UNIVERSITÉ
de Picardie
Jules Verne

UNIVERSITE PICARDIE JULES VERNES
INSTITUT UNIVERSITAIRE POLYTECHNIQUE
GENIE ELECTRIQUE ET INFORMATIQUE
INDUSTRIELLE

Projet

Télécommande multicanaux HF

TUTEUR DU PROJET: Mr Frédéric COLLET
ENTREPRENEUR: Mr Jean Marc VASSEUR

AUTEUR : DEBUT GORGAS, Adrien

4 Avril 2008

IUP GEII

33 Rue St Leu 03-22-82-78-23
80039 AMIENS Cedex 1 03-22-82-78-21

Remerciements

Je veux remercier tout d'abord à me parents, car ce sont les acteurs les plus importants de mon voyage Erasmus, sans oublier que le programme Erasmus en fait partie active. Et bien sûr à mon université d'origine « Universidad Carlos III de Madrid » et mon université d'accueil, l'Université Picardie Jules Vernes et plus précisément à L'IUP GEII.

D'autre part je veux remercie mon tuteur de projet Mr Frédéric COLLET et l'entrepreneur Mr Jean Marc VASSEUR, qui m'a permis de m'engagé dans ce projet si motivant.

Finalement nous ne voulons pas oublier nos amis Erasmus, qui ont sût respecter et appuyer notre travail.

Et dernier lieu à ma copine María à eu toute la patience du monde et ma aidé pendant tout le trajet.

Indice

I) **INTRODUCTION** 5

II) **PRESENTATION DU PROJET** 6

A) **DESCRIPTION DU PROJET** 6
B) **POSITION ENVISAGEE DANS LE PROJET** 6
C) **PRINCIPAUX ASPECTS DU PROJET** 6

III) **CAHIER DES CHARGES** 7

A) **ASPECTS PRINCIPAUX** 7
B) **ASPECTS TECHNIQUES** 8
1) L'émetteur 8
2) Le récepteur 8

IV) **L'EMETTEUR** 9

A) **ÉLECTION DES COMPOSANTS PRINCIPAUX** 9
1) Élection du micro contrôleur 9
2) Choix du codeur 9
3) Choix de l'émetteur radiofréquences. 10
4) Clavier et LCD 10
B) **CONCEPTION DU CIRCUIT :** 11
1) Schéma fonctionnel 11
2) Connexion du Codeur et émetteur FM 11
3) Clavier et LCD 12
4) Conception finale 12
C) **LE LOGICIEL** 13
1) Programmation du PIC 13
2) Chargement du logiciel 13
3) Structure du programme 13

V) **LE RECEPTEUR** 15

1) Source d'alimentation 15
2) Réception 15

VI) **CONCLUSION** 17

VII) **BILAN PERSONNEL** 18

VIII) **ANNEXES** 19

I) Introduction

Ce projet pluridisciplinaire à pour but de développer les compétences générales des étudiants, afin de motiver y d'enseigner les méthodes de travail qui ferons partie de sont futur domaine de travail. Ainsi l'élève pourra ce familiariser dans tous les domaines faisant parties du projet. Depuis le début, avec la conception du cahier de charges avec l'entrepreneur et la décision des buts a atteindre, jusqu'à la conception du prototype et modifications pour en aboutir au produit commerçable.

Notre projet intitulé « Télécommande Multicanaux HF », rempli toutes les objectifs d'un projet industriel. Pour cela nous réalisons notre projet à partir des étapes suivantes :

- Présentation du projet
- Cahier des charges
- L'émetteur
- Le récepteur
- Conclusion
- Annexes

Nous allons à partir d'ici détailler toutes les démarches nécessaires que nous avons suivit, jusqu'à l'aboutissement de projet, depuis la théorie, jusqu'à la pratique avec les prototypes et les essais.

II) Présentation du projet

A) Description du projet

Ce projet est naît à partir du besoin des forains de trouver une solution face aux auto-tamponneuses, l'handicap qui survient est lorsque une voiture à un problème, le forain est obligé à stopper toutes les voitures en coupant le courant, et donc à redonner des fiches car le cycle de jeux n'était pas encore finit.

Il faudra donc contrôler les auto-tamponneuses individuellement depuis la cabine de contrôle. Nous envisageons comme contrôle la capacité d'allumer ou éteindre une auto-tamponneuse par une simple commande. Ou bien plusieurs voitures, selon les besoins du forain.

B) Position envisagée dans le projet

Puisque nous sommes dans milieu sans contact direct avec la voiture il faudra envisager un contrôle par radiofréquence par codage pour pouvoir sélectionner chaque voiture individuellement.

Notre choix c'est donc porté sur ce sujet car il envisage tout les aspects d'un projet réel, depuis le plan théorique, la conception y la réalisation finale du même. D'autre part il envisage aussi la partie des micro-controleurs et antennes de télécommunications, aspects très motivants pour notre formation professionnelle et personnelle.

C) Principaux aspects du projet

Notre projet se concentrera alors à la réalisation de deux prototypes, l'émetteur qui sera situé dans la cabine de commande de la foire, puis un récepteur, qui lui, sera situé dans chaque auto-tamponneuse.

Il faudra tenir en compte alors des alimentations, tels que l'alimentation dans la cabine et l'alimentation dans le véhicule. Il faut comprendre que nos sommes dans un milieu avec beaucoup de contraintes, la source d'alimentation est presque toujours un générateur qui n'a pas de tension stable, d'autre part au niveau du récepteur les contraintes sont plus fortes car il y a des variations plus importantes dues au contact des balais de la voiture, et aux coupures de courant de chaque cycle de jeux.

III) Cahier des charges

A) Aspects principaux

Nous nous situons dans un milieu hostile pour les radiofréquences, il s'agit d'une cage de faraday car le sol est métallique et le toit est une grille métallique, qui a la fois sont chargées d'un courant continu pour alimenter les auto-tamponneuses, négatif au sol et positif sous le toit. Il faudra tenir en compte la distance depuis la voiture la plus lointaine, maximum 50 [m], sans oublier un coefficient de sécurité assez ample. D'autre part le contrôle du récepteur sur la voiture est particulièrement difficile car le moteur électrique à une puissance qui varie entre 500 et 700 [W], ce qui suppose en tenant en compte les variations de tensions, c'est-à-dire de 80 à 100 [V]. Un courant de variable d'entre 5 et 8 [A], toutes ces variations sont à étudier lorsque nous réalisons notre récepteur, car il doit contrôler de fortes variations de courant et tensions tout en restant stable. D'autre part le récepteur doit être très grand car la place dans le véhicule n'est pas importante, ne pas oublier les poussières métalliques que produisent les balais du véhicule, et donc qui peuvent conduire à des court circuits.

Au niveau de l'émetteur, nous devons concevoir un système simple et pratique pour le forain. Ce qui veut dire, une plaque avec un clavier et un écran visible en telle sorte qu'il n'aboutisse pas d'erreurs. Le contrôleur de la plaque émettrice doit être suffisamment puissant pour communiquer avec tous les périphériques, tout en restant simple pour les aptitudes de l'entrepreneur et les futures modifications. Par rapport au logiciel, nous devons comprendre les demandes des forains, donc tenir en compte d'une prise de programmation pour des modifications rapides.

Il ne faut pas oublier de la question économique, par exemple pour le récepteur nous devons en avoir minimum trente, autant que de voitures, il faut donc produire notre propre système de régulation, car les contraintes font du même, sur le marché, un composant électronique très cher.

B) Aspects techniques

Nous allons définir maintenant les aspects techniques importants du projet, on va diviser cela en deux parties importantes, l'émetteur et le récepteur.

Après plusieurs réunions avec l'entrepreneur, nous avons conçus le cahier de charges suivant.

