une série d'accords internationaux, des efforts entrepris pour standardiser et harmoniser les panneaux routiers atteignirent un stade crucial en 1968 lorsque fut rédigée la convention sur la signalisation routière, encore appelée « **Convention de Vienne** » et soutenue par un grand nombre de pays en Europe, ainsi que par certains pays asiatiques et africains [2].

Dans cette **Convention de Vienne** historique signée en 1968, l'article 7 était le seul, à faire mention de la signalisation variable : « *Rien dans la présente Convention n'interdit d'employer, pour transmettre des renseignements, des avertissements ou des règles applicables seulement à certaines heures ou certains jours, des signaux dont les indications ne sont visibles que lorsque les renseignements qu'ils transmettent sont pertinents.* » [3]

Donc Les panneaux à messages variables ont été officiellement introduits dans la **Convention de Vienne** de 1968 sur la signalisation routière par un amendement entré en vigueur le 30 novembre 1995.

III. L'émergence de la signalisation à messages variables

Au cours des dernières décennies du vingtième siècle, le nombre de voitures sur les routes et le nombre de problèmes routiers ont rapidement augmenté. Peu de temps

en revanche a été consacré à développer le stade le plus avancé du cycle d'information routière, à savoir les signaux routiers affichés sur les PMV. Les PMV gagnant en popularité, différentes administrations chargées de gérer les routes et des exploitants routiers ont tenté de tirer un parti maximal des possibilités offertes.

IV. Types de PMV

On distingue 4 types de PMV [4] :

PMV permanents - généralement montés sur des portiques en bordure de route, les autoroutes et les artères principales, capable d'afficher plusieurs lignes d'informations et de texture.

PMV permanents améliorés – semblables aux PMV permanents - en termes d'endroits d'installation, mais capables d'afficher des graphiques et des symboles, en plus de plusieurs lignes d'informations textuelles.

PMV mobiles - généralement montés sur une remorque ou un autre véhicule qui peut être déplacé dans un endroit au besoin, ceux-ci sont utilisés à des endroits où il n'y a pas de VMS permanents actuels, et sont capables d'afficher plusieurs lignes d'informations textuel (mais beaucoup moins que les PMV améliorés).

PMV montés sur véhicules - montés sur la face avant, le dos ou sur le toit des véhicules et peuvent être utilisés sur les sites de travaux routiers, les unités d'intervention de la circulation et d'autres véhicules de gestion de la circulation, et par les véhicules des services de police et d'urgence.

Les PMV peuvent être utilisés dans un certain nombre de façons incluant:

- Gère les incidents (tels que le détournement de trafic).
- Donner l'alerte pour les automobilistes sur les conditions de circulation en aval du signe, la limite de vitesse (variable), et les situations qui ont le potentiel d'être un problème de sécurité routière.
- Fournir des informations sur les futurs événements prévus.
- Fournir des conseils de la circulation pendant les travaux.
- Fournir des informations générales (telles que des messages de sécurité routière, information concernant le stationnement, l'information temps de parcours, et les conditions).

V. Spécifications et installation

VI.1 Caractéristiques du panneau

Lorsque la vitesse est limitée à 60 km/h ou moins, le panneau a une dimension de quatre lignes et 18 caractères de largeur fixe par ligne au maximum (de type A) [5].

Lorsque la vitesse est limitée à 70 km/h ou plus, le panneau a une dimension de trois lignes et 18 caractères de largeur fixe par ligne au maximum (type B et C) [5].

Pour assurer la clarté du message, les caractères de chaque ligne devraient être séparés par un minimum de deux pixels.

Les PMV améliorés contiennent une matrice de texture semblable à celle d'un PMV typique permanent, cependant, avec 19 caractères par ligne.

En plus d'une matrice de texture, les PMV améliorés contiennent également un écran secondaire capable d'afficher des graphiques et des symboles. Ces graphiques ou symboles sont généralement utilisés pour afficher un signe d'avertissement, un signe d'aide, ou un signe temporaire.

VI.2 Dimensions

Les dimensions d'un PMV sont dictées par leur emplacement, la taille et le nombre de caractères et de lignes, ainsi que la vitesse. Les zones à vitesse plus élevée (comme les autoroutes) nécessitent des tailles de caractères plus grandes pour fournir la distance de lisibilité appropriée.

VI.3 Police

L'exigence minimale doit être de générer des caractères simples et clairs, en général les caractères doivent être disposés de manière à avoir un minimum de 2 pixels entre eux et de 2 pixels entre les lignes [6].

VI.4 Contraste

Le contraste est la luminance du PMV divisé par la luminance de l'arrière-plan du panneau. Le tableau (1.1) exploite l'exigence sur le contraste.

Tableau 1.1 : Le contraste d'un PMV

Luminance du panneau	Ratio minimal	Luminance (Cd/m ²)	
		Min	Max
40,000	10	6,200	120,000
4,000	10	1,100	21,000
400	10	300	3,700
40	-	200	2,000
4	-	60	300

VI.5 Placement

Un PMV peut être installé soit sur le bord, soit au-dessus de la route ; cependant il faut tenir compte de son impact visuel. L'installation routière qui est une option plus économique et peut être moins intrusive, exige toutefois une route plus large et peut éprouver une visibilité réduite en raison de la végétation ou d'une circulation forte [5].

Un PMV doit être placé de sorte qu'il soit clairement lisible à tous les usagers de la route et ne devrait pas être placé de sorte d'être en concurrence avec d'autres panneaux de signalisation, dispositifs de contrôle de la circulation routière, ou des meubles. Un PMV ne devrait pas être placé dans des endroits où il sera partiellement caché par un objet, mobilier urbain ou de la végétation.

Le placement d'un PMV devrait prendre en considération toutes les conditions environnantes. Par exemple, sur les autoroutes éclairées, le placement du PMV doit être compris entre les poteaux d'éclairage pour éviter les effets d'ombre.

En règle générale, les PMV peuvent être confortablement lus quand ils sont à moins de 10 ° de l'horizontale et 5 ° de la verticale de l'axe qui définit le champ de vision du conducteur [5].

Les PMV devraient être positionnés de façon à ce que les usagers aient le temps de répondre aux messages affichés, et devraient donc être placés à une distance suffisante avant d'atteindre le point de décision (détournements par exemple).

Il est désirable qu'un PMV soit situé à 900 - 1200m avant chaque point de dérivation sur l'autoroute (principalement des sorties). Les panneaux entre les points de sortie doivent être espacés selon les principes suivants:

- L'espacement minimal entre les PMV: 400m.
- L'espacement souhaitable entre PMV: 3000m.
- L'espacement maximal souhaitable entre VMS: 5000m.

Ces espacements sont également valables pour les routes urbaines. Cependant, les PMV doivent être positionnés 400 - 700m avant les intersections [5].

VI.6 Distance de lisibilité

La distance de lisibilité est dictée par la taille des caractères, la vitesse d'approche, le décalage vertical ou horizontal. Le temps nécessaire pour lire et comprendre un écran de message relativement complet est d'environ 2,5 secondes [6].

Les messages doivent être aussi courts que possible pour éviter toute interférence.

La distance de lisibilité totale, comme le montre la figure (1.1), est le résultat de:

- L'observation du panneau avant la lecture.
- La lecture du message.
- L'incapacité de lire le message.

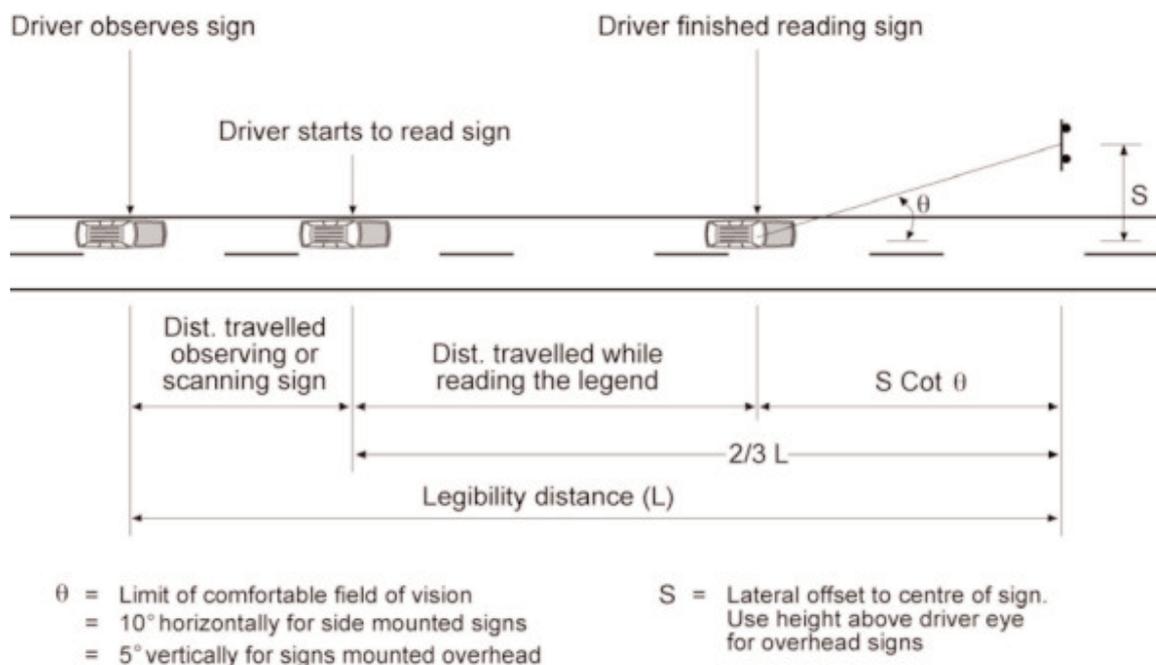


Figure 1.1: Distance de lisibilité

VI.7 Distance de lisibilité requise

La formule suivante peut être utilisée pour déterminer la distance totale de lisibilité nécessaire: [7]

- Monté sur le bord de la route : $L = 0.105NV - 8.55S$
- Monté au-dessus de la route : $L = 0.105NV - 17.1S$

Avec :

L = distance de lisibilité totale nécessaire (en mètre).

N = nombre de mots sur l'écran.

V = vitesse d'approche en km / h

S = décalage entre la position de l'œil du conducteur et le centre de panneau.

VI.8 Distance de lisibilité réelle

La distance de lisibilité réelle est basée sur la taille des caractères utilisés sur les PMV. La distance de lisibilité réelle (m) = $0,6 \times H$ [6].

Avec :

H = la hauteur des caractères (mm).

Pour les PMV de type : A, B, et C, la distance de lisibilité réelle est citée dans le tableau suivant :

Tableau 1.2 distance de lisibilité réelle

Type du PMV	H	Distance de lisibilité réelle
A	150mm	90m
B	300mm	180m
C	400mm	240m

VI. Les PMV dans le monde

Il n'est pas possible de fournir dans notre rapport un aperçu complet de tous les PMV utilisés dans le monde. Par ailleurs, il est plus intéressant de mettre en évidence des nouvelles tendances et certains panneaux spécifiques.

VII.1 Une nouvelle tendance, l'affichage sous forme de diagramme

Il est possible d'afficher des informations sur la durée du parcours et l'état du trafic sur un panneau dédié recourant à certaines couleurs, graphismes et/ou à du texte. Il s'agit d'une information au contenu variable, dont le concept s'utilise partout dans le monde. Voici quelques exemples :

- Le Japon et la Corée du Sud se sont hissés à la pointe dans le développement de panneaux dédiés renseignant sur la durée du parcours et l'état du trafic, et fournissant ainsi des informations sur le réseau.

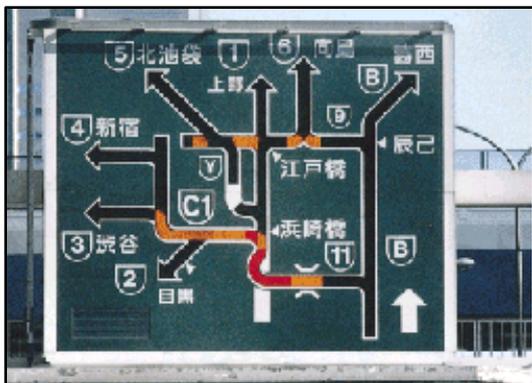


Figure 1.2 (a) : PMV spécial au Japon



Figure 1.2 (b) : PMV spécial au Corée du sud

- la Chine utilise maintenant des panneaux graphiques (en particulier à Pékin et Shanghai)



Figure 1.3 : Panneau graphique en chine

• certains pays européens ont adopté ce type d'affichage ces dernières années. Voici quelques exemples au Pays-Bas et en Espagne :



Figure 1.4 (a) : Panneau graphique au Pays-Bas

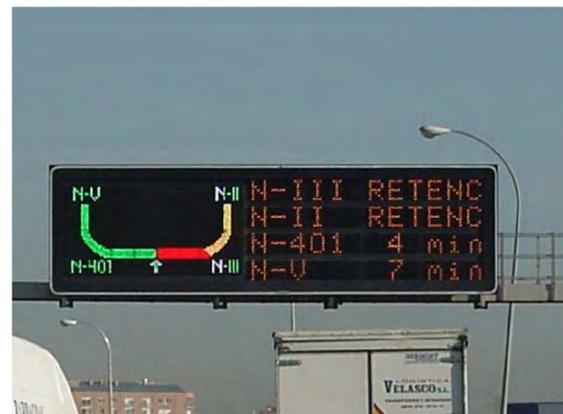


Figure 1.4 (b) : Panneau graphique en Espagne

• les PMV peuvent aussi afficher des informations numériques comme l'illustrent ces deux exemples photographiés au Danemark et en France.



Figure 1.5 (a) : Affichage numérique en Danemark



Figure 1.5 (b) : Affichage numérique en France

- La forte demande d'informations multimodales a entraîné l'apparition de PMV multimodaux :



Figure 1.6 (a): Panneau multimodal en Allemagne

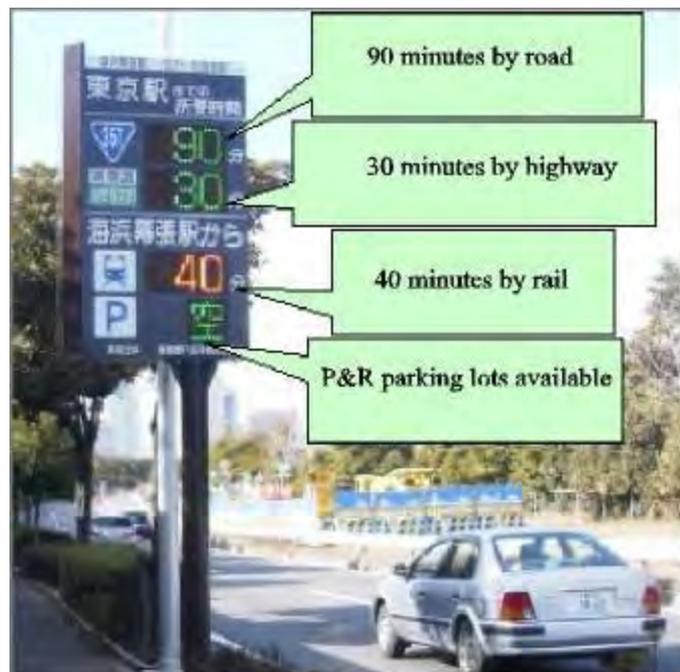


Figure 1.6 (b): Panneau multimodal au Japon

« 90 minutes par la route ; 30 minutes par l'autoroute ; 40 minutes par le train ;
emplacements disponibles en parking. »

Plusieurs recommandations visant ces types de panneaux cités ont été faites pendant un atelier d'experts européens tenu à Heathrow en mars 2003. Les principales recommandations figurent ci-dessous : [8]

- Il faudrait utiliser des lignes/blocs rouges pour désigner les gros bouchons.
- Il faudrait utiliser des lignes/blocs jaunes pour désigner les ralentissements.

- L'absence d'éclairage assortie d'une mention sur la durée du parcours devrait servir à indiquer que le trafic s'écoule fluide. A titre d'alternative, la couleur verte là où l'absence de lumière ne convient pas pourrait servir à signaler que le trafic est fluide.

- Il faudrait approfondir les recherches pour déterminer si l'utilisation de texte avec un code-couleurs aide à indiquer les durées de parcours sur les itinéraires les plus courts.

Les panneaux basés sur des tronçons devraient afficher la durée totale du parcours entre l'endroit actuel et la destination. S'il est proposé d'afficher les durées de parcours sur des tronçons particuliers, il faudrait entreprendre des recherches pour savoir si l'information est clairement comprise et si elle n'est pas préjudiciable à la compréhension des panneaux affichant la durée totale du parcours.

VII.2 Panneaux spécifiques

Il vaut la peine de mettre en évidence un certain nombre de panneaux utilisés dans certains pays.

- l'utilisation de PMV du type « image vidéo » en Corée du sud fournit une certaine quantité d'informations sur la circulation, y compris, comme sur la figure suivante, l'état actuel du trafic quelques kilomètres plus loin



Figure 1.7 : Panneau à image vidéo en Corée du Sud

- l'utilisation de 2 pictogrammes avec du texte (en Italie) :



Figure 1.8 : Pictogramme avec texte en Italie

VII.3 Différents points intéressants visant l'utilisation des PMV

Historiquement une séparation stricte avait été maintenue entre l'utilisation du texte et celle de pictogrammes. Dans le cas des systèmes de pilotage des voies (tels celui illustré par la figure 1.9), on recourt à des pictogrammes pour signaler des restrictions et/ou avertissements. Dans la plupart des cas, ces pictogrammes proviennent du code de la route.

Normalement, on ne se sert pas de panneaux à texte pour gérer les voies, mais pour afficher des messages d'avertissement et/ou d'information sur l'état du trafic. Dans la plupart des cas, la partie textuelle de ces panneaux ne convient pas pour afficher des pictogrammes. Dans de nombreuses applications toutefois, un ou plusieurs pictogrammes séparés sont installés sur le(s) côté(s) des panneaux. Ceci permet de « faire passer » graphiquement le contenu du message d'une façon à le rendre encore mieux compréhensible, en particulier aux conducteurs étrangers (figure 1.9).



Figure 1.9 : Utilisation combinée de textes et de pictogrammes

Vu que les PMV modernes utilisent souvent une technologie avancée (plus de LED, pixels plus serrés), il est également devenu possible d'utiliser des symboles graphiques plus sophistiqués, incrustés dans le texte, tels que :



signifiant respectivement « carrefour », « sortie », « bouchon », « chantier », « pont mobile ouvert » et « accident ».



Figure 1.10 : Utilisation de symboles graphiques

Et enfin les PMV librement programmables fournissent encore plus de possibilités pour combiner des pictogrammes, du texte et des pictogrammes au trait comme sur la figure

(1.11). Il est toutefois conseillé de restreindre cette « liberté » au moyen d'un règlement adéquat sur la façon dont ces pictogrammes devraient être utilisés pour éviter d'obtenir images indésirables pour les autorités routières ou inutilement complexes .



Figure 1.11 : PMV librement programmables

VII. Conclusion

Dans ce chapitre nous avons fait une brève étude sur les panneaux routiers, plus précisément les panneaux routiers à messages variables. Nous avons aussi cité les normes de taille, d'emplacement et d'affichage pour ces panneaux afin d'appliquer ces normes lors de la conception et la réalisation de notre projet. Nous avons également cité quelque exemples sur les panneaux spéciaux qui sont très peu utilisés pour le moment, mais ils pourront être les panneaux du futur.