1) L'émetteur

On parle de la plaque électronique qui enverra le signal aux récepteurs situés sur les voitures, les éléments principaux qui constituent l'émetteur sont :

- Clavier
- LCD minimum 2x16
- Emetteur radiofréquence
- Source standard d'alimentation
- Protection pont de diodes pour éviter polarisation inverse
- Codeur pour envoi d'information chiffré (60 voitures)
- Protection face au bruit magnétique

2) Le récepteur

- Alimentation variable 70-110 [V]
- Récepteur et décodeur
- Système relais commande pédalier

IV) L'émetteur

A) Élection des composants principaux

1) Élection du micro contrôleur

C'est la partie la plus importante du circuit, elle va nous permettre contrôler tous les périphériques, tels que le clavier, le LCD, le codeur et l'émetteur radiofréquence. Nous avons choisi le PIC18F4580 pour les raisons suivantes, la première est la connaissance du même au sein de l'université, nous avons à notre disposition tout les logiciels et programmateurs de la gamme PIC de Microchip. Nous avons choisi alors un des micro contrôleurs plus puissants afin de ne pas être handicapés au niveau d'entrées/sorties du micro contrôleur, ni au niveau de puissance de calcul du même.

2) Choix du codeur

Le codeur que nous avons choisi est le Control 44 de Radiométrix

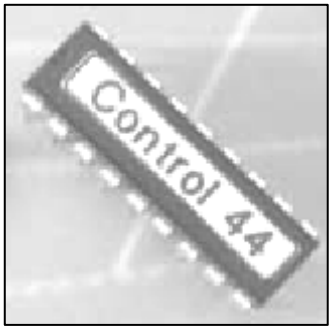


Figure 1 : Circuit Intégré du control 44 en support DIP

Ce codeur contient 4 bits d'adresse, puis 4 bits de données ce qui nous permet 16x4=64 possibilités, ce qui nous en accord avec le nombre maximum de voitures (60).

3)Choix de l'émetteur radiofréquences.

L'élection du récepteur est bien simple, la distance d'émission, cet engin est capable d'émettre 75 [m] en terrain défavorable et 300[m] en terrain ouvert, ce qui est en accord avec le cahier des charges. Sa fréquence d'émission est de 433,92 [Mhz] qui est réglementé. Son débit est petit 40 [bits/sec] et son alimentation de 5 [V]qui est plus que suffisant pour nos besoins. Finalement le composant est petit est pratique à installer.

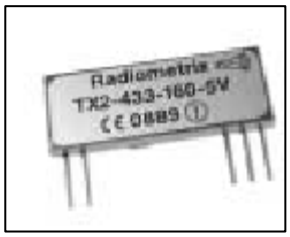


Figure 2 : Circuit Intégré de l'émetteur radiofréquence

4)Clavier et LCD

Pour le clavier la seule contrainte était qu'il soit résistant et bien sûr des touches suffisantes. Pour le LCD par contre nous sommes partis d'un display 2x16 caractères, mais n'étant pas commode à visualiser nous avons choisi un display de 4x20 qui était plus intéressant pour les besoins du client.

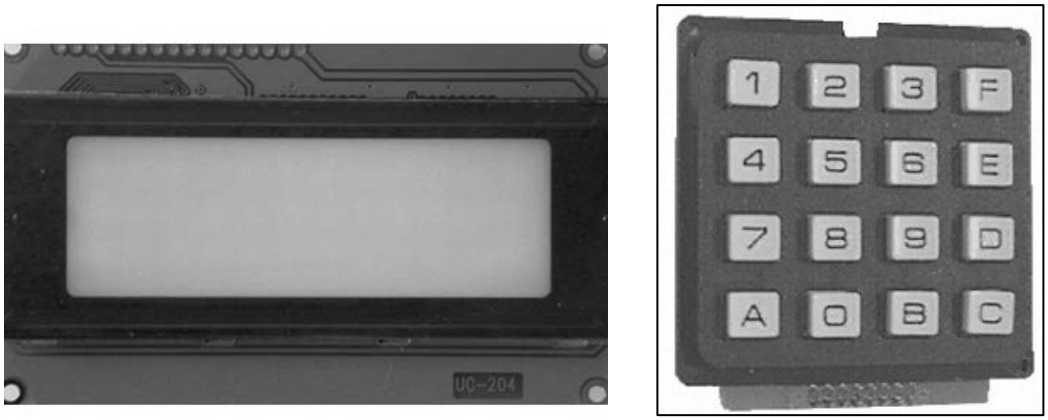


Figure 3 :LCD 4x20 et Clavier de 16 touches

B) Conception du circuit :

1) Schéma fonctionnel

Nous avons du organiser en premier lieu les périphériques du circuit, nous avons conçu alors le schéma fonctionnel suivant :

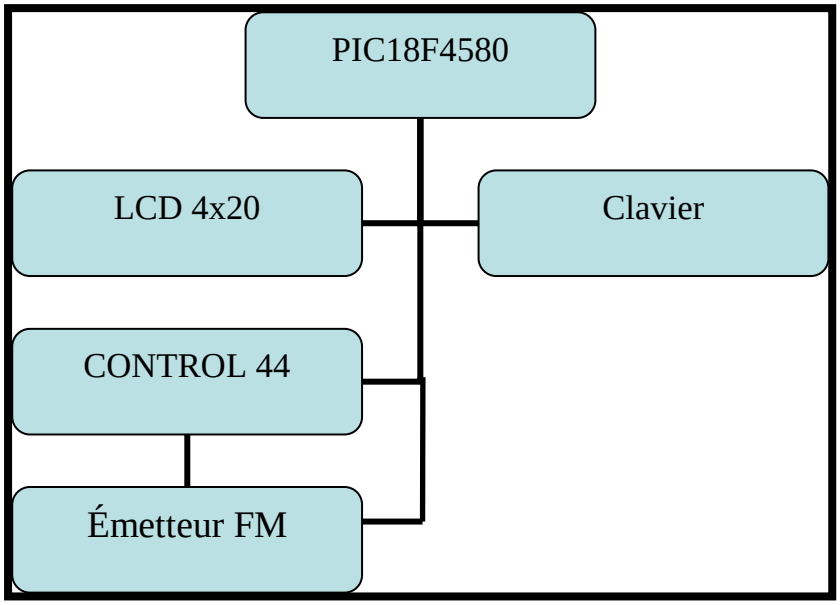


Figure 4 : Schéma fonctionnel

Le circuit sera conçu comme dans le schéma ci-dessus. On peut observer que l'émetteur FM est connecté au codeur et au micro contrôleur, car ça sera le calculateur qui enverra le signal d'émission des données.

2) Connexion du Codeur et émetteur FM

Le control 44 sera branché au PIC par les 4 bits d'adresse d'un coté, qui nous offrira la possibilité de coder en 16 combinaisons différentes, puis 4 bits en plus de données que l'on utilisera pour avoir les 64 combinaisons nécessaires. L'entrée SEND du codeur sera branchée micro contrôleur afin de choisir le moment pour stopper la voiture. Sur le schéma suivant nous pouvons observer les connexions réalisées sur les différents composants participant sur l'émission des données :

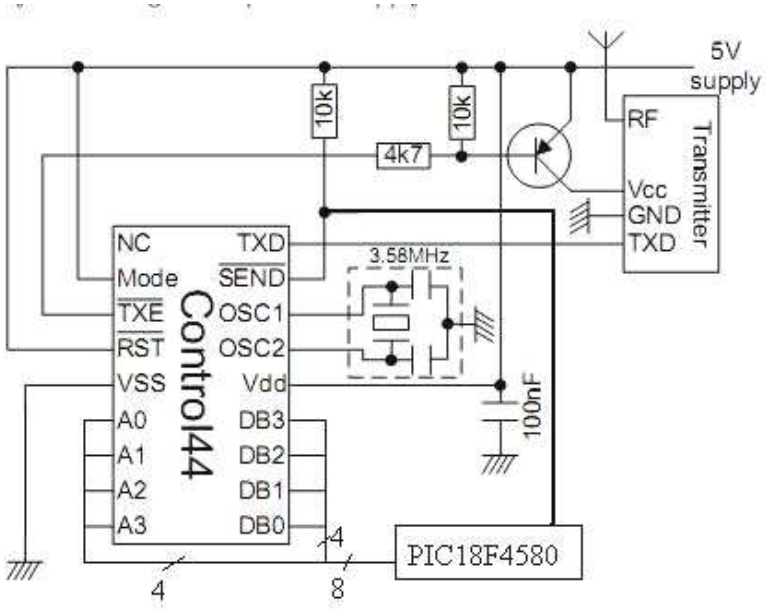


Figure 5 : Branchement du codeur et émetteur FM

3)Clavier et LCD

En premier lieu le clavier n'a aucune complication particulière sauf qu'il a besoin de résistances de Pull-Up, car le signal du PIC n'a pas suffisamment de courrant pour la boucle de retour. Pour le display, on l'a configuré en mode 4 bits, par simplification, on perd de la vitesse, mais ce n'est pas une contrainte pour notre circuit.

4)Conception finale

Lors du routage du circuit on a cherché à réduire le maximum l'influence de bruit, cela signifie que l'on doit tenir en compte toutes les sources de bruits de plaque. En premier lieu on a créé un plan de masse, les pistes de puissances sont courtes et directes, de plus, elles sont plus grandes pour éviter les différences de potentiel. Le signaux importants tels que l'horloge du calculateur sous placés le plus prés du même. Nous avons aussi placés un condensateur de 100 [nF] à coté de chaque circuit intégré et un condensateur élétrolitique à l'entrée et sortie du régulateur attiquement plus grands pour mieux filtrer le bruit.

Un autre aspect important est l'introduction d'un pont de diodes, pour éviter le branchement inverse sur la plaque, il s'agit simplement d'une mesure de protection. Tel que aussi la plaque sera recouverte d'un polymère afin de protéger la plaque de choques et poussières.

C) Le logiciel

1) Programmation du PIC

À l'aide du MPLAB ID fourni gratuitement par MICROCHIP qui est un outil de programmation et aussi à l'aide du compilateur MCC18 nous allons programmer notre carte. Le langage que l'on utilisera sera le langage C, plus simple que le langage assembleur et plus répandu que d'autres langages connus. D'autre part le MPLAB a l'option de débugger et de simuler ce qui nous permet d'avancer très rapidement lors de la programmation.

2) Chargement du logiciel

Pour charger le logiciel sur notre carte nous disposons d'un outil MPLAD qui nous permet de programmer notre PIC à partir du ICD 2. Cet outil nous donne accès aussi au débugger en temps réel.

3) Structure du programme

Afin de simplifier le programme nous diviserons tout le code en librairies, auquel le programme principal y accédera. Dans chaque librairie nous allons trouver des fonctions spécifiques pour gérer chaque partie principale des cartes.

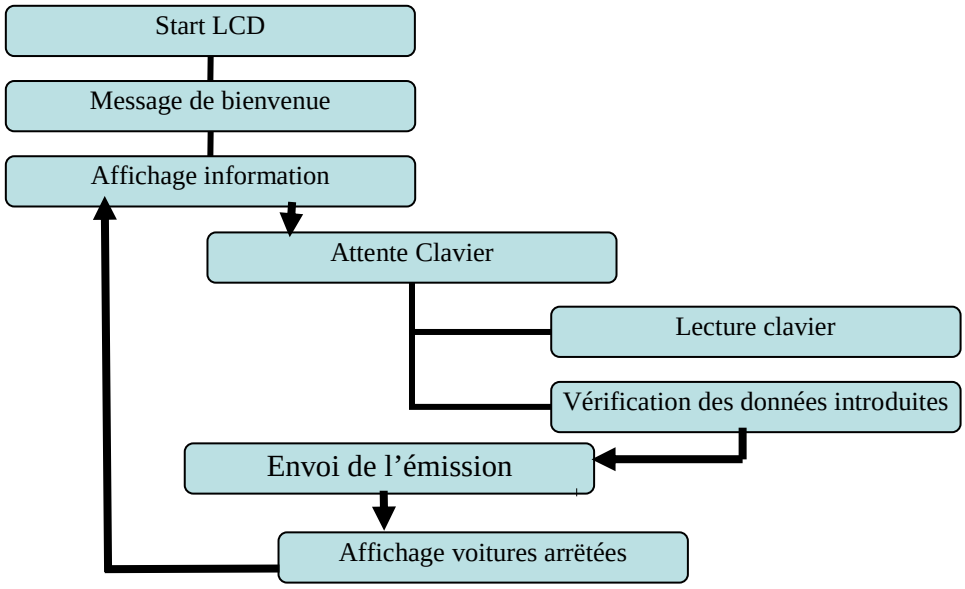


Figure 6 : Schéma structure du programme

- Programme principal, selectiondevoiturex4.c

Il est le responsable d'initialiser le PIC et appeler les librairies. Il contient les ordres nécessaires au contrôle de la plaque. Ces ordres sont les fonctions qui se trouvent dans les librairies.

- DelaysPAAD.h

Nous allons trouver les fonctions pour faire des délais ajustées au PIC 18F4580

- startLCDPAADx4.h :

Nous allons trouver la séquence d'initialisation du LCD sur 4 lignes qui suit le standard Hitachi.

- Control44PAAD.h :

Configuration des pattes du composant et de la fonction qui identifie le numéro de la voiture avec la carte.

- CommandsLCDPAADx4.h :

Configuration de toutes les fonctions qui nous permettront agir sur le LCD, pour écrire et changer des paramètres.

- CgramPAAD.h

Le LCD dispose de 8 positions de mémoire pour garder des caractères personnalises, dans cette librairie nous avons programmé quelques des caractères dont nous avons besoin pour les animations.

- kbdPAAD.h

Dans cette librairie nous allons trouver les fonctions pour lire les pulsations du clavier et le système pour les traduire au code ASCII qui nous permettra faire des opérations mathématiques ou écrire des données sur le LCD.

V) Le récepteur

Le récepteur est composé de quatre éléments principaux, la source d'alimentation, le récepteur, le décodeur et le système relais.

1) Source d'alimentation

La source d'alimentation a été le principal handicap car sur la voiture, on dispose que d'une tension à courant continu qui oscille entre 70 et 110 [V] en fonction du nombre de voitures actives à chaque moment.

Dans la première génération on a adapté comme solution un régulateur Maxim à 5V qui est en fait un diode zener à hautes prestations. Lorsque l'on a dessiné cette première carte, on ne connaissait pas encore qu'est-ce que on allait à commander exactement et lorsque finalement, nous avons connus les spécifications complètes on a dû refaire le système d'alimentation.

La solution que finalement on a trouvée est basé en deux diodes zener avec deux niveaux de tension, le première à 24 [V] pour alimenter la bobine du relais et le deuxième à 5,1 [V] pour alimenter le reste des composants. Comme la différence entre les deux tensions zener et l'entrée est grande, la puissance dissipé dans les résistances de polarisation est aussi importante, dans le pire des cas, 15 W et 1 W respectivement. On peut dire qu'énergétiquement la source d'alimentation n'est rien du tout efficiente mais c'est la solution plus simple et plus rentable que nous on a trouvé. On a pensé aussi pour un futur proche, et selon les possibilités de facilité d'implémentation que on devrait étudier la possibilité d'utiliser des lampes de couleurs pour ne pas perdre toute cette puissance dissipée en chaleur.