Chapitre II : *Etude et présentation de la carte et* *ses composants*

I. Introduction

Pour l'affichage d'un message variable sur un tableau de taille bien défini, la carte doit sauvegarder le message et continuer à l'afficher tant qu'il n'y a pas un nouveau message. Sur cette carte on doit utiliser des composants efficaces et moins coûteux et s'assurer que ces composants sont convenables et peuvent communiquer entre eux.

II. Les différents modules

Notre carte peut être divisée en deux parties, une partie de contrôle, et une partie d'affichage (ou carte d'affichage).

II.1 Carte de contrôle

Cette carte a pour rôle de contrôler la carte d'affichage afin de gérer l'affichage, elle contient « le cerveau » du système, qui est un microcontrôleur capable d'enregistrer les messages à afficher et commander les matrices d'affichage en temps réel. La carte contient aussi un driver pour augmenter le courant passant dans les matrices d'affichage pour atteindre la valeur nominale qui fait éclairer les LED de chaque matrice.

La carte de contrôle peut lire les données à afficher d'un ordinateur via la connexion série RS232, ou bien directement d'un clavier de type PS/2, et elle peut aussi afficher le message localement sur un afficheur LCD pour visualiser le message à afficher avant de l'envoyer vers le panneau à LED.

II.1.1 Microcontrôleur PIC 18F252

Les microcontrôleurs PIC (ou PICmicro dans la terminologie du fabricant) forment une famille de microcontrôleurs de la société Microchip. Ces microcontrôleurs sont dérivés du PIC1650 développé à l'origine par la division microélectronique de General Instrument.

Le nom PIC n'est pas officiellement un acronyme, bien que la traduction en « **P**eripheral **I**nterface **C**ontroller » (contrôleur d'interface périphérique) soit généralement admise.

Cependant, à l'époque du développement du PIC1650 par General Instrument, PIC était un acronyme de « **P**rogrammable **I**ntelligent **C**omputer » ou « **P**rogrammable **I**ntegrated **C**ircuit ».

Un microcontrôleur est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants annexes. Un microcontrôleur peut donc fonctionner de façon autonome après programmation.

Les PIC intègrent une mémoire de programme, une mémoire de données, des ports d'entrée-sortie (numériques, analogiques, MLI, UART, bus I²C, etc.), et même une horloge, bien que des bases de temps externes puissent être employées. Certains modèles disposent de port et unités de traitement de l'USB.

Pour réaliser la carte de commande on choisit le PIC 18F252 qui appartient à la famille des PIC 18F, Cette famille introduite par MicroChip en 2000 a un jeu d'instructions plus complet puisqu'il en comprend quelques 75. Cette palette d'instructions étendue lui permet de faire fonctionner du code C compilé de manière nettement plus efficace que les familles précédentes. Sur les dernières versions (sous-famille "K"), on peut les utiliser avec un quartz fonctionnant jusqu'à 64 MHz (16 MIPS). [9]

Cette famille propose une multitude de dérivés intégrant l'USB, ETHERNET, le CAN, des canaux de MLI dédiés au contrôle moteur.

Tableau 2.1: Caractéristiques générales du PIC 18F252

PIC18F252	
contrôleur Famille / Série	PIC18
Nombre de bits	8
Nombre de broches	28
Nombre d'E/S	22
Taille mémoire de programme	16 kWords
Taille EEPROM	256Byte
Taille mémoire, RAM	1536Byte
Vitesse de processeur	40 MHZ
Type d'oscillateur	External
Nombre de timers	4
Périphériques	ADC, PWM, Timer
Nombre de voies PWM	2
Type de boîtier CI numérique	SDIP
Tension d'alimentation Maximale	5.5 V
Tension d'alimentation Typique	5.0 V
Tension de fonctionnement Minimum	4.2 V
Température de fonctionnement Maximum	+85°
Température de fonctionnement Minimum	-40°
Caractéristiques du MCU	2 PWM, WD , ISP, 2 CCP
Interface	I2C, SPI, USART, 3-Wire
Marquage composant	PIC18F252-I/SP
Nombre d'entrées CAN	5
Nombre de bits CAN	10
Nombre de MIPS	10
Nombre de bits	8
Hauteur du produit	3.3mm
Largeur du produit	7.24mm
Longueur du produit	34.67mm
Montage	Montage traversant
Taille mémoire étendue Maximum	2Mo

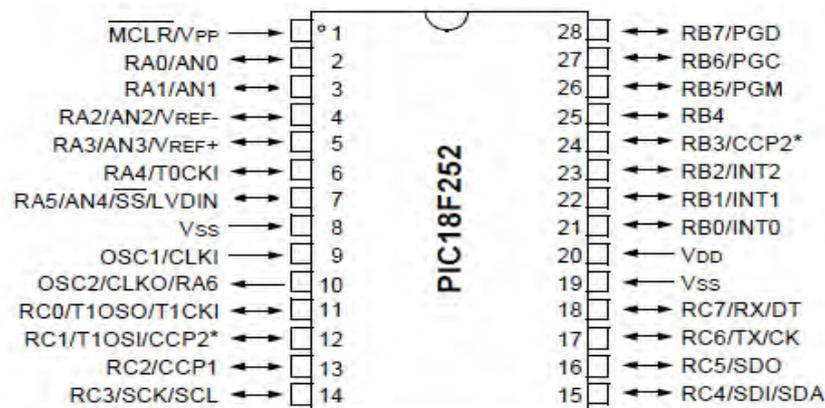


Figure 2.1: Schéma des broches du PIC 18F252

II.1.2 Le driver ULN2804

C'est un driver pour alimenter les matrices à LED. Il est composé de 8 transistors NPN Darlington. C'est un composant idéal pour faire l'interfaçage entre les circuits numériques de faible puissance (TTL, CMOS ou PMOS/NMOS) et les composants exigeant un niveau de courant/tension plus important, tel que les lampes, relais, marteaux d'impression ou d'autres charges similaires.

C'est un circuit intégré optimal pour alimenter les matrices d'affichage car le PIC ne peut les alimenter toutes.

Tableau 2.2 : Caractéristiques générales de l'ULN2804

Paramètre	Valeur Maximale
Tension de sortie	50 V
Tension d'entrée	30 V
Courant de collecteur	500 mA
Courant de base	25 mA
Température de fonctionnement ambiante	[0,70] °c
Température de stockage	[-55,150] °c
Température de jonction	+125 °c

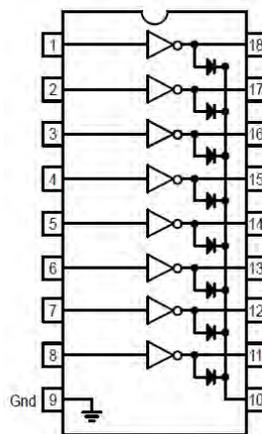


Figure 2.2 : Schéma des broches de l'ULN2804

❖ Schéma d'un bloc :

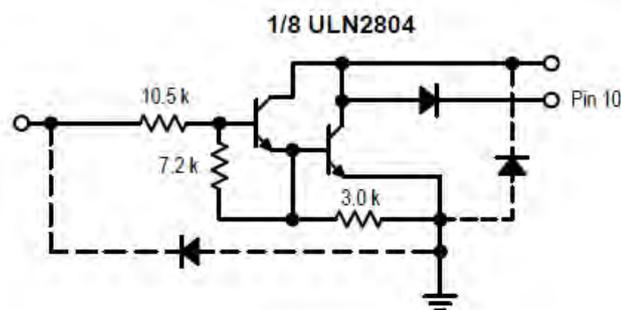


Figure 2.3 : schéma illustratif d'un bloc de l'ULN2804

II.1.3 L'afficheur LCD

Le PVC200401PTN - 20X4 est un afficheur LCD qui permet de visualiser le message à afficher avant de l'envoyer vers la carte d'affichage (Matrices à LED).

Il peut afficher un message composé de 4 lignes de 20 caractères sur chaque ligne, chaque caractère est composé de 8 X 5 pixels.

Tableau 2.3 : Caractéristiques générales du PVC200401PTN

Afficheur LCD	
Digits / Alpha	80
Compteur Ligne / caractère	20 x 4
Rétro-éclairage de couleur	Blanc
Taille de caractère	5.56 mm
Tension, alimentation	5 V
Mode d'affichage	Transflectif
Largeur, surface d'affichage	64.5 mm
Hauteur, surface d'affichage	16 mm
Température de fonctionnement	0°C à +50°C
Rétro-éclairage	Bleu
Type de connecteur	B, 1 x 16
Couleur, police	Blanc
Largeur (externe)	84 mm
Largeur, caractère	2.94 mm
Longueur/hauteur	44 mm
Nombre de colonnes	16
Nombre de rangées	2
Profondeur	10 mm
Technologie d'afficheur	LCD
Tension, alimentation	5VDC

II.1.4 Clavier PS2

Le port PS/2 (sigle de **P**ersonal **S**ystem/2) ou port Mini-din est un port de connexion de dimensions réduites pour souris ou clavier.

Il est apparu avec les ordinateurs IBM PS/2 vers 1987. Cependant, le port PS/2 ne s'est généralisé que plus tard avec l'introduction de la norme ATX pour carte mère en 1995.

Auparavant le clavier se branchait sur un connecteur DIN tandis que la souris se branchait sur un port série, ou sur un port propriétaire.

En 2010, les ports PS/2 sont devenus obsolètes, le port USB étant utilisé en remplacement. Néanmoins la technologie du port PS/2 permet de fabriquer des claviers sans pertes d'information en cas de frappes multiples et simultanées de touches, chose impossible techniquement avec l'USB. On peut également utiliser des claviers et souris sans fil grâce à la technologie Bluetooth [10].

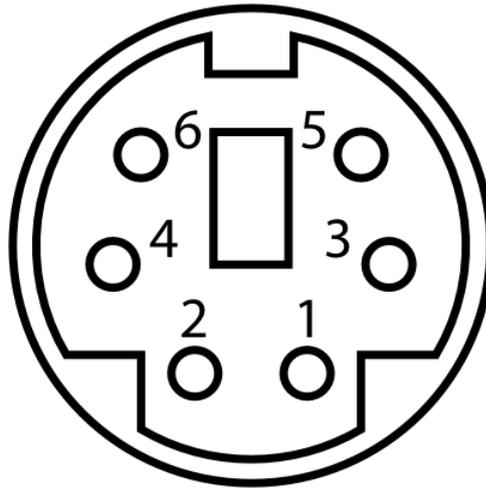


Figure 2.4: Une entrée PS/2

- Broche 1 : Données, fil rouge ou vert
- Broche 2 : Non connecté, fil vert
- Broche 3 : 0V (GND), fil blanc
- Broche 4 : + 5V, fil jaune
- Broche 5 : Horloge, fil noir
- Broche 6 : Non Connecté, fil bleu

II.1.5 La liaison RS232 (DB9)

RS-232 (parfois appelée EIA RS-232, EIA 232 ou TIA 232) est une norme standardisant un bus de communication de type série sur trois fils minimum (électrique, mécanique et protocole). Disponible sur presque tous les PC jusqu'au milieu des années 2000, il a été communément appelé le « port série ». Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont désignés par les noms COM1, COM2, etc. Cela leur a valu le surnom de « ports COM », encore utilisé de nos jours. Cependant, il est de plus en plus remplacé par le port USB [11].

Les liaisons RS-232 sont fréquemment utilisées dans l'industrie pour connecter différents appareils électroniques (automate, appareil de mesure, etc.).

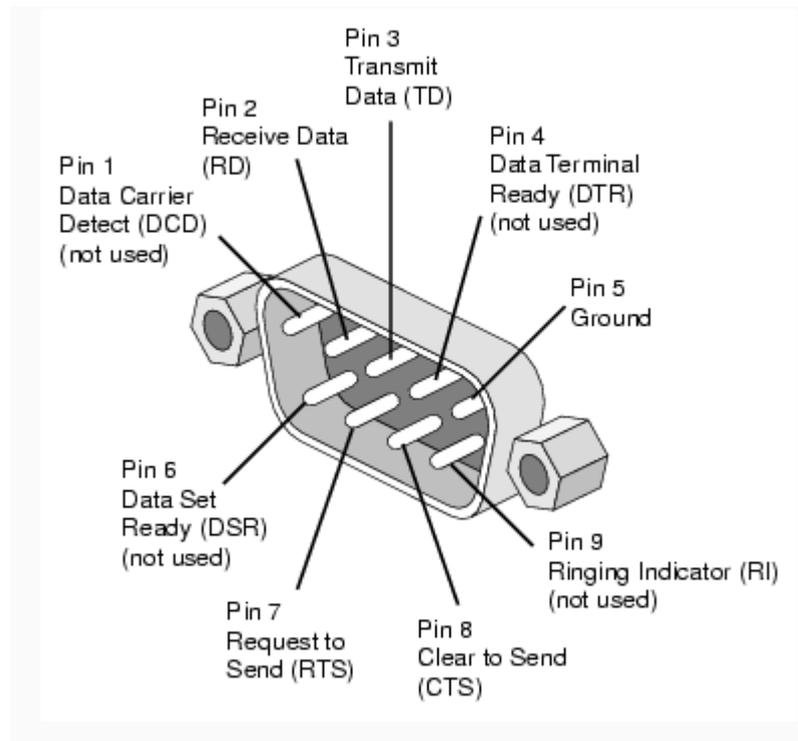


Figure 2.5 : Une sortie RS232

Broche N°1 : **DCD** (Data carrier detect), 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. Les données de l'ordinateur vers le correspondant sont véhiculées par son intermédiaire.

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 [12].

le microcontrôleur ne peut générer que des signaux [0v, 5v]. Un problème d'adaptation de tension se pose : on ne peut pas lui relier directement les fils série [-12v, +12v] provenant du PC.

On insère donc entre la prise série et le microcontrôleur un circuit adaptatif , c'est le **MAX232**

❖ **Le MAX 232 :**

Ce convertisseur permet de convertir les signaux des communications séries bidirectionnelles, entre des entrées/sorties TTL et un port série COM RS232 d'un PC.

Ce convertisseur peut être utilisé avec tous les microcontrôleurs PIC et Atmel, ou avec n'importe quel autre microcontrôleur disposant d'entrées/sorties séries TTL ayant besoin d'être converties vers un port RS232 [13].

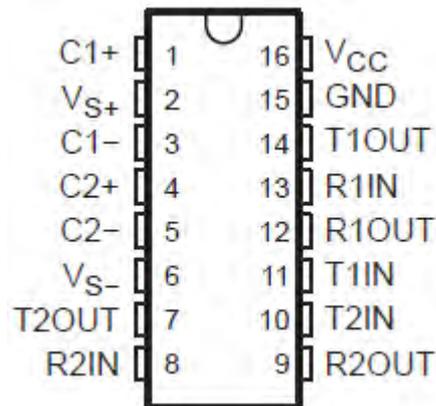


Figure 2.6 : Schéma des broches du MAX232

❖ **Montage :**

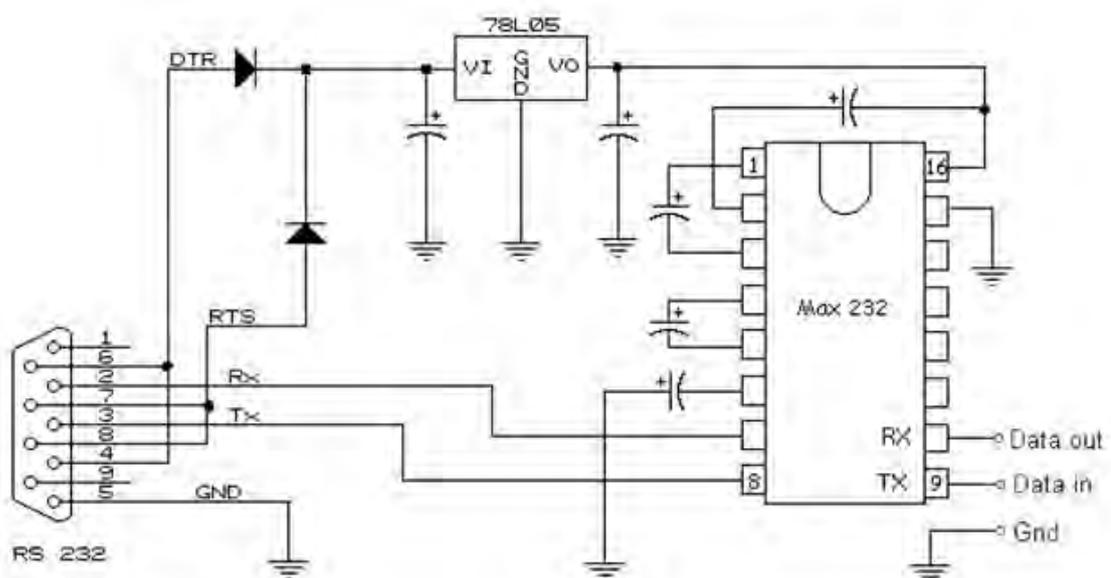


Figure 2.7 : Schéma du montage d'un MAX232

Donc on obtient un connecteur DB9 allant vers la carte comme suit :

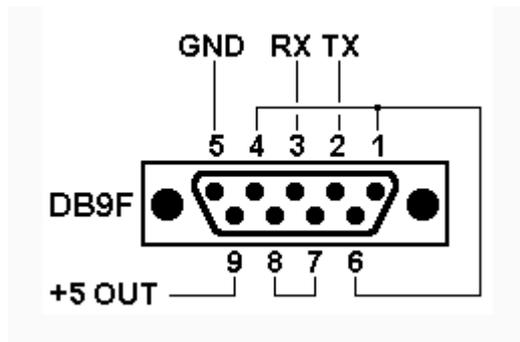


Figure 2.8 : connexion RS232 utilisant le MAX232

II.2 Carte d'affichage

Cette carte a pour rôle l'affichage du message envoyé par la carte de contrôle, elle contient essentiellement deux composants :

II.2.1 Matrice à LED HS2088B

La matrice HS2088B est une matrice à LED avec 8 lignes et 8 colonnes, et donc elle contient 64 LED de couleur rouge.