2) Réception

Le signal est reçu par la antenne et interprété par le binôme du émetteur, le RX2-333-40-5V qui la transfère au décodeur.

(a) Le décodeur

Le responsable de la décodification c'est le même composant que nous avons utilisé pour réaliser la codification des données, le contrôle 44, il identifie si la commande va dirigé à la voiture sur laquelle il se trouve grâce à l'identification par adresse sur 4 bits, et nous obtenons une sortie 0 ou 5 [V],

que nous utiliserons pour pouvoir activer ou désactiver le relais, c'est ici dans la première génération où nous nous sommes arrêtés.

(b) Le système relais

Le but final de la carte récepteur est de pouvoir agir sur le câble du pédalier, il s'agit alors de connecter le câble au relais de façon qu'il soit normalement fermé, de cette façon, après chaque reset de la foire, toutes les voitures seront prêtes pour fonctionner et le responsable pourra les arrêter à son critère.

Le relais choisi est de 24[V] pour avoir une baise consommation relative de courant (20 [mA]) et il supporte 10[A] satisfaisant la spécification d'au moins 6-7[A]. Pour commander le relais on va s'aider d'un transistor MOSFET 50[V], et du signal du control44 connecté à sa porte (Gate). Lorsque nous obtenons 5[V] le MOSFET s'active et la bobine du relais est mise à terre, le résultat est une coupure du courant sur le câble du pédalier. Pour protéger le circuit il ne faut pas oublier d'ajouter une diode en parallèle au relais pour absorber les tensions inverses produites par la bobine.

VI) Conclusion

En dépit de la contrainte de temps, je suis arrivé à réaliser l'aspect théorique, conceptuel et logiciel. Grâce à cela, j'ai réussi à réaliser un deuxième prototype, qui a respecté le cahier des charges initial et modifié tout au long du déroulement du projet. La communication avec l'entrepreneur m'a permis comprendre les besoins du même, afin de réaliser le produit souhaité.

Tout ce travail nous a permis comprendre les futures applications auquel se système peut aboutir puisqu'il avait de nombreuses contraintes électromagnétiques et physiques. Des applications tels que la domotique ou les produits de loisir peuvent être intéressants grâce à sa rentabilité.

VII) Bilan Personnel

Ce projet nous a permis suivre l'évolution d'un projet du début jusqu'à la fin du même, tel que dans l'entreprise nous avons du réalisé plusieurs réunions avec l'entrepreneur afin de concevoir le produit a la mesure de ces demandes, tout en négociant les possibilités réelles de même. Cette communication a permis de subir les expectatives initiales et évoluer les plaques avec un nouveau prototype qui respecte toutes les demandes du cahier de charges. La motivation au cours du projet n'a pas baissé car il a rassemblé plusieurs domaines, ce qui nous a permis d'apprendre à utiliser des nouveaux logiciels pour concevoir les plaques et pour les programmer.

Sans oublier que nous faisons partie d'une équipe nous avons divisé le travail tout en communicant constamment. Ceci à été un grand avantage car il y eu motivation mutuelle, nous avons appris les même choses, car d'une certaine façon nous nous sommes aidés constamment en réalisant une supervision mutuelle et un engagement pour avoir le zéro défaut. Et donc une qualité finale supérieure aux attendues, ce qui nous permis un gain de temps considérable.

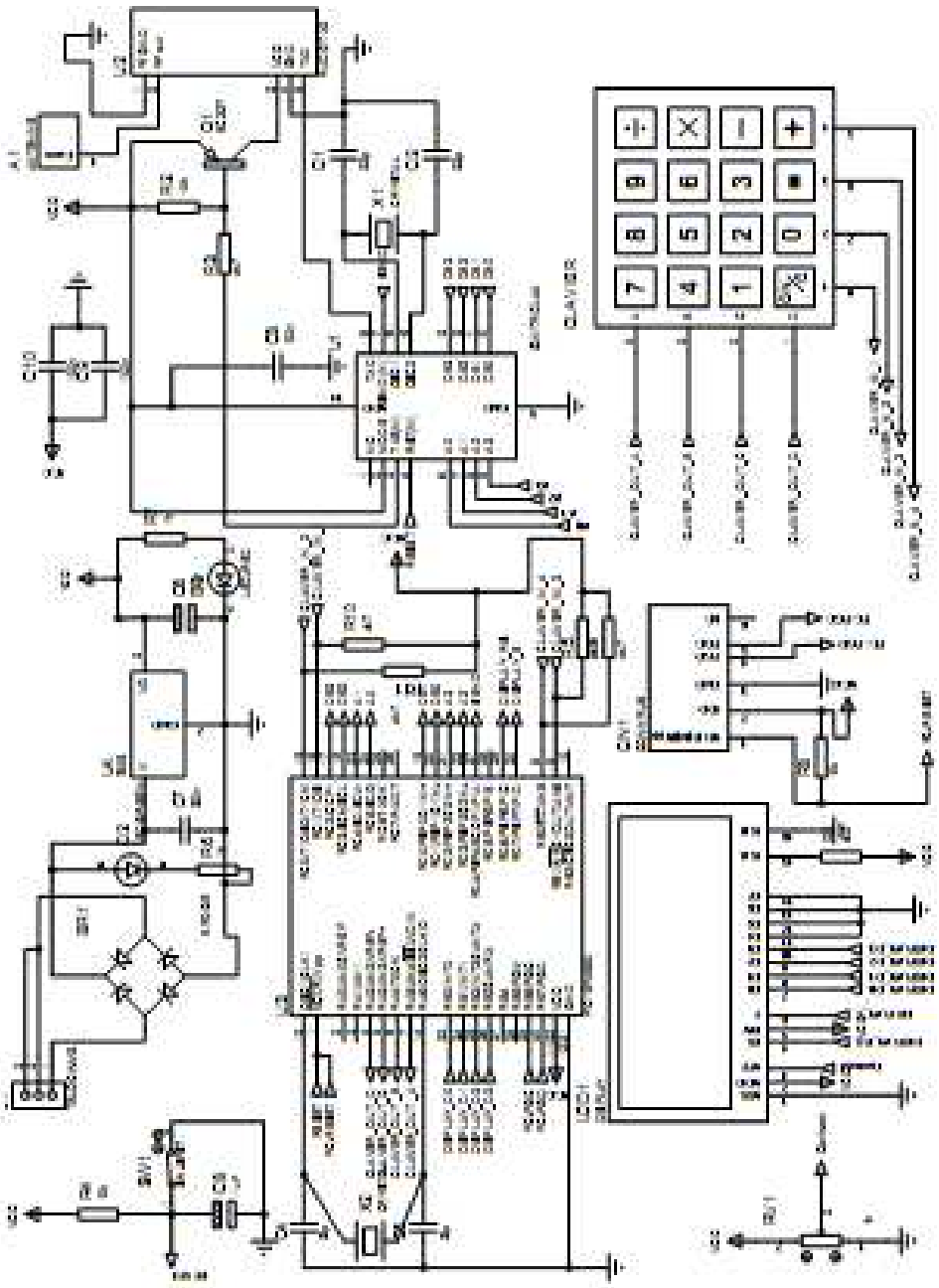
Nous sommes satisfaits de notre travail, nous avons appris beaucoup de choses, tel que le travail en équipe et la supervision du projet.

D'autre part nous avons documenté au maximum dans la mesure de la disponibilité de temps. Pour fournir aux futurs élèves une base qui peut aboutir à de nombreuses applications.

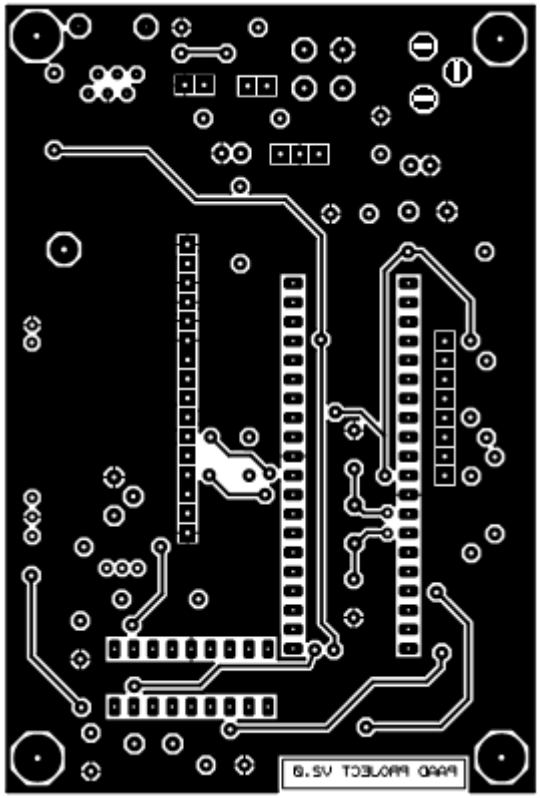
VIII) Annexes

A) ROUTAGE ÉMETTEUR 20
B) LAYOUT ÉMETTEUR 21
C) ROUTAGE RECEPTEUR 22
D) LAYOUT RECEPTEUR 23
E) CODE EMETTEUR 24
1) *Code principal* *24*
2) ***Librairie clavier*** *27*
3) ***Librairie Start LCD*** *30*
4) ***Librairie fonctions LCD*** *32*
5) ***Librairie control 44*** *37*
6) ***Librairie cgram*** *38*
F) PHOTOS 39
1) ***Version 1.0*** *39*
2) ***Version 2.0*** *40*