Le schéma interne de cette matrice est donné par la figure suivante :

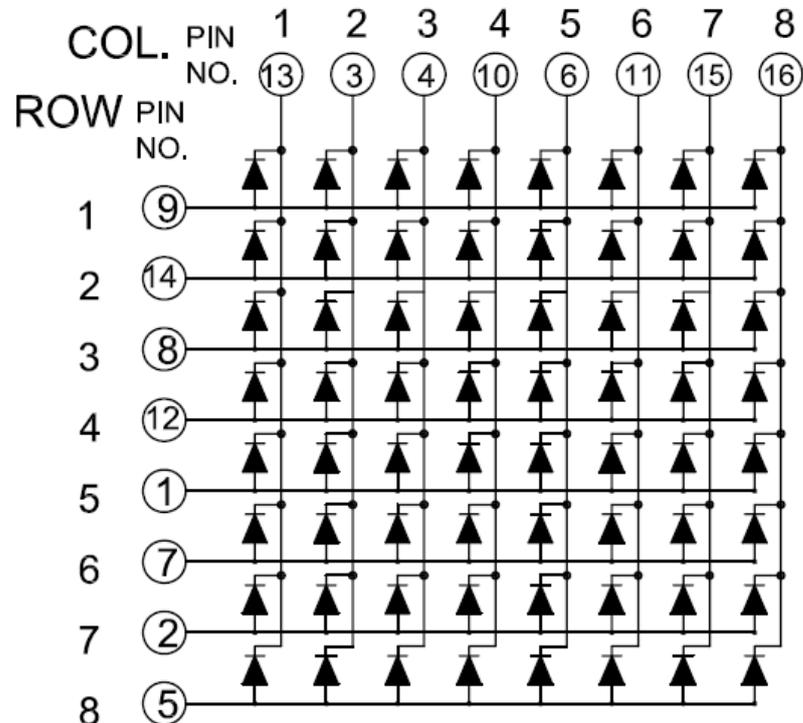


Figure 2.9 : Schéma interne de matrice HS2088B

La HS2088B est une matrice carrée, la largeur de chaque matrice est de 60.20 mm, on peut attacher plusieurs matrice pour former des tableaux plus grands, les autres dimensions de HS2088B sont montrés sur les figures suivantes :

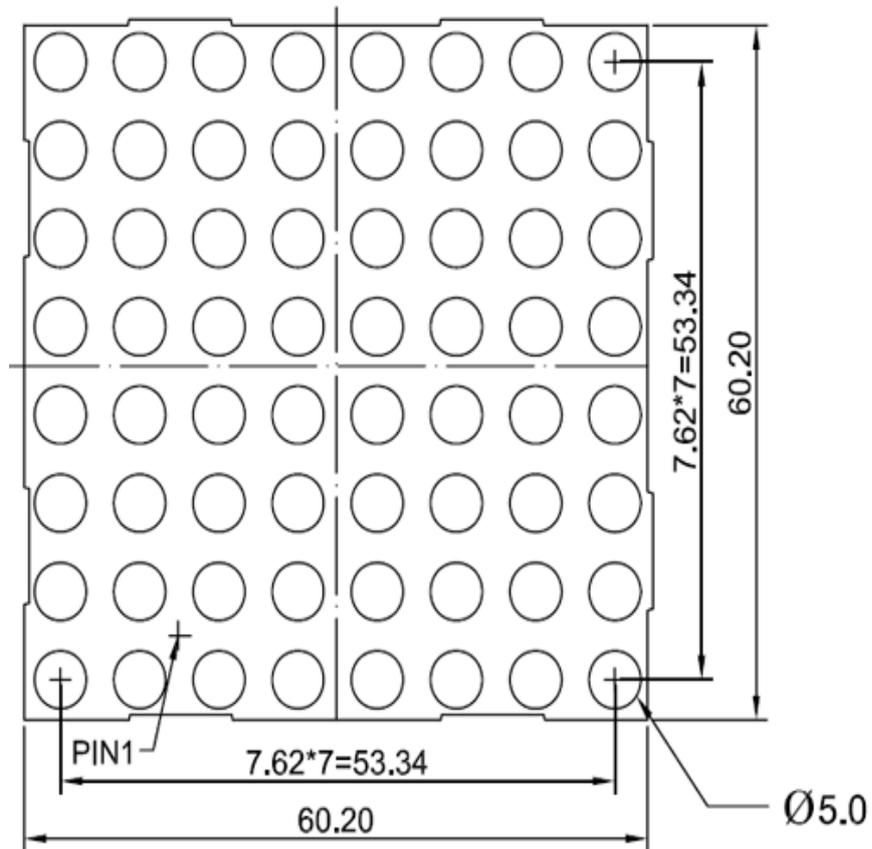


Figure 2.10.(a) : dimensions de la matrice HS2088B

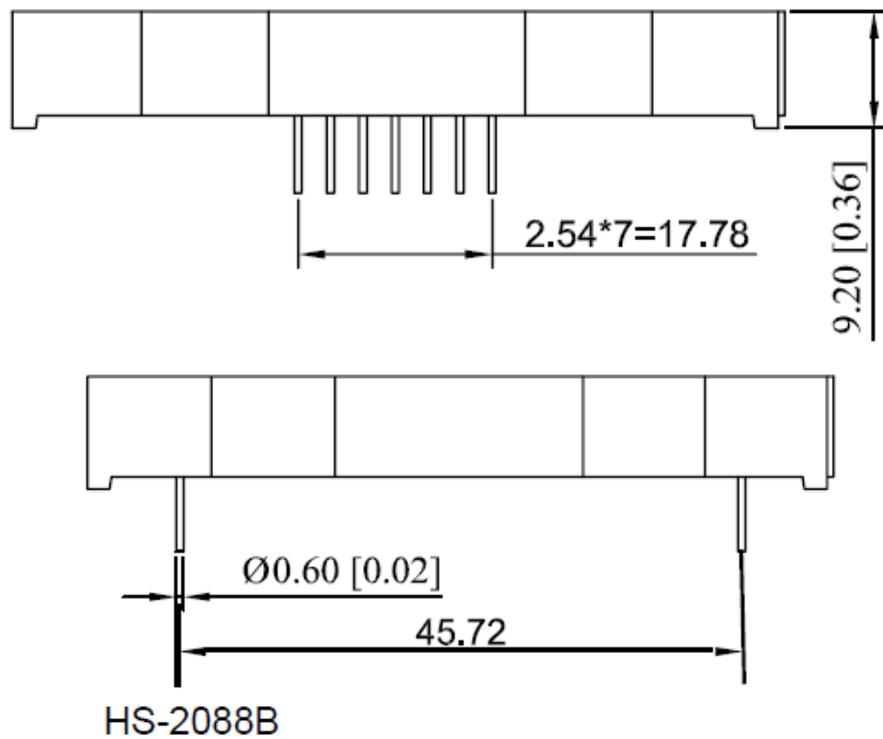


Figure 2.10.(b) : Dimensions de la matrice HS2088B

Les caractéristiques électriques de la HS2088B sont indiqués sur le tableau 2.4 et la figure 2.11 :

Tableau 2.4 : Les caractéristiques électriques de matrice HS2088B

Paramètre	Symbole	Min	Typique	Max	Unité	Conditions du test
Intensité lumineuse	I_V	-	4.5	-	mCd	$I_F = 10\text{ma}$
Tension directe	V_F	-	2.1	25	V	$I_F = 20\text{ mA}$
Longueur d'onde	λ_P	-	668	-	nm	$I_F = 20\text{mA}$
Courant inverse	I_R	-	-	50	uA	$V_r = 5\text{V}$

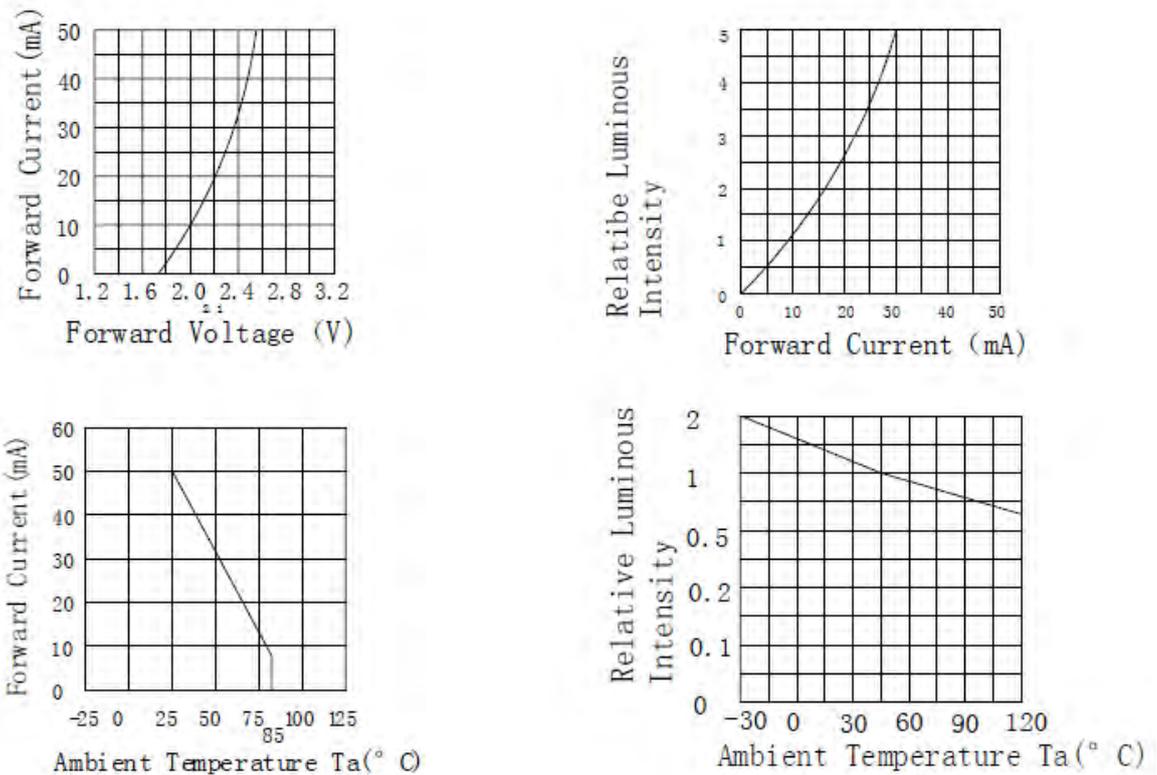


Figure 2.11: Les caractéristiques électriques de matrice HS2088B

II.2.2 Le registre à décalage 74HC595

La famille 74HC est une famille de circuits intégrés CMOS en boîtier DIP ou CMS.

Elle présente les caractéristiques suivantes:

- très faible dissipation de puissance.
- tensions de seuils: 30% et 70% de V_{cc} .
- haute immunité au bruit.
- alimentation de 2V à 6V.
- même brochage que la série TTL.

Le registre à décalage 74HC595 est un circuit à 16 broches comme montre le schéma suivant :

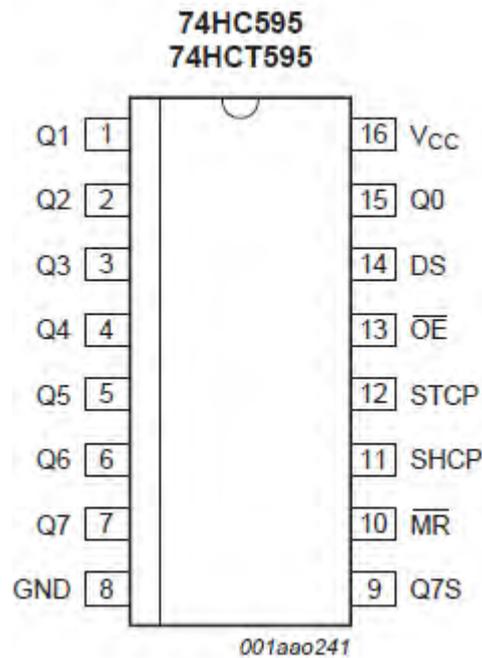


Figure 2.12 : Schéma des broches du 74HC595

Tableau 2.5 : Broches du 74HC595

Symbole	Pin	Description
Q1	1	Sortie parallèle 1
Q2	2	Sortie parallèle 2
Q3	3	Sortie parallèle 3
Q4	4	Sortie parallèle 4
Q5	5	Sortie parallèle 5
Q6	6	Sortie parallèle 6
Q7	7	Sortie parallèle 7
GND	8	La masse (0 V)
Q7S	9	Sortie série
MR	10	Remise à zéro
SHCP	11	Clock
STCP	12	Latch
OE	13	L'entrée active la sortie
DS	14	Entrée de donnée
Q0	15	Sortie parallèle 0
VCC	16	Alimentation +5V

Et il présente les caractéristiques suivantes :

- 8 bits d'entrée en série
- 8 bits de sortie en série ou en parallèle
- Registre de stockage avec 3 états de sorties possibles.
- Possibilité de remise à zéro
- Fréquence de changement d'état de 100 MHz (typique)

- Protection ESD
- Température de fonctionnement de -40 °C à +85 °C

Un chronogramme élémentaire décrivant le fonctionnement du circuit est montré sur la figure 2.13 [14]

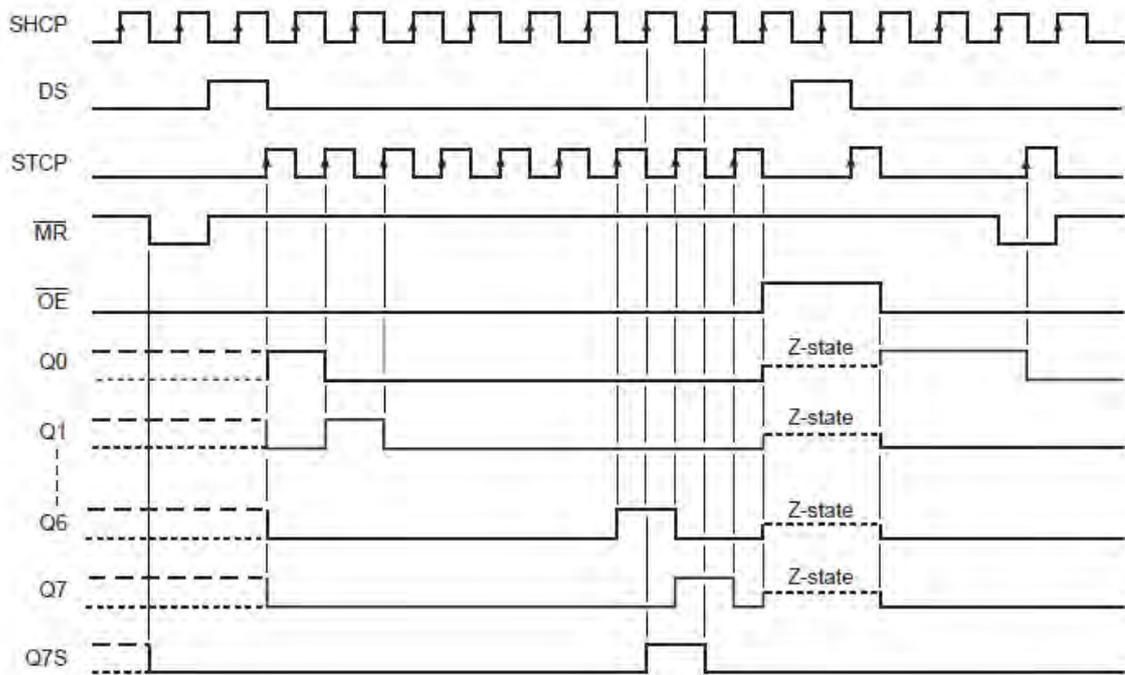


Figure 2.13 : chronogramme élémentaire du 74HC595

III. Conclusion

Pour réaliser ce projet nous avons eu à étudier les différents composants qui constituent nos cartes, pour s'assurer que tous les composants utilisés sont compatibles entre eux, et si ce n'était pas le cas il faut trouver une solution pour adapter le problème. Après notre brève étude sur les différents composants en se basant sur leurs DATASHEET, on peut dire que tous les problèmes d'adaptation sont résolus, et les composants utilisés sont bien compatibles entre eux et capables de faire fonctionner nos deux cartes qui forment notre prototype.

Chapitre III : Conception, réalisation, tests et validation

I. Introduction

L'objectif principal de ce chapitre est la conception et la réalisation :

- D'une carte électronique de commande pour le contrôle d'un afficheur à base de LED, et ses différents périphériques d'entrée et de sortie (clavier, LCD ... etc.)
- D'une carte d'affichage à plusieurs matrices pour afficher les messages reçus par la carte de commande.

II. Principes de fonctionnement et routage

II.1 La carte de commande

La carte est composée de plusieurs blocs tel qu'illustrée sur la figure (3.1). Les fonctions que devra remplir cette carte sont : premièrement la réception du message et la génération d'un code approprié pour chaque lettre de ce message ; et deuxièmement l'affichage du message localement sur un petit écran LCD et l'envoi du même message vers la carte d'affichage.

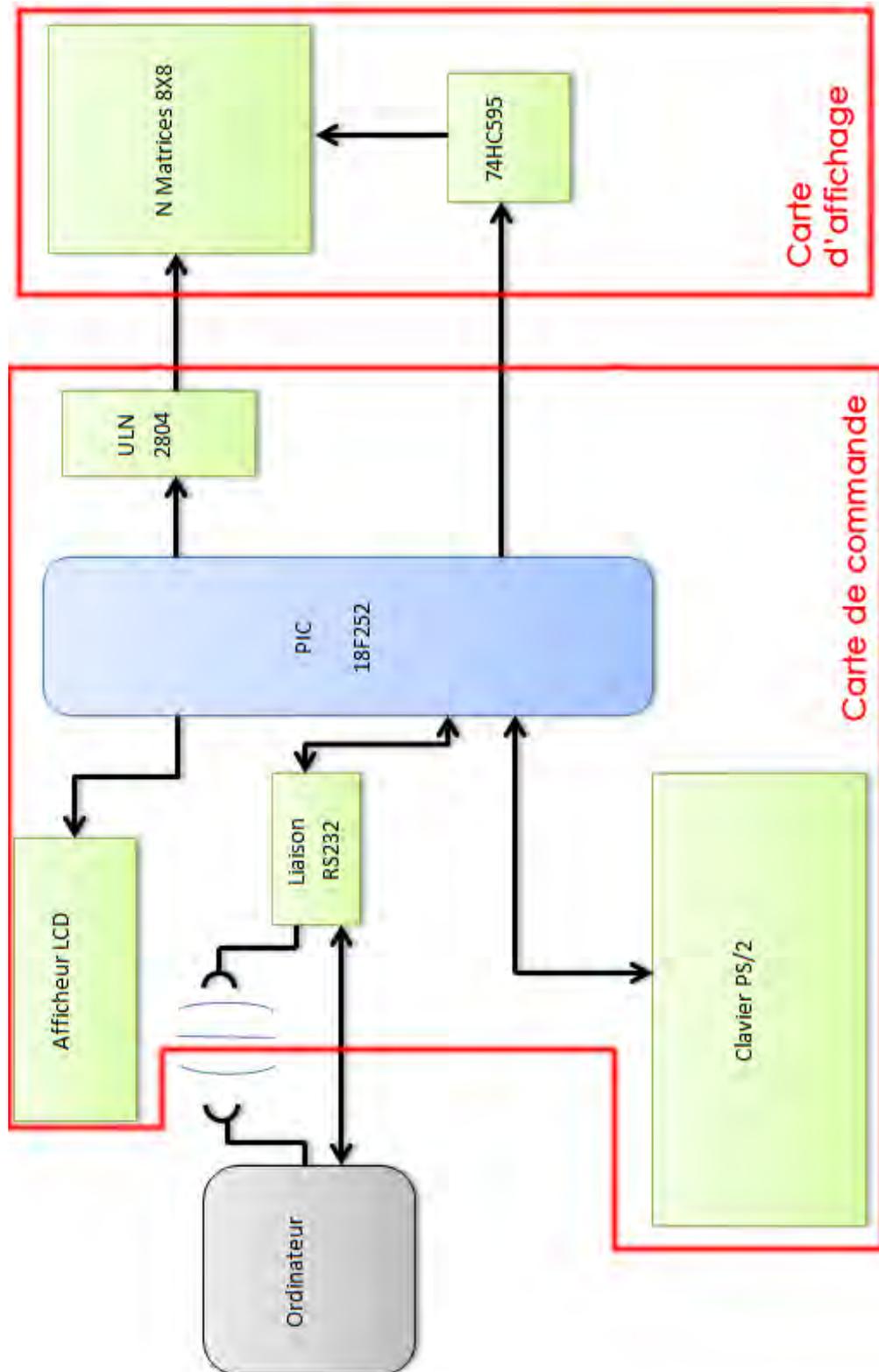


Figure 3.1: schéma bloc de la carte globale

II.1.1 Le microcontrôleur

Il représente la partie responsable de la lecture et de l'affichage de notre message .

On a choisi un microcontrôleur de type PIC de MicroChip, c'est le microcontrôleur 18F252, ce microcontrôleur est la solution optimale pour garantir le fonctionnement de notre

projet, car avec ses 22 pins d'entrée/sotie, il est bien capable de gérer toute la carte (voir le chapitre précédent).



Figure 3.2: Le PIC 18F252

Le PIC ne nécessite pas un câblage particulièrement complexe pour l'alimenter. Il faut juste relier les 2 pins GND à la masse (pin 8 & 19), et relier la pin 20 au +5V, la pin 1 (\overline{MCLR}) au +5V à travers une résistance (R9) de 10 k Ω pour limiter le courant et forcer cette entrée à « 1 logique » [Annexe 1].

Les 6 pins du port A du microcontrôleur (RA0 – RA5) sont reliées avec un afficheur LCD pour afficher le message qui est dans la mémoire du microcontrôleur et aussi les différents commentaires liés au notre programme s'il le faut.

Les 8 pins du port B du microcontrôleur sont reliées au driver ULN2804 à travers ces 8 entrées (pin 1-8), pour contrôler les colonnes des matrices d'afficheur.

Et enfin le port C est utilisé comme suit :

RC0= comme entrée pour activer le clavier PS/2

RC1, RC2, et RC3 : sont utilisés comme des soties pour commander les différents registres à décalage qui sont au niveau de la carte d'affichage.

RC4, et RC5 sont utilisés comme des entrées pour lire la donnée envoyée par le clavier à travers une connexion PS/2.

Les pins 9 & 10 (OSC1 et OSC2 respectivement) sont connectées avec un oscillateur quartz de 20 MHz qui est lui-même connecté avec 2 capacités de 22 pF vers la masse [Annexe 1].



Figure 3.3: Quartz de 20 MHz

II.1.2 Le driver ULN2804



Figure 3.4: Driver ULN2804

Ce driver a pour but de fournir le courant nécessaire pour allumer les différentes LED de l'afficheur, vu que le microcontrôleur ne peut pas fournir un courant aussi important. [Annexe 1]

Il est connecté avec le port B du microcontrôleur d'une part, et d'autre part (en sortie) avec des résistances de $150\ \Omega$ pour protéger les LED contre le courant important fourni par le driver ULN2804.

La pin 9 de ce driver est liée à la masse, et la pin 10 est reliée avec le +5V (VCC).

II.1.3 L'afficheur LCD



Figure 3.5: Afficheur LCD

L'afficheur LCD possède 16 broches, les broches 1 et 5 sont reliées à la masse, la broche 5 au +5V, la broche 3 (VEE) est reliée au 5V à travers un potentiomètre de $5\ \text{k}\Omega$ pour contrôler le contraste de l'afficheur.

La broche 4 avec le RA5 du microcontrôleur, la broche 6 au RA4, la broche 11 au RA0, la broche 12 au RA1, la broche 13 au RA2, et la broche 14 au RA3 du microcontrôleur ; les broches 7 à 10 ne sont pas utilisées [Annexe 2].

II.1.4 Les connecteurs

Un connecteur PS/2 est relié aux pins RC4 & RC3 du microcontrôleur à travers ses broches 1 et 5 respectivement, ces connections sont aussi reliées avec le VCC (+5V) à travers deux résistances R 12 et R 13 de $10\ \text{k}\Omega$ pour faire un montage PULL-UP.

La broche 3 du PS/2 est reliée à la masse, la broche 4 au VCC, et la broche 6 n'est pas connectée [Annexe 3].



Figure 3.6: Connecteur PS/2

Au bout de notre carte de commande on branche un connecteur (8X2) qui transfère les signaux nécessaires pour faire fonctionner la carte d'affichage ; on trouve 3 signaux qui vont vers les registres à décalage de cette dernière et 8 signaux pour contrôler les colonnes des matrices, les broches restant sont connectées à la masse (3 d'entre eux), et les 2 autres sont reliées au +5V.



Figure 3.7 : Connecteur coudé

La carte finale de commande est présentée sur la figure suivante :

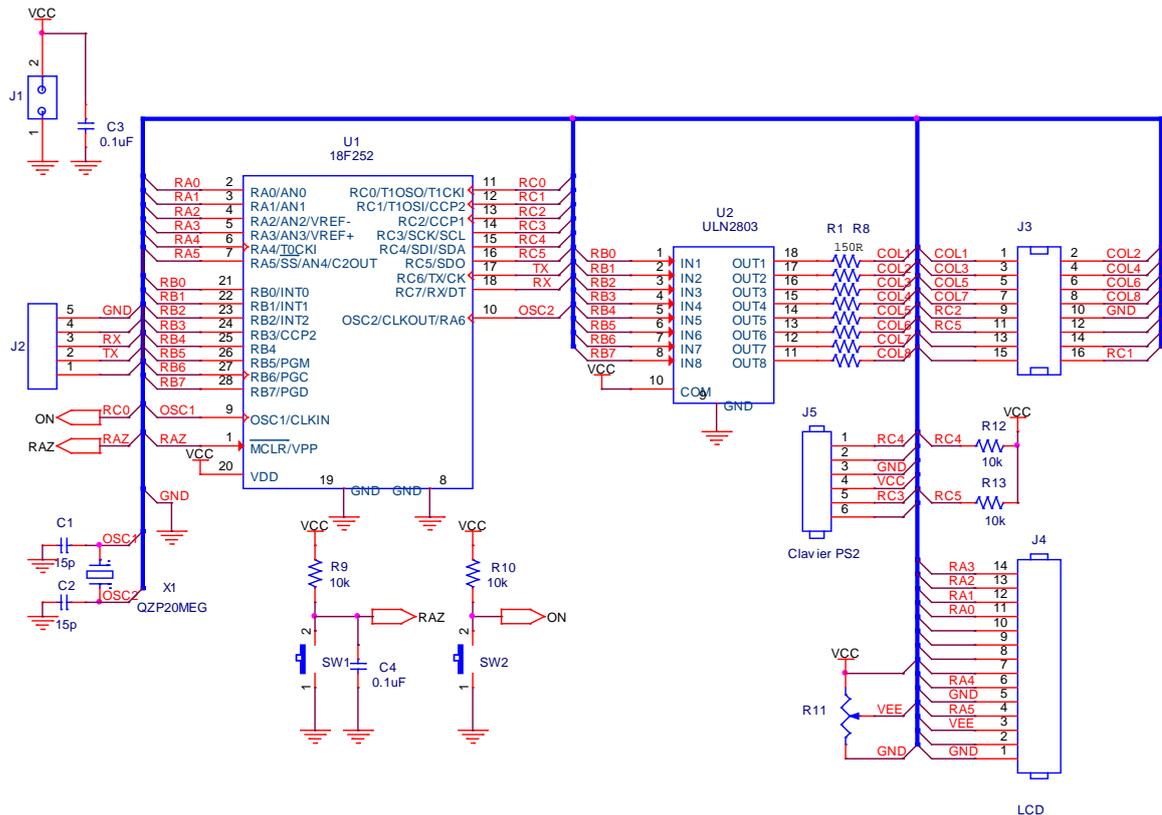


Figure 3.8: La carte de commande

- Le typon complet de cette carte est inclus dans l'annexe.

II.2 La carte d'affichage

La carte d'affichage est conçue principalement avec 4 matrices d'affichage et 4 registres à décalage pour les commander, elle peut être étendue pour contenir plus de 4 matrices.

La carte est aussi modulaire, elle est conçue d'une façon à permettre plusieurs cartes de 4 matrices d'être associées pour créer un tableau plus grand.

II.2.1 La matrice à LED



Figure 3.9: Matrice à LED (Dot Matrix)

C'est une matrice de 8 lignes et 8 colonnes, donc un total de 64 LED de couleur rouge ; les 8 LED d'une même ligne sont connectées en « anode commune » ainsi que les LED d'une

même colonne sont connectées en « cathode commune » ce qui signifie que pour allumer une LED particulière il faut fournir une tension positive sur la ligne où se trouve la LED, et brancher sa colonne à la masse (pour la distribution des pins en fonctions des colonnes et lignes voir chapitre précédent).

Pour pouvoir contrôler toutes les LED d'une matrice il faut utiliser ses 16 pins, donc pour contrôler 4 matrices il nous faut :

$$16 \times 4 = 64 \text{ pins du microcontrôleur.}$$

Une solution est de brancher toutes les colonnes des matrices ensemble ce qui réduit les pins nécessaire à 40, mais ça reste encore élevé.

On propose soit d'utiliser des démultiplexeurs pour s'adresser à chaque ligne d'une matrice (on dispose de 32 lignes), soit d'utiliser des registres à décalage qui permettent de contrôler un nombre théoriquement infini de ligne et donc de matrices avec seulement 3 pins de microcontrôleur.

La matrice à LED ne nécessite pas une alimentation indépendante, car elle ne contient que des LED et donc il faut juste envoyer une combinaison (ligne-colonne) particulière pour allumer une LED particulière, les autres LED restant éteintes.

En fin il nous faut juste :

$8+3 = 11$ pins du microcontrôleur pour adresser toutes les LED en utilisant une astuce qui est une caractéristique bien connue de l'œil humain : la « persistance rétinienne », qui est : *'la capacité ou défaut de l'œil à conserver une image vue superposée aux images que l'on est en train de voir. Elle est plus forte et plus longue si l'image observée est lumineuse. Ceci est dû en partie au temps de traitement biochimique du signal optique.'* [15]

C'est-à-dire qu'en réalité on ne s'adresse qu'à un nombre fini de LED dans un petit intervalle du temps, puis on passe vers un autre groupe de LED, mais avec la caractéristique de l'œil humain on a l'impression que ces différents groupes de LED sont traités en même temps et donc adressés en même temps.

Les matrices utilisées sont du type HS2088B (plus d'informations dans le chapitre II), chaque LED nécessite un courant direct $I_F = 20 \text{ mA}$ avec une tension directe typique de $V_F = 2.1 \text{ V}$ [Annexe 3]

Appliquant la loi d'Ohm sur le montage de figure (3.10) on trouve :

$$R = \frac{V_{CC} - V_F}{I_F} = \frac{5 - 2.1}{20 \times 10^{-3}} \approx 150 \Omega.$$

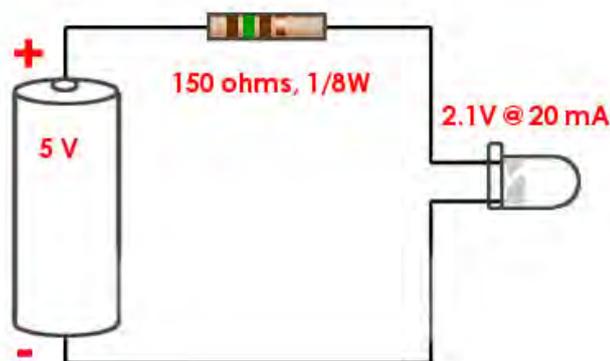


Figure 3.10: Alimentation d'une LED

Vu que l'on n'allume qu'une seule LED en même temps sur la colonne, cette valeur reste fixe et on n'a pas besoin de la changer chaque fois qu'on veut en allumer plus d'une.

Les résistances sont reliées au driver ULN2804 qui se trouve dans la carte de commande ; Le signal reçu au niveau de la carte d'affichage est déjà adapté pour assurer l'allumage des LED appropriées.

II.2.2 Le registre à décalage 74HC595

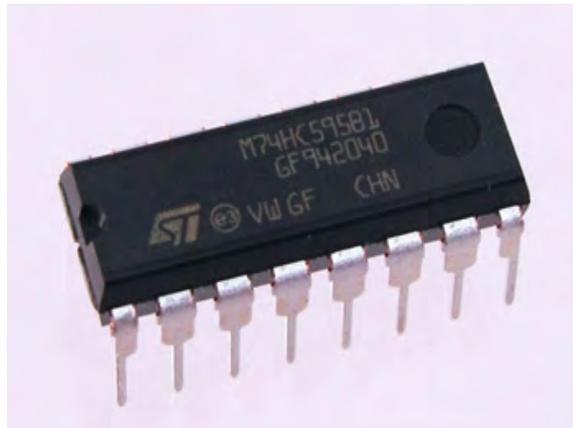


Figure 3.11: Registre à décalage 74HC595

Son rôle est de faire le balayage ligne par ligne de chaque matrice constituant la carte d'affichage, ainsi que minimiser le nombre de pins I/O pour l'adressage de toute la carte.

En premier lieu, on a pensé à utiliser le registre TTL bien connu 74LS595, mais avec des problèmes prévisibles concernant les pertes sur les longues distances on a décidé d'utiliser le registre à décalage CMOS 74HC595 qui est aussi plus rapide et peut supporter des fréquences atteignant le 100 Mhz [Annexe 4]

Le 74HC595 a 16 pins (DIP 16) dont la pin 8 est reliée à la masse et la pin 16 au +5V, la pin 14 appelée DS est la pin de donnée, elle est reliée avec la pin RC1 (12) du microcontrôleur à travers un connecteur. Les pins 11 et 12 du registre sont reliées respectivement avec les RC5 et RC2 du microcontrôleur.

La pin 10 est reliée au +5V et la pin 13 est reliée à la masse.

La pin 14 est une sortie, elle reproduit le signal original une fois le décalage est terminé, elle est reliée à l'entrée du deuxième registre pour effectuer un montage en cascade, i-e que la donnée envoyée par le microcontrôleur vers le premier registre va être re-envoyée vers le deuxième registre puis le troisième et ainsi de suite jusqu'au dernier registre.

Les pins restants sont des sorties parallèles (Q0 - Q7) ; elles sont reliées avec les lignes de la première matrice tel que la pin Q0 va vers la première ligne, Q1 vers la deuxième ligne... etc.

La carte finale d'affichage est présentée dans la figure suivante :

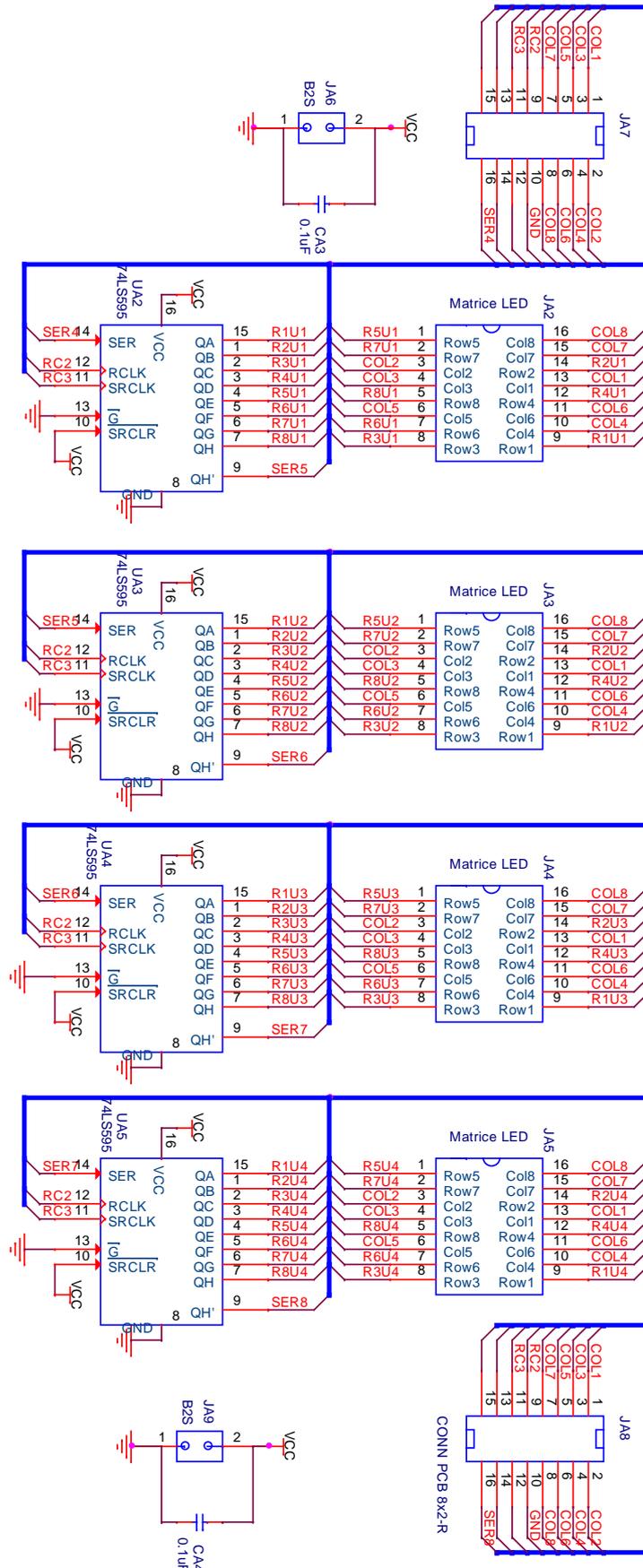


Figure 3.12: La carte d'affichage

- Le typon complet de cette carte est inclus dans l'annexe.

III. Réalisation et programmation

III.1 Programmation du PIC

Dotés d'une logique programmée, les microcontrôleurs sont capables de réagir à la manière d'automates programmables, leurs propriétés ne se limitent pas à offrir un certain nombre d'entrées/sorties logiques. Ils sont parfois dotés de fonctions supplémentaires telles que la conversion analogique numérique, l'horloge et le comptage rapide. Ces fonctions sont exploitées par la programmation du microcontrôleur en utilisant un langage de haut niveau ou un langage assembleur.

Dans notre cas, on va programmer le PIC en utilisant le langage de haut niveau C qui permet d'alléger considérablement le développement d'une application, et cela en respectant les étapes de programmation suivantes :

- L'élaboration de l'organigramme décrivant le fonctionnement du programme.
- La création du code source C du programme.
- La compilation du programme.
- Le chargement du programme sur le PIC.
- L'exécution de tests, et la validation du programme.

III.1.1 Le langage C

Le langage C est apparu au cours de l'année 1972 dans les Laboratoires Bell. Il était développé en même temps que l'UNIX, par Dennis Ritchie [1941 - 2011] et Kenneth Thompson [1943 - aujourd'hui].

Le langage C est un langage de programmation conçu pour la programmation système. Il est devenu l'un des langages les plus utilisés. De nombreux langages plus modernes comme C++, Java et PHP reprennent des aspects du C.

Les caractéristiques du C en font un langage privilégié quand on cherche à maîtriser les ressources utilisées. Ce langage est donc extrêmement utilisé dans des domaines comme la programmation embarquée sur microcontrôleurs, les calculs intensifs, l'écriture des systèmes d'exploitation et tous les modules où la rapidité de traitement est importante. Il constitue une bonne alternative au langage d'assemblage dans ces domaines, avec les avantages d'une syntaxe plus expressive et de la portabilité du code source [16].

III.1.2 Le compilateur mikroC PRO

Créer par mikroElektronika, mikroC PRO est un compilateur en langage C monoposte pour microcontrôleurs PIC bénéficie d'une prise en main très intuitive et d'une ergonomie sans faille. Ses très nombreux outils intégrés (mode simulateur, terminal de communication Ethernet, terminal de communication USB, gestionnaire pour afficheurs 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code, mode Débug ICD...) associé à sa capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie [17].