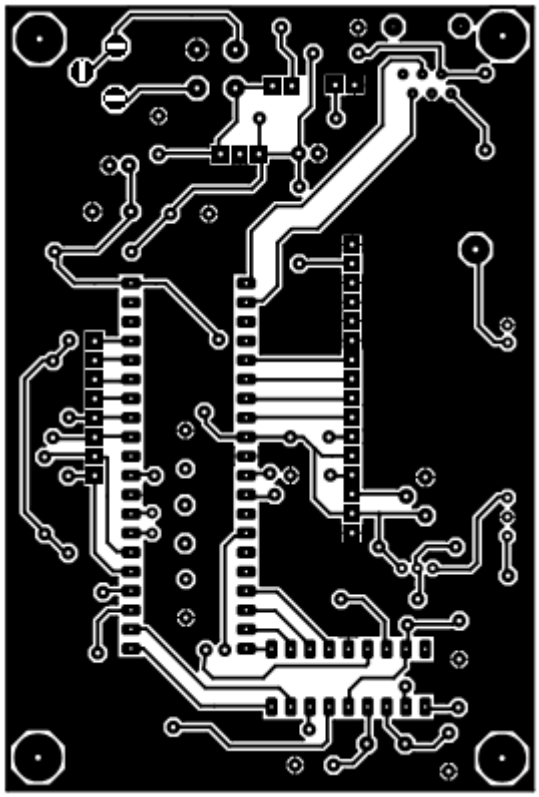
A) Routage Émetteur



B) Layout Émetteur

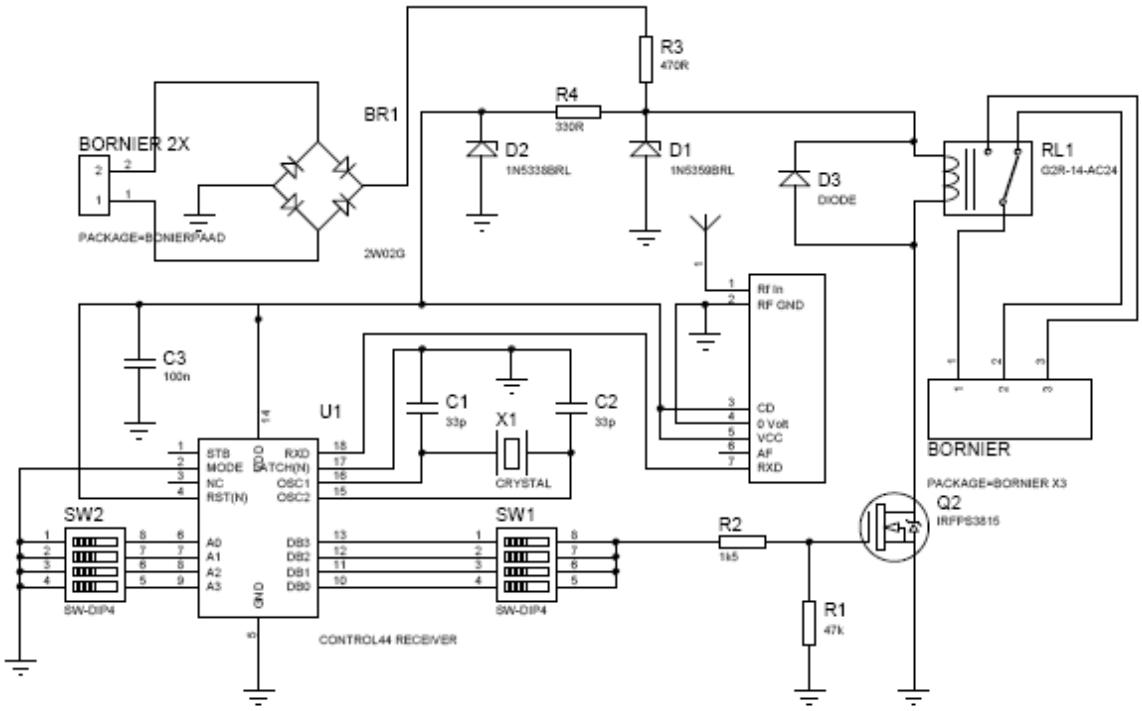


Top mirror

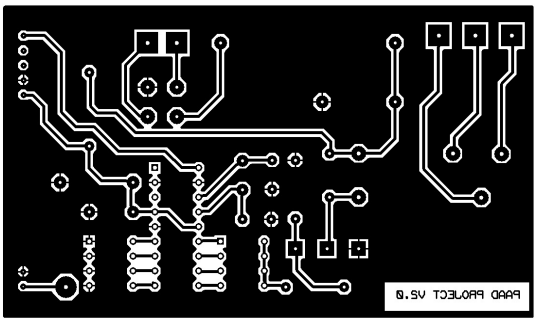


Bottom

C) Routage Récepteur



D) Layout Récepteur



E) Code Emetteur

1) Code principal

```
// selectiondevoiturex4.c

#include <p18f4580.h>
#include <p18cxxx.h>

#include <delaysPAAD.h> // definition des delays
#include <startLCDPAADx4.h> // initialization du LCD
#include <commandsLCDPAADx4.h> // librerie pour commander le LCD
#include <kbdPAAD.h> // librerie clavier
#include <control44PAAD.h> //librerie control44
#include <cgramPAAD.h> //librerie pour génerer des characteres


#define sendN   LATDbits.LATD4

#pragma config OSC = XT
#pragma config WDT = OFF

char presentation1[]=" WWW.ALPHAMATIC.NET ";
char presentation2[]="Control radio 433Mhz";
char presentation3[]="pour autotamponeuses";
char presentation4[]="Tel: +33-322-896-105";
//char presentation5[]="Avec la colaboration";
//char presentation6[]="  de la UPJV-GEII   ";
//char presentation7[]="Adrien Debut Gorgas ";
//char presentation8[]="Pablo A. Martinez A.";
char composezvoiture1[]="Tapez voiture entre";
char composezvoiture2[]="[00#63] ou (E)ffacer";
char num[]="num: ";
char demarrer[]="Demarrer...";
char arreter[]="Arreter...";

char choisir1[]="Selectionnez action:";
char choisir2[]="A arreter/D demarrer";
char choisir3[]="C annuler";
int f,c,detectado,i;
char tecla;
char control;
int numero,unidades,decenas;
int voiture[63]; // 0 roulant, 1 arrêté


void main(void){

// conf LCD//

////Configuration Ports en mode sortie
   TRISBbits.TRISB0=0; //Data pin 0
   TRISBbits.TRISB1=0; //Data pin 1
   TRISBbits.TRISB2=0; //Data pin 2
   TRISBbits.TRISB3=0; //Data pin 3

   TRISDbits.TRISD6=0; // Rs
   TRISDbits.TRISD7=0; // Enable

// fin conf LCD//

// conf Clavier

////Configuration des pates en mode:
// sortie (0) et mode entrée (1)
```

```
TRISCbits.TRISC1=0; //c1
```

```
TRISCbits.TRISC0=0; //c2
TRISEbits.TRISE1=0; //c3
TRISEbits.TRISE0=0; //c4

TRISAbits.TRISA5=1; //f1
TRISAbits.TRISA4=1; //f2
TRISAbits.TRISA3=1; //f3
TRISAbits.TRISA2=1; //f4

// Disable analog inputs
ADCON1 = 0x0F;

// fin conf Clavier;

//conf control44

TRISCbits.TRISC5=0; //a0
TRISCbits.TRISC4=0; //a1
TRISDbits.TRISD3=0; //a2
TRISDbits.TRISD2=0; //a3

TRISDbits.TRISD1=0; //d0
TRISDbits.TRISD0=0; //d1
TRISCbits.TRISC3=0; //d2
TRISCbits.TRISC2=0; //d3

TRISDbits.TRISD4=0; //SendN--> (SendN=0,envoyer) (SendN=1,rien)

// fin conf control44

//test control44, Inicialitation de pates adresse et donnée
a0=a1=a2=a3=d0=d1=d2=d3=1;
//////////////////////////////////////////////////////

start_lcd(); //demarrage du LCD
ecrire_lcd(0,blink_off);
ecrire_lcd(0,cursor_off);
lignex_lcd(1,presentation1,100);
lignex_lcd(2,presentation2,100);
lignex_lcd(3,presentation3,100);
lignex_lcd(4,presentation4,100);
delay_xs(4);


/////////
c1=c2=c3=c4=1; //
sendN=1;

/////// Initialisation des états des voitures
for (i=0;i<64;i++) // situation des voitures: 0 roulant, 1 arrêté,
{
voiture[i]=0;
}


////// programme recursif
while(1)
   {
   ecrire_lcd(0,clear_lcd);
   //état des voitures sur la quatrième ligne
   i=0;
   ecrire_lcd(0,return_home4);
   for (i=0;i<64;i++)
      {
      if (voiture[i]==1)
         {
         numero=i;
         decenas=(numero/10);
         unidades=(numero-decenas*10);
         decenas=decenas+48; //conversion vers ASCII
         unidades=unidades+48; //conversion vers ASCII
         ecrire_lcd(1,decenas);
```

```
        ecrire_lcd(1,unidades);
        ecrire_lcd(1,';');
        }
    }
//
inicio: // étiquette
lignex_lcd(1,composezvoiture1,0); // demande de numero
lignex_lcd(2,composezvoiture2,0); // ou effacer
lignex_lcd(3,num,0);
// on active le cursor et le blink pour le champ numerique
ecrire_lcd(0,cursor_on);
ecrire_lcd(0,blink_on);
// on lit la première touche, on ignore les pulsations incorrectes
tecla='N';
 while   ((tecla=='N')|(tecla=='A')|(tecla=='B')|(tecla=='C')|(tecla=='D')|(tecla=='F')|(tecla=='7')|(tecla=='8')|(tecla=='9'))   //attente   première
     cifre
    {
    tecla=get_char();
    delay_xms(10);
    }
// si la touche est E, on initialise les états des voitures et on saute à l'étiquette inicio, sinon on continue
if (tecla=='E')
    {
    for (i=0;i<64;i++) // situation des voitures: 0 roulant, 1 arrêté,
        {
        voiture[i]=0;
        }
    ecrire_lcd(0,clear_lcd);
    goto inicio;
    }
//Conversion du valeur ASCII au integer
decenas=tecla-48;
ecrire_lcd(1,tecla);
//
tecla='N';
// si la première touche est 6 on doit apliquer des resctrictions pour la seconde
if (decenas==6)
    {
    while
     ((tecla=='N')|(tecla=='A')|(tecla=='B')|(tecla=='C')|(tecla=='D')|(tecla=='E')|(tecla=='F')|(tecla=='4')|(tecla=='5')|(tecla=='6')|(tecla=='7')|(t
     ecla=='8')|(tecla=='9')) //attente deuxième cifre
        {
        tecla=get_char();
        delay_xms(10);
        }
    }
else
    {
    while ((tecla=='N')|(tecla=='A')|(tecla=='B')|(tecla=='C')|(tecla=='D')|(tecla=='E')|(tecla=='F')) //attente deuxième cifre
        {
        tecla=get_char();
        delay_xms(10);
        }
    }
//on a déjà obtenu les deux chifres
ecrire_lcd(0,blink_off);
ecrire_lcd(0,cursor_off);
// converion du valeur ASCII au integer de la deuxième touche
unidades=tecla-48;
ecrire_lcd(1,tecla);
numero=(decenas*10)+unidades; // conversion dizaines + unitées = integer
select_voiture(numero); //on envoye le numero aux pates
delay_xs(1);
ecrire_lcd(0,clear_lcd);
//menu
lignex_lcd(1,choisir1,0);
lignex_lcd(2,choisir2,0); //arrêter ou demarrer
lignex_lcd(3,choisir3,0); //annuler
// on attend la selection et on applique des restrictions
tecla='N';
while ((tecla=='N')|((tecla!='D')&(tecla!='A')&(tecla!='C')))
    {
```

```
      tecla=get_char();
      delay_xms(10);
      }
   if (tecla=='D') // Demarrer data + Logos + signal
         {
         d0=d1=d2=d3=1; // data demarrer
         voiture[numero]=0; // changer état
         //logos
         ecrire_lcd(0,clear_lcd);
         lignex_lcd(1,demarrer,0);
         loading(2);
         cursor_right(1);
         ligne_lcd(num,0);
         ecrire_lcd(1,decenas+48);
         ecrire_lcd(1,unidades+48);

         coche(0);
         coche(4);
         coche(8);
         coche(12);
         coche(16);
         clear_lignex(2,0);
         clear_lignex(3,0);
         clear_lignex(4,0);
         //signal 100 fois
         for (i=0;i<100;i++)
            {
            sendN=0;
            delay_xms(20); // le pulse ne doit pas être superieur à 50ms.
            sendN=1;
            }// fin de for
         } //fin de if
   else if (tecla=='A') // Arreter
         {
         d0=d1=d2=d3=0; //data
         voiture[numero]=1; // changer data
         // Logos
         ecrire_lcd(0,clear_lcd);
         lignex_lcd(1,arreter,0);
         loading(2);
         coche(1);
         ecrire_lcd(0,return_home3);
         cursor_right(6);
         //ligne_lcd(num,0);
         ecrire_lcd(1,0b01111110);
         cursor_right(1);
         ecrire_lcd(1,decenas+48);
         ecrire_lcd(1,unidades+48);
         cursor_right(1);
         ecrire_lcd(1,0b01111110);
         stop(13);
         // signal 100 fois
         for (i=0;i<100;i++)
            {
            sendN=0;
            delay_xms(20); // le pulse ne doit pas être superieur à 50ms.
            sendN=1;
            }
         }
   else if (tecla=='C') // Annuler
         {
         // on fait rien
         }

   }//fin de while
}//fin de main
```