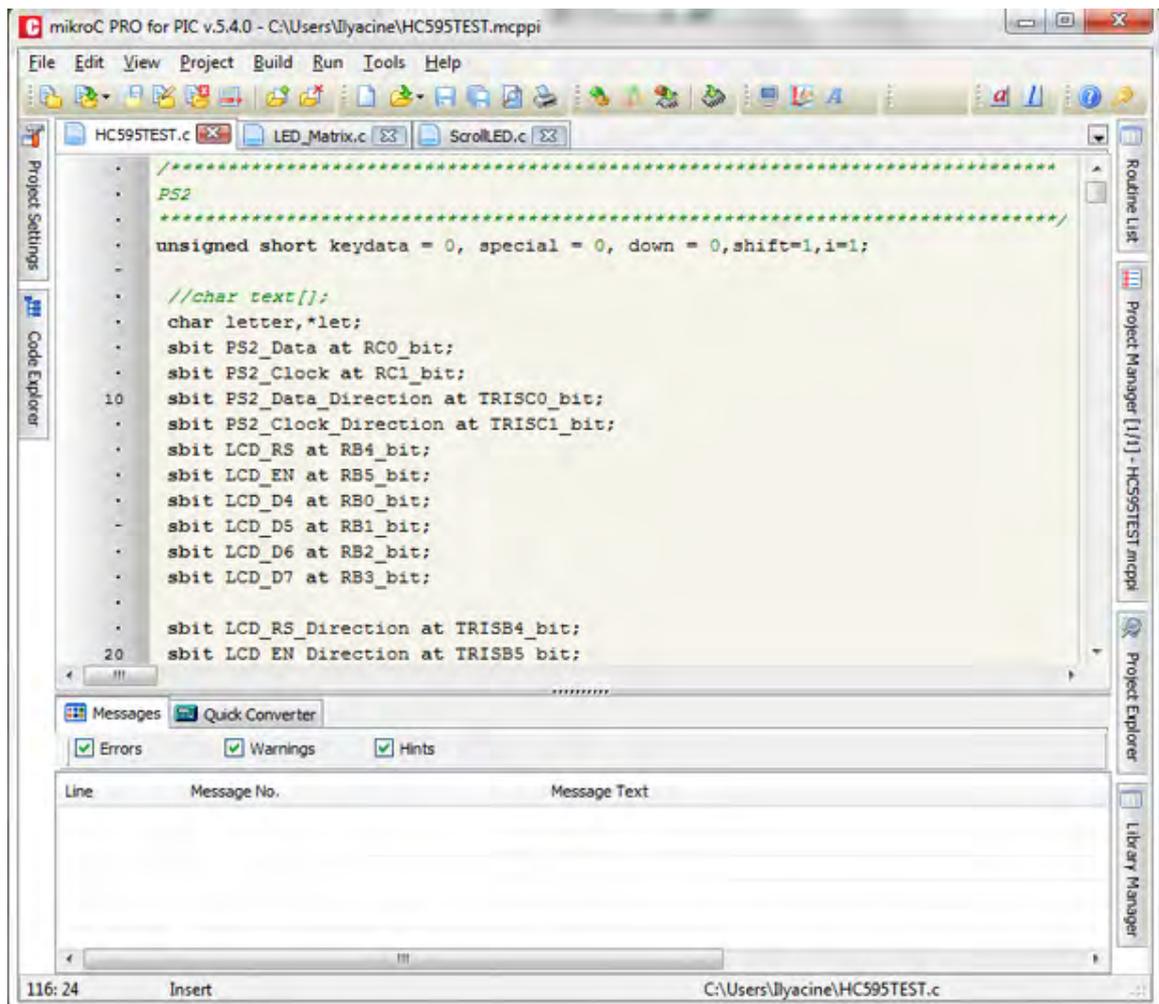


Figure 3.13: Fenêtre du compilateur MikroC PRO pour PIC.

III.1.3 Le programmeur PICFLASH

Le PICFLASH est un logiciel qui accompagne mikroC PRO et qui permet de charger le fichier programme au format (.hex) généré par le compilateur MikroC PRO dans le microcontrôleur.

Le microcontrôleur à programmer devrait être branché sur la carte de développement EASYPIC qui contient un bloc de programmation hardware des PICs. Il est très recommandé de formater le PIC avant toute nouvelle programmation [18].

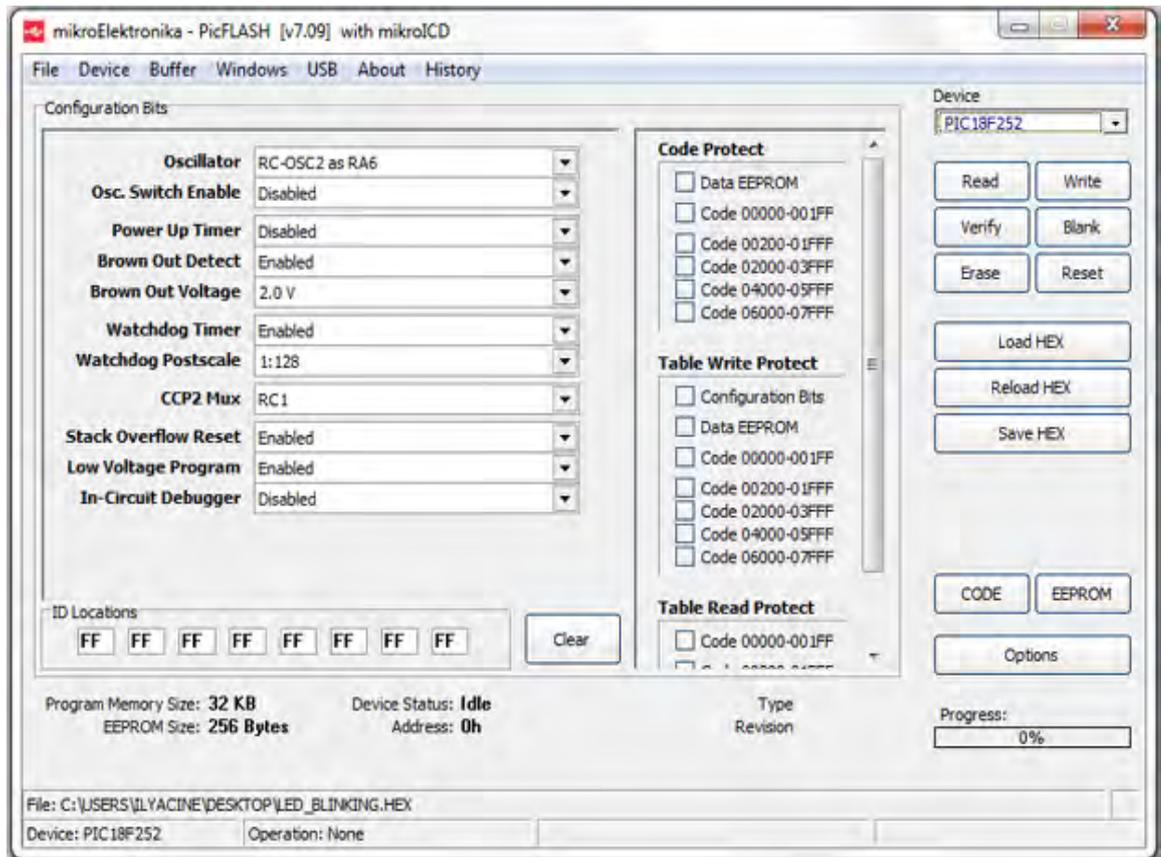


Figure 3.14: Fenêtre de PICFLASH

III.1.4 La carte de développement EASYPIC5

La carte EASYPIC5 un outil de développement pour microcontrôleurs, elle dispose d'un programmeur à connexion USB intégré qui permet de télécharger les programmes développés en n'importe quel type de compilateur pour PIC dans un microcontrôleur vierge à 8, 14, 18, 20, 28 ou 40 broches placé sur un des supports prévus à cet effet.

La carte dispose également d'un étage débogueur ICD intégré qui permet lors du développement des applications à l'aide des compilateurs mikroBASIC, mikroC ou mikroPASCAL de vérifier sur l'écran du PC les valeurs des variables, les valeurs des registres spéciaux (SFR) ou encore l'état de la mémoire EEPROM pendant l'exécution du programme afin de bénéficier d'un outil de mise au point extrêmement puissant et performant [19].

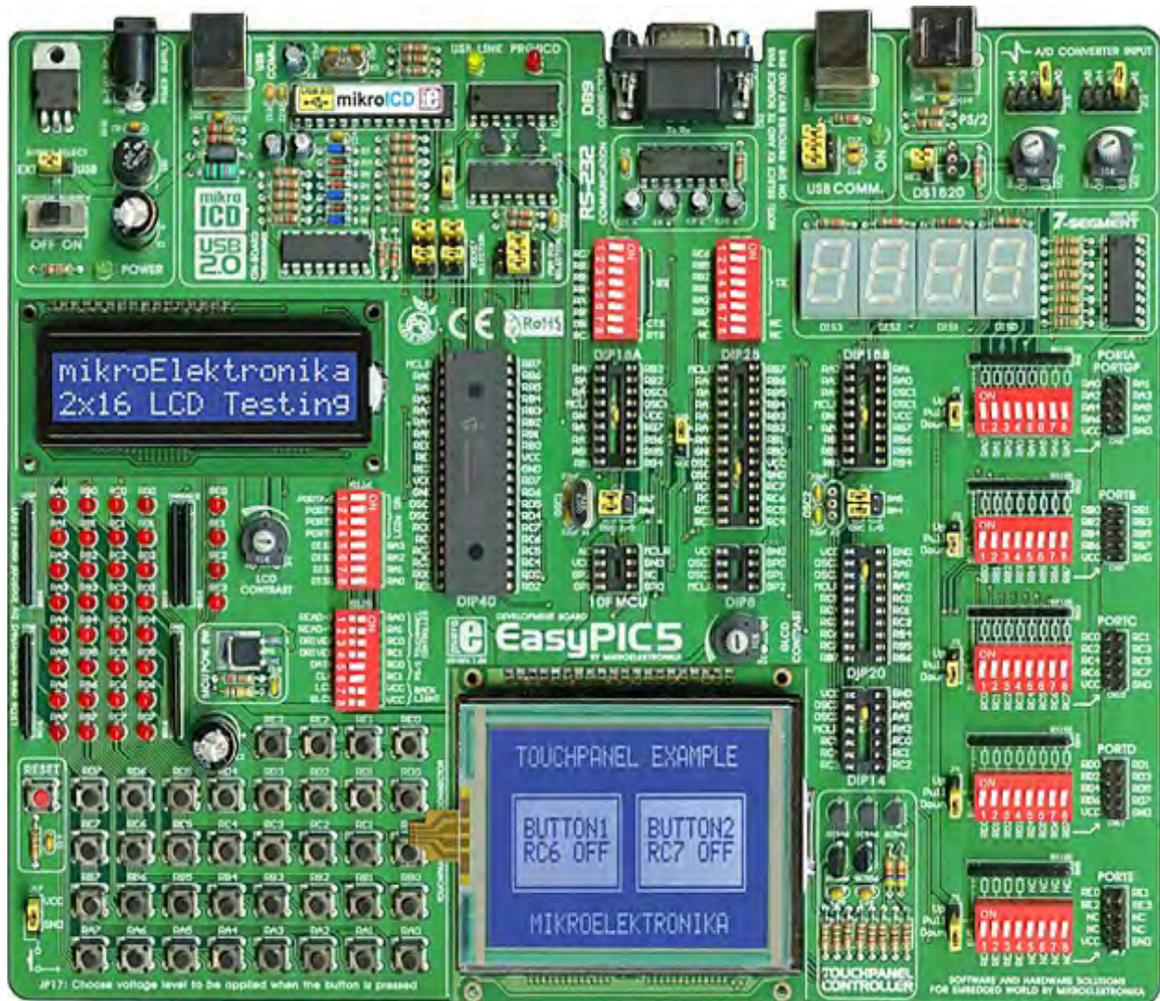


Figure 3.15: La carte de développement EASYPIC5

III.2 Le fonctionnement du programme

Le pic utilisé contient 3 ports, le port A, B, et C. Nous avons réservé le port A pour l'afficheur LCD, donc ce port est une sortie, de même pour le port B qui est une sortie permettant de contrôler les colonnes de nos matrices à LED. Le port C est une entrée qui reçoit l'information (le message à afficher) d'un ordinateur via le protocole RS232, ou bien directement d'un clavier via la liaison PS/2.

III.2.1 Les initialisations

On doit initialiser les registres TRIS [Annexe 1] de chaque port pour les faire fonctionner soit en entrée soit en sortie :

```
TRISA=0X00 ; //portA en sortie.
TRISB=0X00 ; //portB en sortie.
TRISA=0XFF ; //portc en entrée.
```

Il faut aussi noter que le port A présente des entrées analogiques dont on n'a pas besoin, donc il faut initialiser le portA pour l'utiliser comme entrée numérique, et cela est fait en mettant le registre ADCON1 =00001111 [Annexe 1]

Pour des raison du démarrage on met le portB égal à 0 ; et ça nous permettra d'éteindre toutes les LED de l'afficheur au démarrage (message vide).

```
PORTB=0X00 ;
```

III.2.2 Les déclarations

Après les initialisations, il faut déclarer les variables ainsi que les constantes utilisées dans le programme.

III.2.2.1 Les constantes

Notre programme contient un nombre important de variables, qui vont prendre des différentes valeurs lors de l'exécution du programme. On trouve seulement 3 constantes : deux messages d'aide, et une bibliothèque sous forme d'une matrice contient les différents codes attribués au portB du microcontrôleur pour contrôler les colonnes d'une matrice d'affichage à la fois, il faut noter que les colonnes de chaque matrice sont des cathodes, donc pour allumer une LED de cette matrice il faut activer la ligne dont laquelle se trouve l'anode par un « un » logique (5 V) et attribuer un « zéro » logique à la cathode de cette LED particulière.

Attribuer un « zéro logique » à la colonne revient à donner une valeur de (0 V) à cette dernière, mais vu que nous utilisons un driver ULN2804 qui est un inverseur, notre code doit attribuer un « un logique » à la colonne pour allumer une LED.

Notre constante est donc une matrice de 8 colonnes et un nombre de lignes égal à la valeur entrée lors de sa déclaration. On a déclaré tout l'alphabet français en majuscule et en minuscule, les chiffres de 0 à 9 ainsi que les caractères spéciaux ce qui donne une matrice de 95 lignes et 8 colonnes.

```
unsigned short Alphabets[][8]={
  { 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000 },      // code pour espace
  { 0b00010001, 0b00010001, 0b00010001, 0b00011111, 0b00010001, 0b00010001,
  0b00010001, 0b00001110 },      //code pour A
  { 0b00011111, 0b00010000, 0b00010000, 0b00001000, 0b00000100, 0b00000010,
  0b00000001, 0b00011111 },      // code pour Z
```

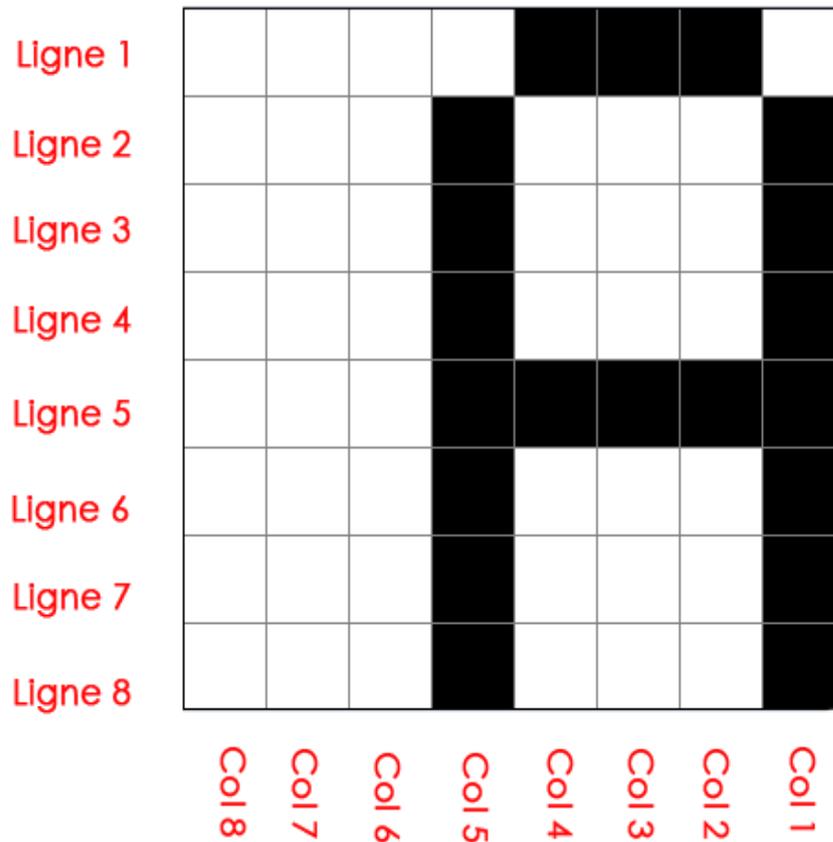


Figure 3.16: Code pour le caractère A

III.2.1.3 Les Variables

Notre programme contient plusieurs variables choisies arbitrairement, On trouve des variables « indice » utilisées dans les différentes boucles du programme, ainsi que des variables de lecture, et aussi des variables d’affichage.

Voici quelques exemples de variables utilisées :

```
int j,k,count,ent,l; //on utilise ces variables dans les différentes boucles du
                    programme.
char message[20]; // c'est une variable de lecture, elle enregistre le message entré qui
                  sera affiché plus tard.
int x1,x2,x3,x4 // variables d'affichages, chaque Xi sert à afficher un caractère
                sur une matrice d'affichage.
```

III.2.3 Les fonctions

Pour faciliter notre programme on crée une fonction pour chaque bloc de la carte. Principalement on a les fonctions suivantes :

```
Main ()
HC595 ()
Ps2 ()
RS232 ()
lcd ()
```

❖ Main () :

C'est la fonction principale de chaque programme écrit en langage C ; le PIC va d'abord exécuter cette fonction puis si elle pointe vers d'autres fonctions, le pic va les exécuter à leur tour.

Essentiellement, notre fonction main() contient des déclarations de chaque port du microcontrôleur (voir paragraphe III.2.2), des initialisations, et les différentes fonctions à exécuter l'une après l'autre.

❖ HC595 () :

C'est la fonction du balayage et du contrôle du registre 74HC595, on peut dire que c'est dans cette fonction que la partie essentielle du programme s'exécute.

Notre fonction HC595() fait un balayage suivant le nombre de matrices présentes dans la carte d'affichage, on active une ligne d'une matrice à la fois par « un logique », puis on décale ce « un logique » pour activer les lignes suivantes, par l'envoi d'une série de « zéro logique ».

Il faut noter qu'on active une ligne d'une matrice à la fois, si on arrive à la dernière ligne d'une matrice, on saute à la première ligne de la deuxième matrice et ainsi de suite .

Au fur et à mesure qu'on active une ligne d'une matrice on envoie le code approprié des différentes colonnes pour dessiner le caractère voulu.

Cette méthode utilise un phénomène bien connu qui est une caractéristique de l'œil humain : la « persistance rétinienne » (voir paragraphe II.2.1), pour chaque matrice d'affichage il faut rafraichir le dessin au moins 24 fois chaque seconde, i.e utiliser 24 images/sec.

Ayant choisi une fréquence de rafraichissement de 60 images/sec; pour une matrice de 8X8 on trouve qu'il faut rafraichir le dessin chaque $1/60 = 16$ ms, et cela est pour une matrice seule.

Donc chaque ligne est activée pendant $(1/8)*16 = 2$ ms .

Pour un prototype de 4 matrices il faut diviser ce délai par quatre, donc on va activer chaque ligne pendant $2/4 = 0.5$ ms.

Pour chaque ligne activée il faut envoyer le code colonne approprié pour dessiner le caractère voulu, à titre d'exemple pour afficher la lettre « A » on procède comme suit :

On active la 1ère ligne (1 logique) et on envoie le code colonne de la première ligne (00001110) :

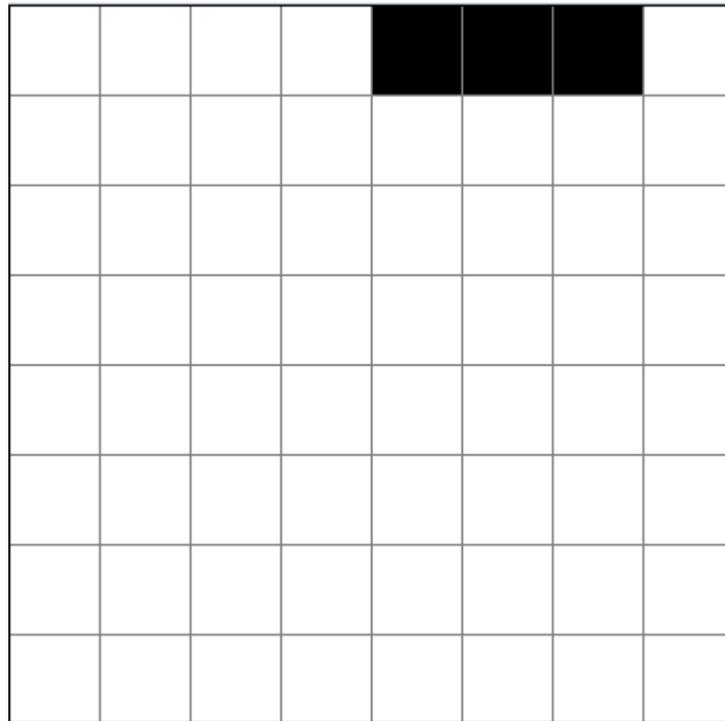


Figure 3.17.(a) : Formation de la lettre 'A'

On envoie un « zéro logique » au registre à décalage ainsi que 2 signaux d'horloge pour commencer le décalage du premier « un » envoyé précédemment, en même temps en envoi le code colonne de la 2^{ème} ligne (qui est activée maintenant) pour afficher la deuxième ligne constituant la lettre A : (00010001)

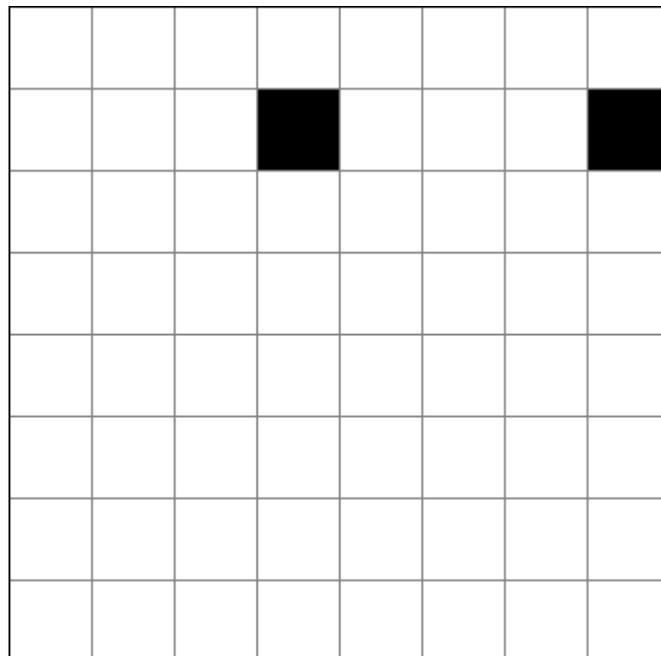


Figure 3.17.(b) : Formation de la lettre 'A'

On continue à activer les lignes l'une après l'autre tout en envoyant le code colonne approprié.

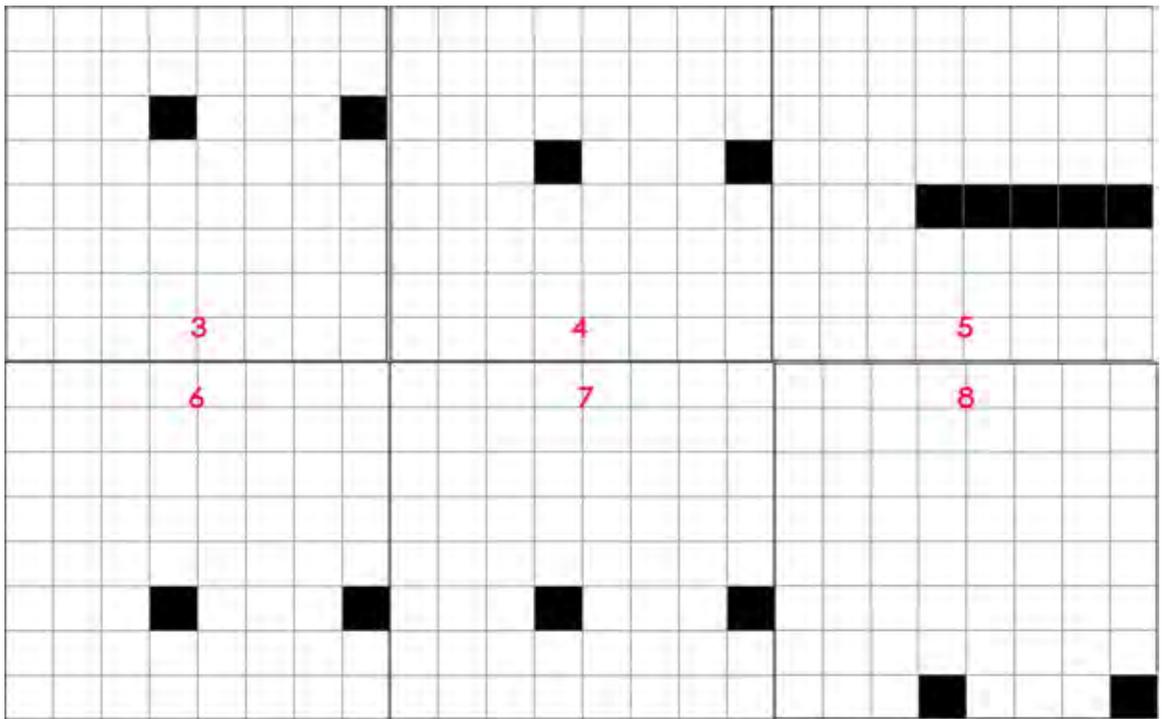


Figure 3.17.(c) : Formation de la lettre 'A'

Répétant cela à une fréquence assez élevée, on verra un caractère stable affiché sur la matrice de la forme :

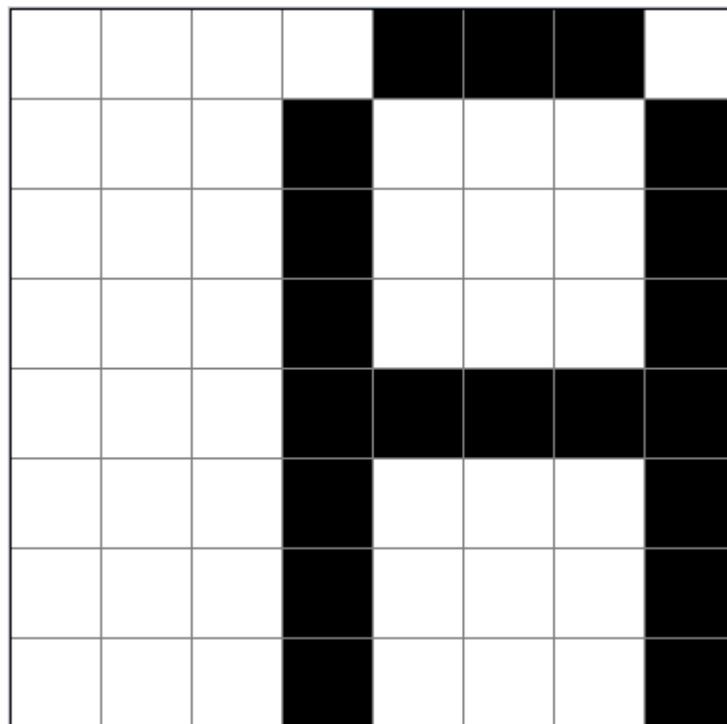


Figure 3.18 : Résultat de formation de la lettre 'A'

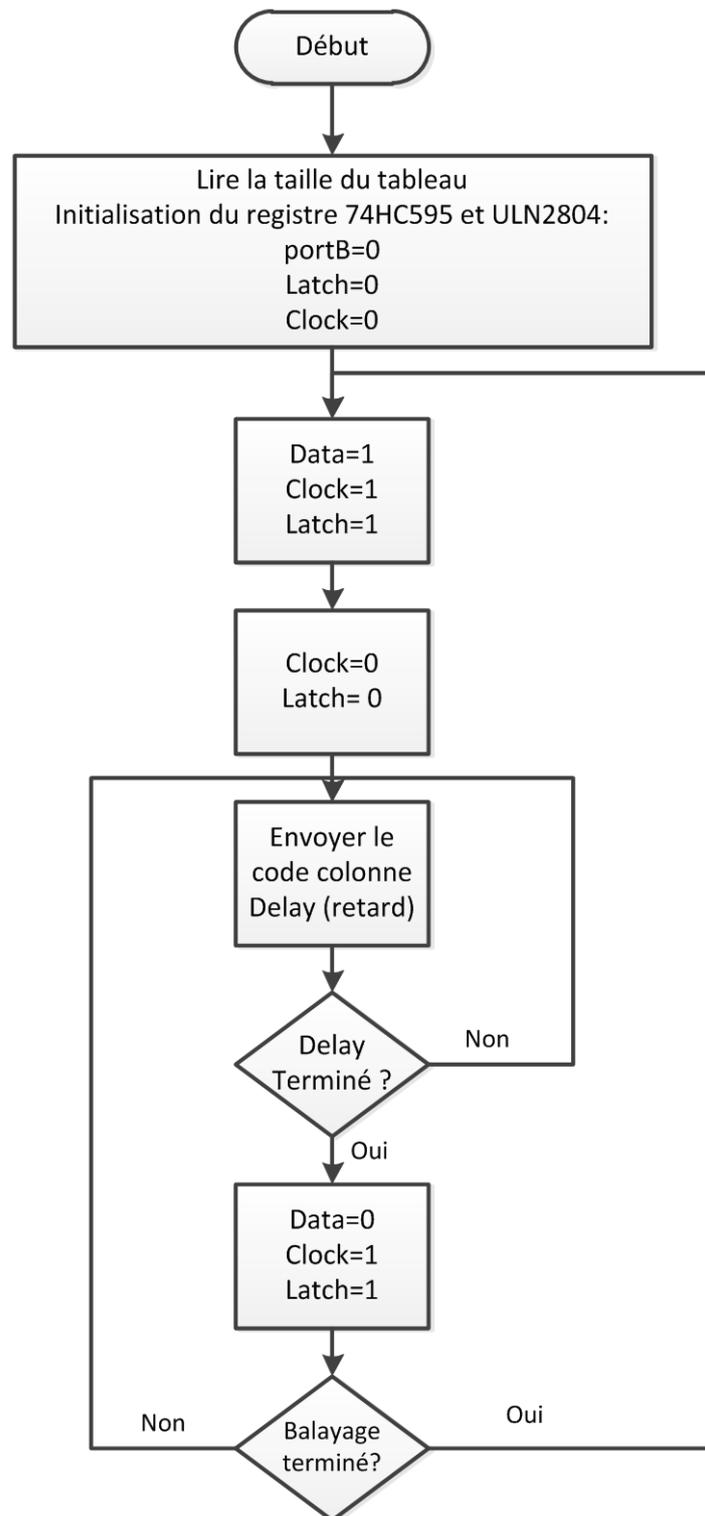


Figure 2.19 : Organigramme d’affichage d’un caractère HC595()

❖ PS2() :

C’est une fonction qui sert à lire la touche appuyée d’un clavier PS/2, puis former le message à afficher par les lettres envoyées.

Le clavier est un périphérique d’entrée et même de sortie pour l’écriture d’un message dans un ordinateur ou autre machine.

Chaque touche appuyée d’un clavier envoie un code spécial au récepteur (microcontrôleur dans notre cas), il faut traiter et traduire ce code pour former un code

compréhensible par l'unité de traitement qui sera par suite utilisé dans des tâches différentes (dans notre cas l'affichage et le sauvegarde).

Les différents codes envoyés par un clavier PS/2 si une touche est appuyée sont indiqués sur la figure (3.20).

Lorsque la touche est relâchée, on reçoit un code fixe égal à (0F) suivi du code original de la touche appuyée. Par exemple si on tape la lettre « A » on reçoit le code (1C) et si on relâche cette touche on reçoit un code de fin de balayage qui est égal à (0F 1C).

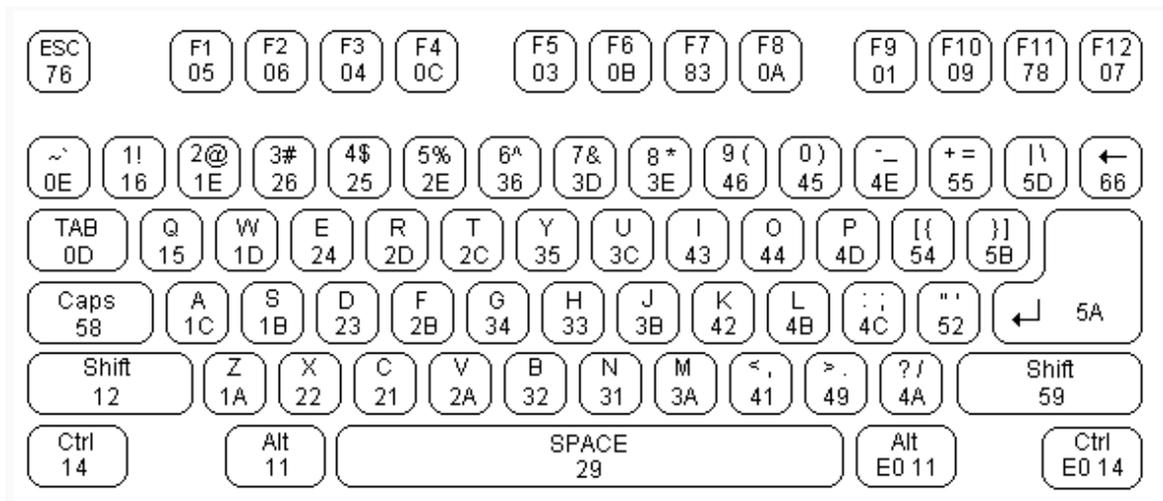


Figure 3.20.(a) : Codes du balayage d'un clavier PS/2

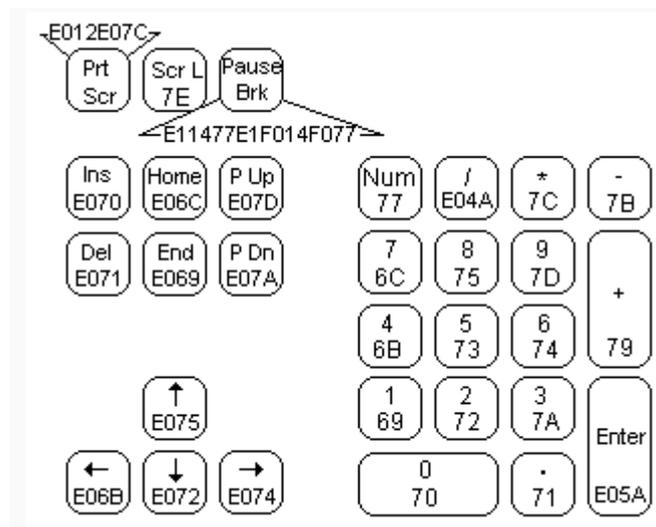


Figure 3.20.(b) : Codes du balayage d'un clavier PS/2

Après qu'on reçoit ce code, nous devons le transformer en code ASCII afin de faciliter l'enregistrement et l'affichage du message plus tard.

L'organigramme de la figure (3.21) explique comment on procède.

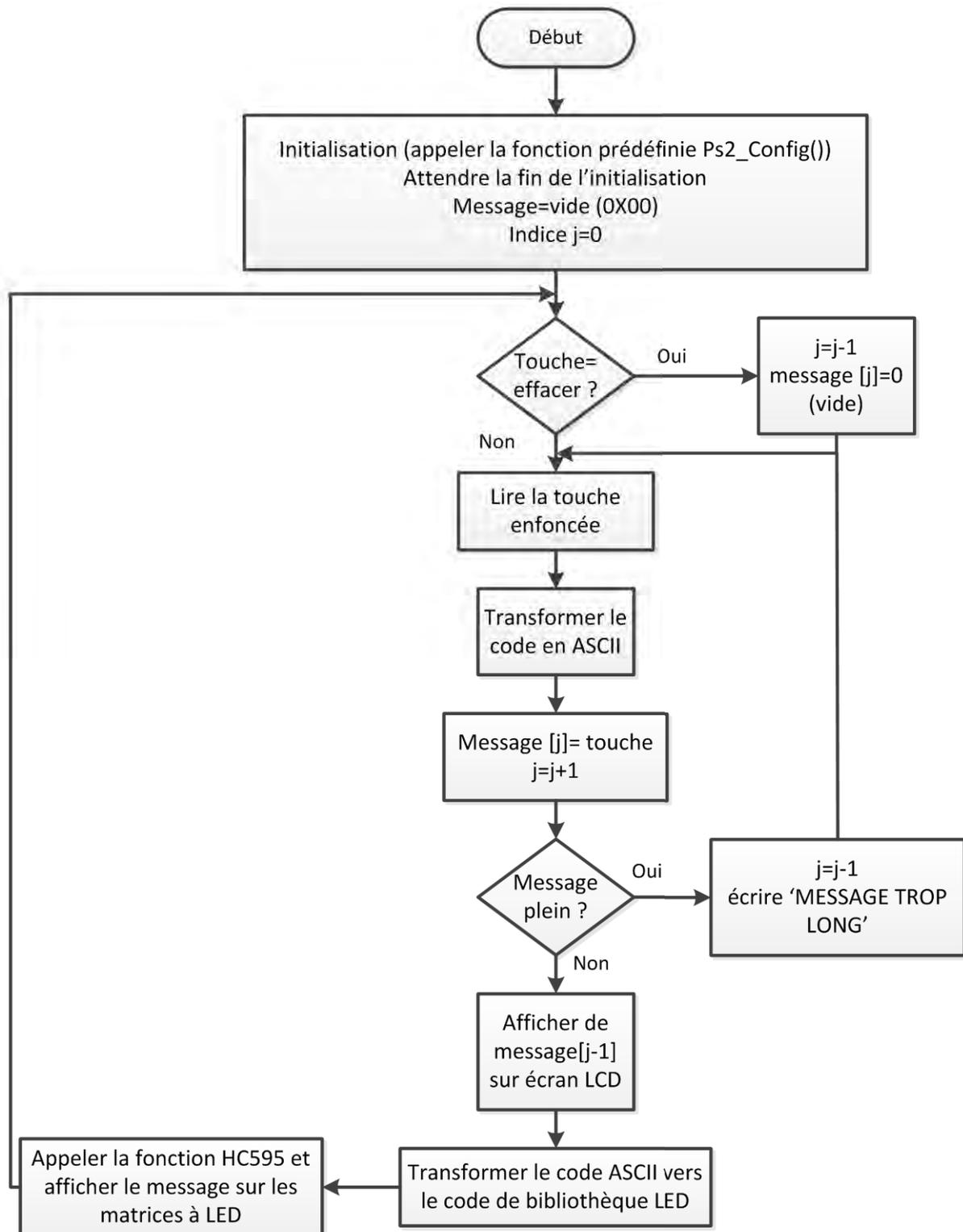


Figure 3.21: Organigramme de fonction PS2

❖ RS232() :

Cette fonction est similaire à la fonction PS2 en terme de rôle ; elle reçoit les caractères formant le message l'un après l'autre puis elle les envoie aux différentes fonctions d'affichage.