2)Librairie clavier

```
// kbdPAAD.h
```

```
//
// Fonction qui donne comme sortie la touche poussée
// ou 'N' en cas contraire
//get_char();

#ifndef __KBDPAAD_H
#define __KBDPAAD_H

#include <p18f4580.h>
#include <p18cxxx.h>

#include <delaysPAAD.h> // definition des delays

// definition des ports du clavier:
// LAT pour les sorties (colonnes), PORT pour les entrées (files)

#define c1  LATCbits.LATC1
#define c2  LATCbits.LATC0
#define c3  LATEbits.LATE1
#define c4  LATEbits.LATE0

#define f1  PORTAbits.RA5
#define f2  PORTAbits.RA4
#define f3  PORTAbits.RA3
#define f4  PORTAbits.RA2

//////////////////////////
char get_char(void);
char lire_file(void);

//// Dispositon du clavier:
char const keys[4][4] = {{'1','2','3','F'},
                         {'4','5','6','E'},
                         {'7','8','9','D'},
                         {'A','0','B','C'}};

//variables globales
int f,c;

//fonction pour lire une touche
char get_char(void)
{
char sortie,file;
sortie='N';
for (c=0;c<4;c++) // On parcourt les 4 colonnes
                  {
                          switch(c)
                                  { // on prends la colonne C
                          case 0 :    c1=0;                   // on envoye un cero à c1
                                      file=lire_file(); // on lit toutes les files
                                      c1=1;                   // on laisse la colonne à 1
                                      if (file!='N')
                                          {sortie=file;}
                                      break;                  // fin de case 0 de c

                          case 1 :    c2=0;
                                      file=lire_file();
                                      c2=1;
                                      if (file!='N')
                                          {sortie=file;}
                                      break;


                          case 2 :    c3=0;
                                      file=lire_file();
                                      c3=1;
                                      if (file!='N')
                                          {sortie=file;}
                                      break;

                          case 3 :    c4=0;
                                      file=lire_file();
```

```
                                                c4=1;
                                                if (file!='N')
                                                        {sortie=file;}
                                                break;

                                        }//fin de switch
                }// fin de for de files
return sortie;
} // fin de get_char();


//fonction lire file (on l'utilise pour economiser lignes de programme)


char lire_file(void)
{
char touche;
touche='N';
for (f=0;f<4;f++) // On parcourt toutes les files
                                        { switch (f)            { //On prend la file f
                                                case 0:
                                                                        if
(f1==0){touche=keys[f][c];}
                                                                        delay_xms(10);

                                                                        break;
                                                case 1:
                                                                        if
(f2==0){touche=keys[f][c];}
                                                                        delay_xms(10);
                                                                        break;
                                                case 2:
                                                                        if
(f3==0){touche=keys[f][c];}
                                                                        delay_xms(10);
                                                                        break;
                                                case 3:
                                                                        if
(f4==0){touche=keys[f][c];}
                                                                        delay_xms(10);
                                                                        break;
                                                                        } //fin de switch
file
                                        } // fin de for
return touche;
}

#endif
```

3)Librairie Start LCD

```
//startLCDPAAD.h
//Definition des pins que on va utiliser comme Data, Rs et Enable.
//Introduction des commandes pour le demarrage et configuration du LCD.
//Les données sont envoyées de 8 en 8 mais le LCD reçoit seulement les
// 4 derniers dont il faut couper les intructions de 4 en 4 bits.
// ex.  donnée=0b10110010, on envoye 0b00001011 et après 0b00000010



#ifndef __STARTLCDPAADX4_H
#define __STARTLCDPAADX4_H

#include <p18f4580.h>
#include <p18cxxx.h>
#include <delaysPAAD.h>
#define Enable  LATDbits.LATD7
#define Rs                      LATDbits.LATD6
#define Data      LATB

#include <cgramPAAD.h> //librerie pour génerer des characteres


void start_lcd(void);

/****************Inicio LCD***********************/
void start_lcd(void){

//Reset Pins
Enable=0;
Rs=0; //command
Data=0b00000000;

//Attente 15 ms
delay_xms(15);

//Function set1
Data=0b00000011;
Enable=1;
delay_xus(40);
Enable=0;

//Atente 4,1 ms
delay_xms(5);

//Function set2
Data=0b00000011;
Enable=1;
delay_xus(40);
Enable=0;

//Attente 100 us
delay_xus(110);

//Function set 3
Data=0b00000011;
Enable=1;
delay_xus(40);
Enable=0;

//Entrée comandes:

//Interface 4 bits
Data=0b00000010; // on sélectionne 4 bit
Enable=1;
delay_xus(40);
Enable=0;
//Fuction set
```

```
Data=0b00000010; // On répèt parce que on a sélectionné 4 bit
Enable=1;
delay_xus(40);
Enable=0;
Data=0b00001000; // 2 lignes et caractère 5x7
Enable=1;
delay_xus(40);
Enable=0;

//Display off
Data=0b00000000;
Enable=1;
delay_xus(40);
Enable=0;
Data=0b00001000;
Enable=1;
delay_xus(40);
Enable=0;

//Display Clear
Data=0b00000000;
Enable=1;
delay_xus(40);
Enable=0;
Data=0b00000001;
Enable=1;
delay_xus(40);
Enable=0;

delay_xms(2); // Le display clear est plus lente.

//Entry mode set
Data=0b00000000;
Enable=1;
delay_xus(40);
Enable=0;
Data=0b00000110; // Cursor increase, no shift
Enable=1;
delay_xus(40);
Enable=0;

//Display on, cursor on, blinking on
Data=0b00000000;
Enable=1;
delay_xus(40);
Enable=0;
Data=0b00001111;
Enable=1;
delay_xus(40);
Enable=0;

//genere les characteres personalises
genchar();
//

}

//End Initialization



#endif
```