Dans cette fonction les caractères reçus sont en code ASCII, donc la simple modification consiste à transformer les données en code LED pour les afficher sur les tableau à LED .
L'organigramme est montré sur la figure (3.22)

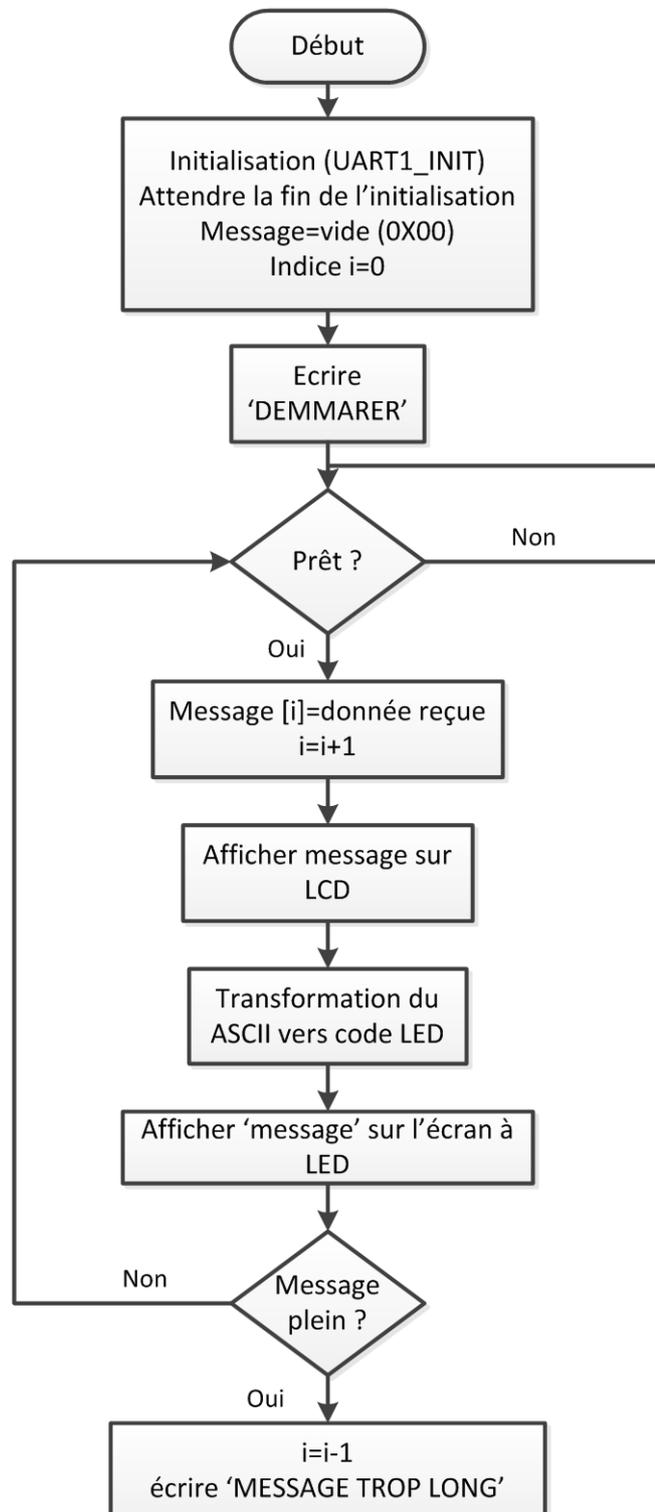


Figure 3.22: Organigramme de fonction RS232

❖ lcd () :

Cette fonction assure l'affichage du message, ainsi que les différents commentaires si nécessaire.

Tout d'abord on affiche un message 'ENTREZ VOTRE MESSAGE' sur la 1ère ligne du LCD, puis le curseur saut vers la 2ème ligne pour commencer à afficher les lettres qui forment le message.

Dans cette fonction on fait également un test chaque fois qu'on reçoit une lettre pour vérifier qu'il y a encore d'espace et on peut donc continuer à recevoir des lettres et former le message désiré.

Si le message est plein, on fait recule d'indice c'est-à-dire qu'on va recevoir la prochaine lettre au même endroit précédent qui est la dernière lettre du message, et on affiche un message d'erreur sur la 3ème ligne du LCD : 'MESSAGE TROP LONG!'.

L'organigramme de la figure (3.23) explique le fonctionnement de cette fonction :

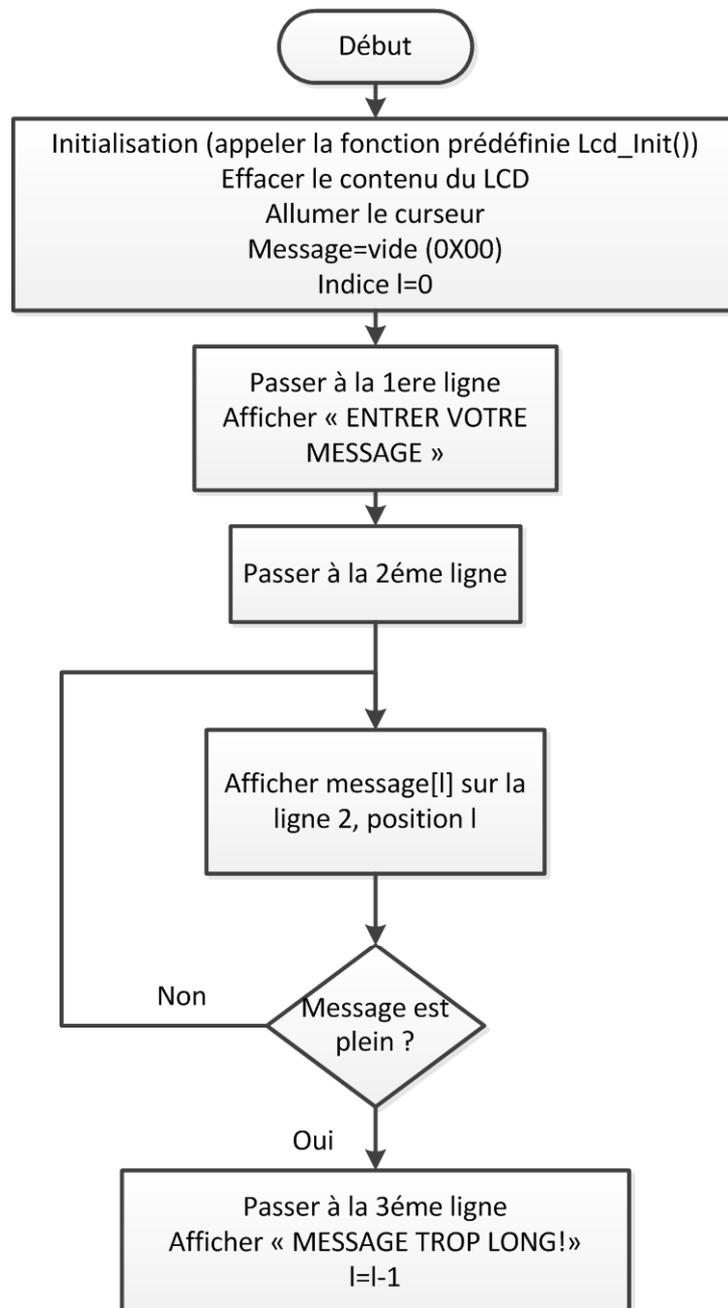


Figure 3.23: Organigramme de fonction Lcd

IV. Test et validation

Afin de valider le prototype, une série de tests a été exécutée pour s'assurer le bon fonctionnement du système. On teste les composant pour vérifier qu'ils fonctionnent bien, ainsi que chaque fonction du programme afin de la sauvegarder et l'utiliser dans le programme final.

IV.1 Allumer les matrices

On trouve rarement un circuit imprimé défectueux, cependant une matrice à LED n'est qu'une matrice de 64 LED routées sous forme matricielle, il se peut qu'une ou plusieurs LED ne fonctionnent pas.

Un simple programme est écrit pour tester chaque matrice séparément et allumer une LED d'une ligne à la fois puis passer à la prochaine ligne et ainsi de suite.

Sur 16 Matrices on a trouvé 2 matrices défectueuses.

IV.2 Commander une matrice par le 74HC595

Nos matrices sont commandées directement par des registres à décalage du type 74HC595, le registre active une ligne de la matrice à la fois, puis active la prochaine ligne après certain temps et ainsi de suite...

On branche toutes les colonnes à la masse (0 V) et on commande le registre d'une façons à faire un décalage chaque 100ms (ce qui est raisonnable et peut être observé par l'œil humain).

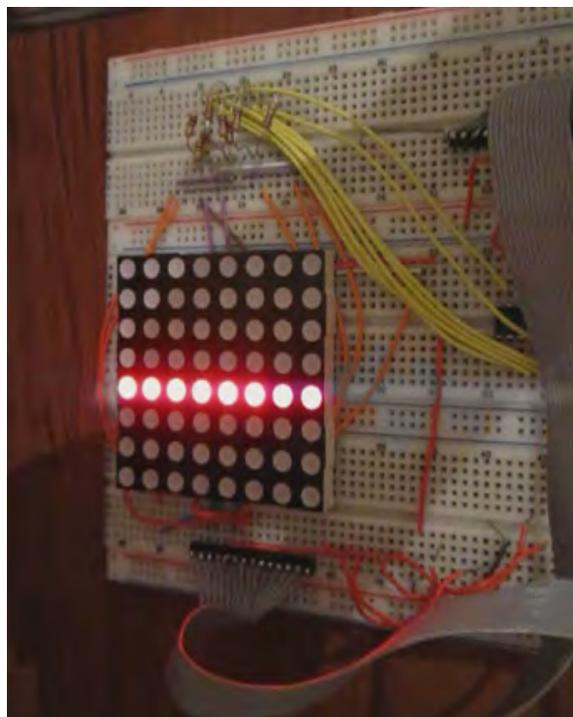


Figure 3.24: Balayage par ligne

IV.3 Afficher l'alphabet sur une matrice

Après validation du test précédent, on passe à une tâche plus compliquée qui est l'affichage d'une lettre sur la matrice puis l'affichage de tout l'alphabet sur la même matrice après la création d'une bibliothèque « ALPHABETS » qui donne la bonne commande aux colonnes pour l'affichage de chaque lettre.

Pour exécuter ce test on branche les 8 colonnes de matrice avec les 8 pins du microcontrôleur (port B), on essaye d'afficher un caractère, en activant une ligne tout en donnant la bonne commande au port B et donc au colonnes de matrice, ensuite on passe à la ligne suivante tout en donnant une nouvelle commande au portB et ainsi de suite ; pour voir que les différentes commandes sont bien synchronisées et correctement exécutées on saute d'une ligne à l'autre avec une période raisonnablement observable par l'œil humain (disons 100 ms) ; après validation du bon fonctionnement on change la période à 2 ms ; arrivant là l'œil humain ne peut plus voir la ligne activée, mais plutôt un caractère complet sur la matrice.

Pour vérifier que notre bibliothèque 'ALPHABETS' est correcte et qu'aucune LED ne reçoit la mauvaise commande on crée une boucle pour afficher tous les 95 éléments de cette bibliothèque, et si un caractère n'est pas affiché correctement, il faut changer l'élément de bibliothèque correspondant ; l'affichage d'une lettre doit durer au moins 100 ms pour que nous puissions détecter les mauvaises commandes, cependant il est préférable d'enregistrer une vidéo pour l'analyser confortablement.

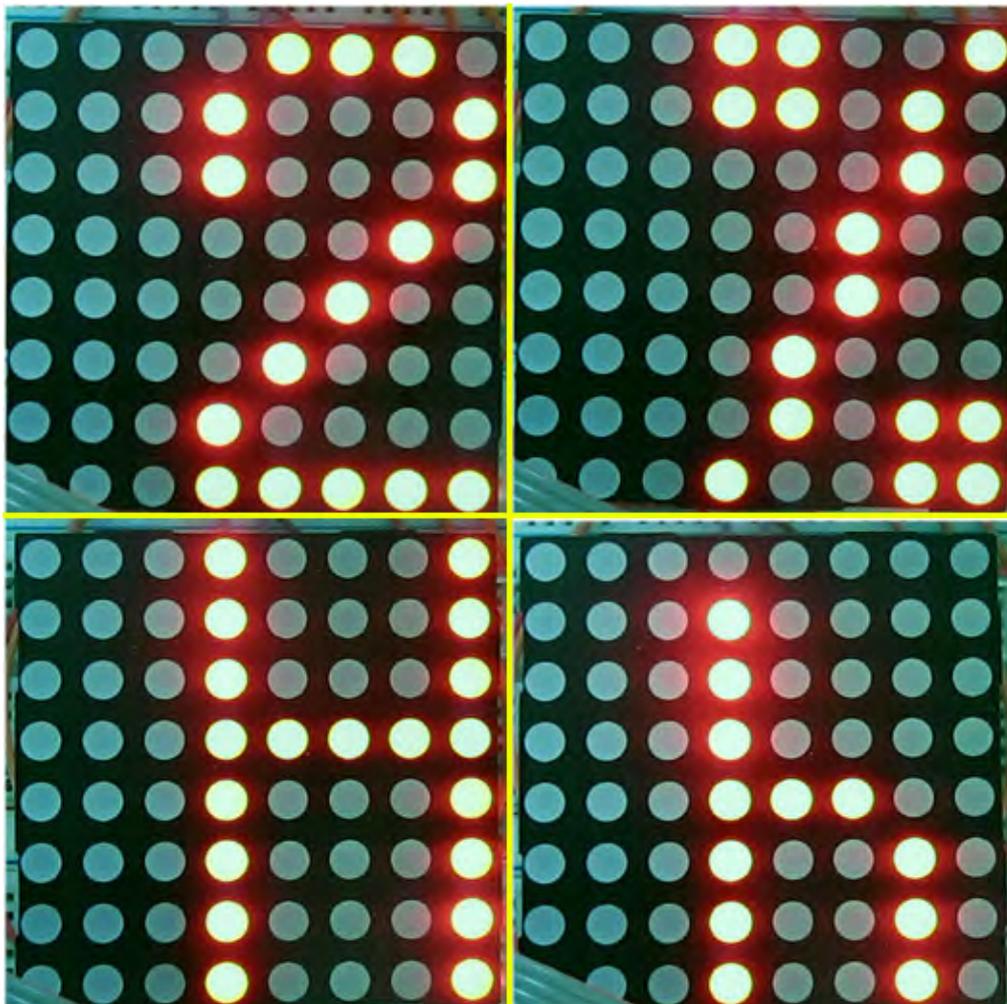


Figure 3.25: Affichage d'alphabets

IV.4 Tester l'écran LCD

Sur la carte EasyPic5 un LCD peut être placé sur le portB du microcontrôleur, afin de tester cet écran on définit des messages fixes dans la mémoire du PIC et on les affiche sur le LCD par l'algorithme de la figure (3.23)

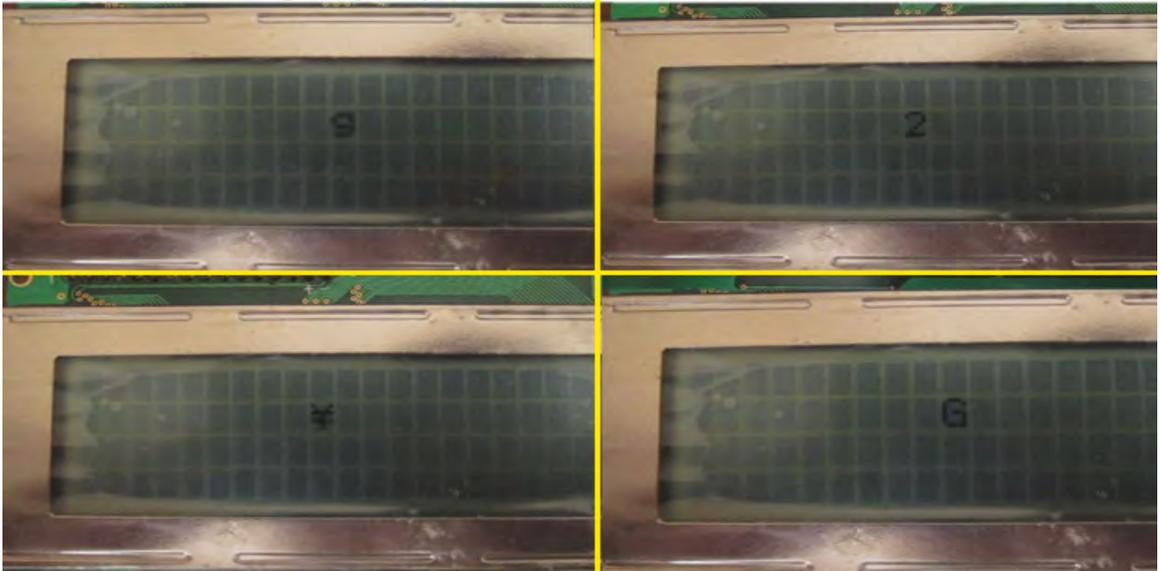


Figure 3.26 : Test d'écran LCD

IV.5 Tester les connexions PS/2 et RS232

Avant d'afficher les messages directement sur les matrices, on doit vérifier qu'on reçoit les bonnes valeurs via les différentes connexions de notre prototype ; afficher les messages reçus sur un écran LCD est aussi une bonne solution pour s'assurer qu'on envoie le bon code reçu par une liaison série RS232 ou par un clavier PS/2 (voir paragraphe III.2.2.3).

Après validation, les différentes lettres reçues seront affichées sur la matrice testée auparavant.

IV.6 Balayer tout le panneau

Après validation des différents blocs de notre prototype, il ne nous reste que la validation de notre carte d'affichage (panneau d'affichage) ; notre prototype contient 4 matrices d'affichage voir $4 \times 64 = 256$ LED ; toutes les colonnes de ces matrices sont commandées en même temps par seulement le portB de notre PIC, cela veut dire que si on active en même temps une ligne de chaque matrice tout en envoyant le code colonne voulu, le panneau va afficher la même lettre répétée 4 fois.

Il faut donc balayer une matrice à la fois (chaque ligne) puis passer vers la prochaine matrice jusqu'à la fin du panneau puis répéter le balayage de nouveau, et cela en envoyant le code colonne approprié pour chaque ligne de chaque matrice.

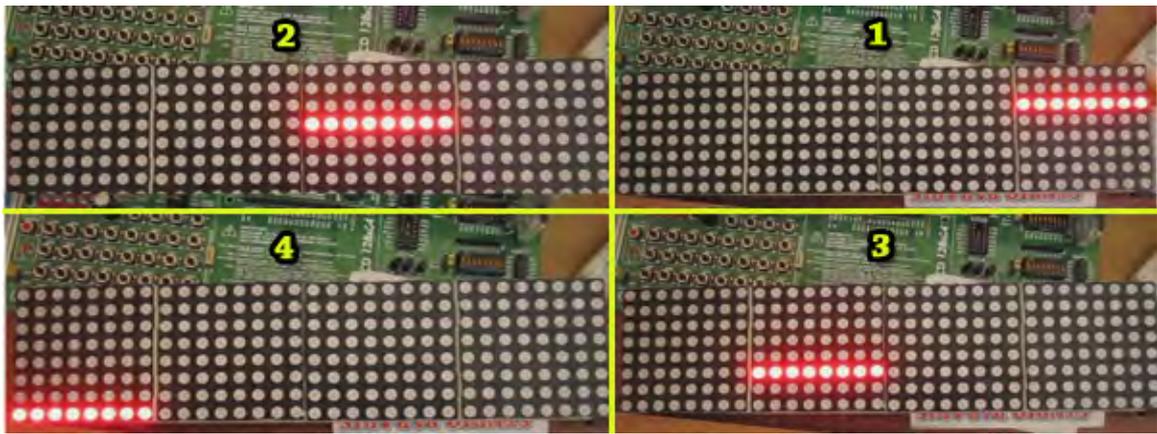


Figure 3.27: Balayage du panneau complet

Avec l'utilisation de 4 matrices et le balayage d'une ligne de chaque matrice à la fois, on doit augmenter la fréquence du balayage pour tromper l'œil humain qui voit un message stable entrain d'être affiché, mais en réalité on n'affiche qu'une ligne d'une matrice à la fois, et donc l'utilisation d'une période de 2ms pour une matrice se traduit par une nouvelle période de $2/4 = 0.5$ ms voir $500\mu\text{s}$ pour 4 matrices, et en règle générale :

$$\text{Période} = \frac{2}{N}$$

Tel que :

N est le nombre de matrices utilisées.

On complète le programme par l'ajout des différents commentaires et messages d'aide, ainsi que des conditions sur l'utilisation concernant la taille du message et les différents accessoires du prototype.



Figure 3.28: Test global

V. Conclusion :

Un panneau routier à messages variables peut être réalisé suivant plusieurs méthodes ; vu les dimensions d'un tel panneau, la priorité dans notre travail était de développer une méthode de réalisation la moins coûteuse possible tout en respectant notre cahier de charge concernant sa capacité ainsi que les différentes exigences; il fallait exécuter plusieurs tests afin de confirmer et valider nos choix. La partie programmation représente le plus gros de ce travail, il fallait développer un programme convenable avec le prototype conçu ; des explications détaillées concernant les différentes fonctions du programme ainsi que les techniques utilisées ont été présentée dans ce chapitre. En effet l'utilisation du PIC est essentiel, toutefois le programme devrait être modifié si le prototype est modifié, nous avons équipé ce dernier avec des différents accessoires afin d'aider l'utilisateur à l'utiliser.

Conclusion Générale

Ce travail consistait en la conception et la réalisation d'un prototype pour un panneau routier à messages variables permettant l'affichage de différentes informations à l'usage des conducteurs; ce panneau doit répondre aux différentes exigences concernant sa taille, sa police d'écriture, et son emplacement. Le prototype que nous avons proposé est basé sur l'utilisation d'un microcontrôleur PIC de la famille 18F (18F252), ainsi que d'autres pièces assurant le bon fonctionnement de nos différents modules.

On a choisi ce PIC parce qu'il présente une solution optimale pour cette tâche, le résultat final nous confortant dans notre position.

Un programme a été créé pour la gestion de tout le système ; celui-ci est constitué de plusieurs fonctions, chacune effectuant une tâche bien définie. Des tests nous ont permis de valider le fonctionnement du programme.

Le prototype final peut être équipé d'un émetteur/récepteur sans fil pour assurer une gestion à distance et envoyer les messages appropriés en temps réel ; nous constatons que l'ajout de ce dernier apporte clairement un avantage supplémentaire. Par ailleurs, à des fins écologiques notre afficheur peut être muni d'une alimentation photovoltaïque, rendant ainsi notre dispositif entièrement autonome en terme d'alimentation.

Nous notons qu'il sera intéressant de compléter ce travail par l'utilisation de ces panneaux sous forme d'un réseau, de tel sorte que chaque panneau peut communiquer avec les autres, afin d'afficher des messages concrets, de même l'ajout d'une série de capteurs sur la route peut rendre le système entièrement autonome surtout dans ce qui concerne l'affichage d'état du trafic.

Bibliographie

- [1]. Conference of European Directors of Roads - CEDR (2003). Framework for harmonized implementation of VMS in Europe, CEDR action FIVE, Draft amended version 3.6, March 2004 (conférence européenne des directeurs des routes- 2003).
- [2]. J. Nouvier, M. Duhamel, A. Arbaiza, A. Lucas, 'The history of traffic signing in France and Europe: The role of international conventions', proceedings of 23rd World Road Congress, (2007).
- [3]. United Nations / Economic Commission for Europe, UN/ECE (1968). Convention on road signs and signals (1968), Vienna (Publication des nations unies, ISBN: 978-92-1-116973-7).
- [4]. Display of Information on Variable Message Signs (Department of Transport and Main Roads, 2009) (Décret gouvernemental –Queensland- 2009).
- [5]. Guidelines for the Location and Placement of Variable Message Signs (Roads and Traffic Authority, New South Wales, 2008), (guide de placement des PMV - Galles du Sud- 2008).
- [6]. Traffic and Road Use Management Manual - Variable Message Signs: Use and Operation (Rev August 2010) (Manuel de gestion des routes –Australie- 2010).
- [7]. Main Roads : ITS Supply and Installation Specification (Rev Oct 04), (guide d'installation de PMV, 2004).
- [8]. Harmonisation des PMV en Europe - conférence européenne des directeurs des routes (CEDR – 2009).
- [9]. <http://www.microchip.com/> ; accédé le 17/06/2012
- [10]. http://fr.wikipedia.org/wiki/Port_PS/2 ; accédé le 18/06/2012
- [11]. <http://fr.wikipedia.org/wiki/RS-232> ; accédé le 18/06/2012
- [12]. Ögren, Joakim "Serial (PC 9) Hardware Book" (2008).
- [13]. Datasheet du circuit intégré MAX232 :
<http://pdfserv.maxim-ic.com/en/ds/MAX220-MAX249.pdf>
- [14]. Datasheet du circuit intégré
74HC595 : <http://www.sparkfun.com/datasheets/IC/SN74HC595.pdf>

- [15]. [http://fr.wikipedia.org/wiki/Persistance rétinienne](http://fr.wikipedia.org/wiki/Persistance_rétinienne) ; accédé le 20/06/2012
- [16]. Brian W. Kernighan et Dennis M. Ritchie "Le langage C" Manson 1990.
- [17]. "mikroC PRO for PIC User Manual v100": Manuel d'utilisateur (2009):
http://www.mikroe.com/pdf/mikroc_pic_pro/mikroc_pic_pro_manual_v100.pdf
- [18]. "PICflash manual" Manuel d'utilisateur (2009) :
http://www.mikroe.com/pdf/picflash_manual.pdf
- [19]. Kits de développement
MikroElektronika : <http://www.mikroe.com/eng/categories/view/1/development-tools/>

Annexes:

Annexe 1: Caractéristiques du PIC 18F252

TABLE 1-2: PIC18F2X2 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	DIP	SOIC			
MCLR/VPP $\overline{\text{MCLR}}$ VPP	1	1	I I	ST ST	Master Clear (input) or high voltage ICSP programming enable pin. Master Clear (Reset) input. This pin is an active low RESET to the device. High voltage ICSP programming enable pin.
NC	—	—	—	—	These pins should be left unconnected.
OSC1/CLKI OSC1 CLKI	9	9	I I	ST CMOS	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode, CMOS otherwise. External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
OSC2/CLKO/RA6 OSC2 CLKO RA6	10	10	O O I/O	— — TTL	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate. General Purpose I/O pin.

2.0 OSCILLATOR CONFIGURATIONS

2.1 Oscillator Types

The PIC18FXX2 can be operated in eight different Oscillator modes. The user can program three configuration bits (FOSC2, FOSC1, and FOSC0) to select one of these eight modes:

1. LP Low Power Crystal
2. XT Crystal/Resonator
3. HS High Speed Crystal/Resonator
4. HS + PLL High Speed Crystal/Resonator with PLL enabled
5. RC External Resistor/Capacitor
6. RCIO External Resistor/Capacitor with I/O pin enabled
7. EC External Clock
8. ECIO External Clock with I/O pin enabled

9.0 I/O PORTS

Depending on the device selected, there are either five ports or three ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Each port has three registers for its operation. These registers are:

- TRIS register (data direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (output latch)

The data latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving.

TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Ranges Tested:			
Mode	Freq	C1	C2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF
These values are for design guidance only. See notes following this table.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

EXAMPLE 9-1: INITIALIZING PORTA

```

CLRF PORTA      ; Initialize PORTA by
                 ; clearing output
                 ; data latches
CLRF LATA       ; Alternate method
                 ; to clear output
                 ; data latches
MOVLW 0x07      ; Configure A/D
MOVWF ADCON1    ; for digital inputs
MOVLW 0xCF      ; Value used to
                 ; initialize data
                 ; direction
MOVWF TRISA     ; Set RA<3:0> as inputs
                 ; RA<5:4> as outputs

```

FIGURE 9-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS

17.0 COMPATIBLE 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has five inputs for the PIC18F2X2 devices and eight for the PIC18F4X2 devices. This module has the ADCON0 and ADCON1 register definitions that are compatible with the mid-range A/D module.

The A/D allows conversion of an analog input signal to a corresponding 10-bit digital number.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

The ADCON0 register, shown in Register 17-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 17-2, configures the functions of the port pins.

REGISTER 17-1: ADCON0 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
						bit 7	bit 0

bit 7-8 **ADCS1:ADCS0:** A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	Frc (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	Frc (clock derived from the internal A/D RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits

000 = channel 0, (AN0)
 001 = channel 1, (AN1)
 010 = channel 2, (AN2)
 011 = channel 3, (AN3)
 100 = channel 4, (AN4)
 101 = channel 5, (AN5)
 110 = channel 6, (AN6)
 111 = channel 7, (AN7)

Note: The PIC18F2X2 devices do not implement the full 8 A/D channels; the unimplemented selections are reserved. Do not select any unimplemented channel.

bit 2 **GO/DONE:** A/D Conversion Status bit

When ADON = 1:

1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)
 0 = A/D conversion not in progress

bit 1 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit

1 = A/D converter module is powered up
 0 = A/D converter module is shut-off and consumes no operating current

22.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings ^(†)

Ambient temperature under bias	-55°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to V _{SS} (except V _{DD} , $\overline{\text{MCLR}}$, and RA4)	-0.3V to (V _{DD} + 0.3V)
Voltage on V _{DD} with respect to V _{SS}	-0.3V to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to V _{SS} (Note 2)	0V to +13.25V
Voltage on RA4 with respect to V _{SS}	0V to +8.5V
Total power dissipation (Note 1)	1.0W
Maximum current out of V _{SS} pin	300 mA
Maximum current into V _{DD} pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > V _{DD})	±20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > V _{DD})	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA, PORTB, and PORTE (Note 3) (combined)	200 mA
Maximum current sourced by PORTA, PORTB, and PORTE (Note 3) (combined)	200 mA
Maximum current sunk by PORTC and PORTD (Note 3) (combined)	200 mA
Maximum current sourced by PORTC and PORTD (Note 3) (combined)	200 mA

Note 1: Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

2: Voltage spikes below V_{SS} at the $\overline{\text{MCLR}}$ /V_{PP} pin, inducing currents greater than 80 mA, may cause latchup. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{\text{MCLR}}$ /V_{PP} pin, rather than pulling this pin directly to V_{SS}.

3: PORTD and PORTE not available on the PIC18F2X2 devices.

Annexe 2 : Caractéristiques du PVC200401PTN

6. Interface pin description

Pin no.	Symbol	External connection	Function
1	V _{SS}	Power supply	Signal ground for LCM
2	V _{DD}		Power supply for logic for LCM
3	V ₀		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Annexe 3 : Caractéristiques de HS2088B**ABSOLUTE MAXIMUM RATINGS (Ta=25° C)**

PARAMETER	SYMBOL	GREEN
REVERSE VOLTAGE	V_R	5v
FORWARD CURRENT	I_F	20mA
PEAK FORWARD CURRENT	I_{PEAK}	120mA
POWER DISSIPATION	P_D	80mW
OPERATING TEMPERATURE RANGE	T_A	-30°C ~ +85°C
STORAGE TEMPERATURE RANGE	T_{STG}	-30°C ~ +85°C
TEMPERATURE OF THE SOLDERING BATH (4mm)	3sec	<260°C

ELECTRICAL-OPTICAL CHARACTERISTICS (Ta=25° C)

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
Luminous Intensity	I_V		4.5		mcd	IF=10mA
Forward Voltage	V_F		2.1	2.5	v	IF=20mA
Peak Emission Wavelength	λ_p		568		nm	IF=20mA
Spectral Line Half-Width	$\Delta \lambda$		30		nm	IF=20mA
Reverse Current	I_R			50	uA	VR=5v

Annexe 4 : Caractéristiques du 74HC595

2. Features and benefits

- 8-bit serial input
- 8-bit serial or parallel output
- Storage register with 3-state outputs
- Shift register with direct clear
- 100 MHz (typical) shift out frequency
- ESD protection:
 - ◆ HBM JESD22-A114F exceeds 2000 V
 - ◆ MM JESD22-A115-A exceeds 200 V
- Multiple package options
- Specified from -40 °C to $+85\text{ °C}$ and from -40 °C to $+125\text{ °C}$

Table 4. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134). Voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions	Min	Max	Unit	
V_{CC}	supply voltage		-0.5	+7	V	
I_{IK}	input clamping current	$V_I < -0.5\text{ V}$ or $V_I > V_{CC} + 0.5\text{ V}$	-	± 20	mA	
I_{OK}	output clamping current	$V_O < -0.5\text{ V}$ or $V_O > V_{CC} + 0.5\text{ V}$	-	± 20	mA	
I_O	output current	$V_O = -0.5\text{ V}$ to $(V_{CC} + 0.5\text{ V})$				
		pin Q7S	-	± 25	mA	
		pins Qn	-	± 35	mA	
I_{CC}	supply current		-	70	mA	
I_{GND}	ground current		-70	-	mA	
T_{stg}	storage temperature		-65	+150	$^{\circ}\text{C}$	
P_{tot}	total power dissipation	DIP16 package	[1]	-	750	mW
		SO16 package	[2]	-	500	mW
		SSOP16 package	[3]	-	500	mW
		TSSOP16 package	[3]	-	500	mW
		DHVQFN16 package	[4]	-	500	mW

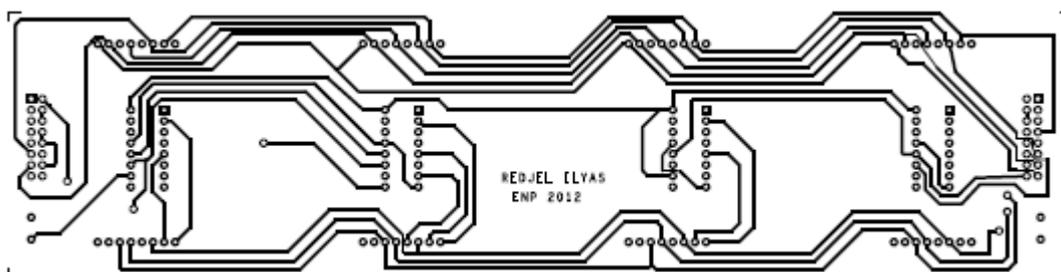
[1] For DIP16 package: P_{tot} derates linearly with 12 mW/K above 70 $^{\circ}\text{C}$.

[2] For SO16 package: P_{tot} derates linearly with 8 mW/K above 70 $^{\circ}\text{C}$.

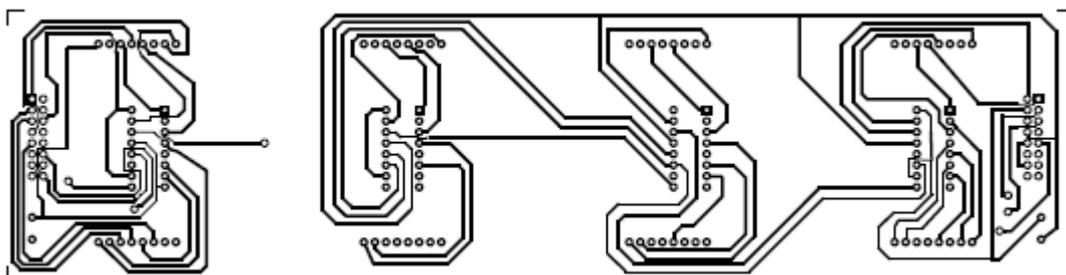
[3] For SSOP16 and TSSOP16 packages: P_{tot} derates linearly with 5.5 mW/K above 60 $^{\circ}\text{C}$.

[4] For DHVQFN16 package: P_{tot} derates linearly with 4.5 mW/K above 60 $^{\circ}\text{C}$.

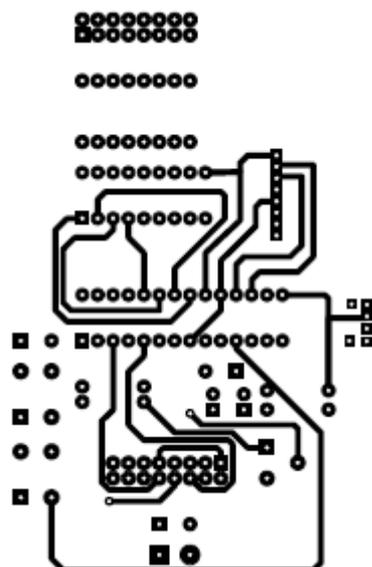
Annexe 5: Typons des différentes cartes



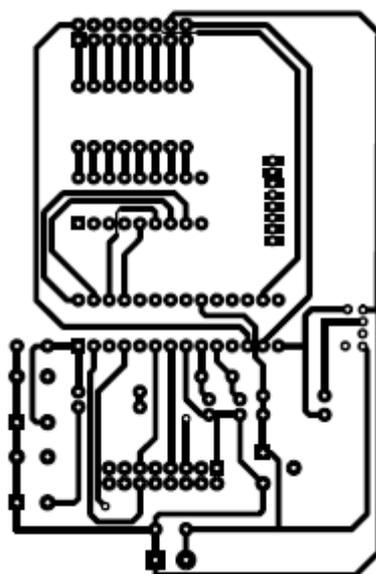
Carte d'affichage (Couche Top - échèle 1:2)



Carte d'affichage (Couche Bottom -échelle 1:2)



Carte de contrôle (Couche Top -échelle 1:1)



Carte de contrôle (Couche Bottom -échelle 1:1)