4)Librairie fonctions LCD

```
//commandsLCDPAADx4.h
//
//Liste des commandes typiques pour gérer le LCD (voir datasheet):
//
//ecrire_lcd(type,data); pour envoyer un seule caractère ou commande
//lignex_lcd(phrase,vitesse); pour envoyer une phrase à la ligne x
//deuxlignes_lcd(ligne1,ligne2,vitesse); aux deux lignes au meme temp
//rebondir(espaces,vitesse) pour faire rebondir le display
//clear_lignex(ligne,vitesse); pour effacer la ligne x

//lignex_shiftingwrite(phrase,vitesse,sense); pour faire apparaître le texte shifting de un coté ou l'autre
//cursor_right(position);


#ifndef __COMMANDSLCDPAADx4_H
#define __COMMANDSLCDPAADx4_H

#include <p18f4580.h>
#include <p18cxxx.h>
#include <delaysPAAD.h>

void ecrire_lcd(char type, unsigned char caractere);
void rebondir(int espaces,int ms);
void ligne_lcd(char chaine[],int vitesse_ms);
void lignex_lcd(int ligne,char chainex[],int vitesse_ms);
void deuxlignes_lcd(char cadena1[],char cadena2[],int vitesse_ms);
void clear_lignex(int ligne,int vitesse_ms);
void lignex_shiftingwrite(char chainex[],int vitesse_ms);
void cursor_right(int numero);
void loading(int fois);
void stop(int pos);
void coche(int pos);


#define clear_lcd                    0b00000001 // clear lcd, cursor move to first digit, first line
#define return_home1        0b10000000 // cursor move to first digit, first line
#define return_home2        0b11000000 // cursor move to first digit, second line
#define return_home3        0b10010100 // 0x94 cursor move to first digit, third line
#define return_home4        0b11010100 // 0xD4 cursor move to first digit, fourth line

/* Display ON/OFF Control defines */
#define display_on  0b00001111  /* Display on      */
#define display_off 0b00001011  /* Display off     */
#define cursor_on   0b00001111  /* Cursor on       */
#define cursor_off  0b00001100  /* Cursor off      */
#define blink_on    0b00001111  /* Cursor Blink    */
#define blink_off   0b00001110  /* Cursor No Blink */

/* Cursor or Display Shift defines */
#define shift_cur_left    0b00010011  /* Cursor shifts to the left   */
#define shift_cur_right   0b00010111  /* Cursor shifts to the right  */
#define shift_disp_left   0b00011011  /* Display shifts to the left  */
#define shift_disp_right  0b00011111  /* Display shifts to the right */


//////////////////////////////////////////////

/*************ecrire_lcd*****************************/
void ecrire_lcd(char type, unsigned char caractere){

unsigned upper;
unsigned lower;
lower=caractere & 0b00001111; // on prend les 4 premieres bit
upper=(caractere & 0b11110000)>>4; // on prend les 4 dernieres bit et on les decale 4 positions a droite

if (type==1) // data pour montrer sur le LCD
{       Rs=1; } // type texte
```

```
else {Rs=0;} // type command

                    delay_ms();

                    Data = upper;
                    delay_xus(40);
                    Enable=1;
                    delay_xus(40);
                    Enable=0;
                    delay_xus(40);

                    Data = lower;
                    delay_xus(40);
                    Enable=1;
                    delay_xus(40);
                    Enable=0;
                    delay_xus(40);

}

//////////////////////////////////////////////////

void rebondir(int espaces,int vitesse_ms){
int p;
for (p=0;p<espaces;p++){
        delay_xms(vitesse_ms);
        ecrire_lcd(0,shift_disp_right);}
for (p=0;p<espaces;p++){
delay_xms(vitesse_ms);
ecrire_lcd(0,shift_disp_left);}
}

//////////////////////////////////////////////////
void lignex_shiftingwrite(char chainex[],int vitesse_ms){
int i;
i=0;
ecrire_lcd(0,0b10001111); // cursor move first line, last position
delay_xs(2);
while (chainex[i]!='\0') { //escribe hasta encontrar fin de array
                    ecrire_lcd(1,chainex[i]);
                    delay_xms(vitesse_ms);
                    ecrire_lcd(0,shift_disp_left);
                    i++;
                    }

}
//////////////////////////////////////////////
void cursor_right(int numero){
int i;
for (i=0;i<numero;i++)
        {ecrire_lcd(0,shift_cur_right);}
}

void cursor_left(int numero){
int i;
for (i=0;i<numero;i++)
        {ecrire_lcd(0,shift_cur_left);}
}
//////////////////////////////////////////////
void ligne_lcd(char chaine[],int vitesse_ms){
int i;
i=0;
        while (chaine[i]!='\0') { //escribe hasta encontrar fin de array
                    ecrire_lcd(1,chaine[i]);
                    delay_xms(vitesse_ms);
                    i++;
                    }
}
//////////////////////////////////////////////

//////////////////////////////////////////////
void lignex_lcd(int ligne,char chainex[],int vitesse_ms){
```

```
int i;
switch (ligne)
{

case 1:
	i=0;
	clear_lignex(1,0);		// borra y lleva Cursor al inicio de la 1ª línea
	while (chainex[i]!='\0') { //escribe hasta encontrar fin de array
			ecrire_lcd(1,chainex[i]);
			delay_xms(vitesse_ms);
			i++;
			}
break;

case 2:
	i=0;
	clear_lignex(2,0);
	ecrire_lcd(0,return_home2);		// Cursor al inicio de la 2ª línea
	while (chainex[i]!='\0') { //escribe hasta encontrar fin de array
			ecrire_lcd(1,chainex[i]);
			delay_xms(vitesse_ms);
			i++;
			}
break;

case 3:
	i=0;
	clear_lignex(3,0);
	ecrire_lcd(0,return_home3);		// Cursor al inicio de la 2ª línea
	while (chainex[i]!='\0') { //escribe hasta encontrar fin de array
			ecrire_lcd(1,chainex[i]);
			delay_xms(vitesse_ms);
			i++;
			}

break;

case 4:
	i=0;
	clear_lignex(4,0);
	ecrire_lcd(0,return_home4);		// Cursor al inicio de la 2ª línea
	while (chainex[i]!='\0') { //escribe hasta encontrar fin de array
			ecrire_lcd(1,chainex[i]);
			delay_xms(vitesse_ms);
			i++;
			}
break;
}// switch
}// fin function
//////////////////////////////////////////////

//void ligne2_lcd(char chaine2[],int vitesse_ms){
//int i;
//	i=0;
//	ecrire_lcd(0,return_home2);		// Cursor al inicio de la 1ª línea
//	while (chaine2[i]!='\0') { //escribe hasta encontrar fin de array
//			ecrire_lcd(1,chaine2[i]);
//			delay_xms(vitesse_ms);
//			i++;
//			}
//}

//////////////////////////////////////////////

/*******************écriture de 2 strings sur les 2 lignes du LDC*****/

void deuxlignes_lcd (char ligne1[],char ligne2[], int vitesse_ms)
{
	int i;
	i=0;
	ecrire_lcd(0,clear_lcd); // Borra todo y lleva Cursor al inicio de la 1ª línea
	while (ligne1[i]!='\0') { //escribe hasta encontrar fin de array
```

```
                ecrire_lcd(1,ligne1[i]);
                delay_xms(vitesse_ms);
                i++;
                }
        i=0;
        ecrire_lcd(0,return_home2);             // Cursor al inicio de la 2ª línea
        while (ligne2[i]!='\0') { //escribe hasta encontrar fin de array
                ecrire_lcd(1,ligne2[i]);
                delay_xms(vitesse_ms);
                i++;
                }

}
//////////////////////////////////////////////////////////////////////////////

void clear_lignex(int ligne,int vitesse_ms)
{
int i;
switch (ligne)
{
case 1:
        ecrire_lcd(0,return_home1);             // Cursor al inicio de la 1ª línea
        for (i=0;i<20;i++)
                {
                ecrire_lcd(1,0b00100000);
                delay_xms(vitesse_ms);
                }
        ecrire_lcd(0,return_home1);
break;

case 2:
        ecrire_lcd(0,return_home2);             // Cursor al inicio de la 2ª línea
        for (i=0;i<20;i++)
                {
                ecrire_lcd(1,0b00100000);
                delay_xms(vitesse_ms);
                }
        ecrire_lcd(0,return_home2);
break;

case 3:
ecrire_lcd(0,return_home3);                     // Cursor al inicio de la 2ª línea
        for (i=0;i<20;i++)
                {
                ecrire_lcd(1,0b00100000);
                delay_xms(vitesse_ms);
                }
        ecrire_lcd(0,return_home3);
break;

case 4:
ecrire_lcd(0,return_home4);                     // Cursor al inicio de la 2ª línea
        for (i=0;i<20;i++)
                {
                ecrire_lcd(1,0b00100000);
                delay_xms(vitesse_ms);
                }
        ecrire_lcd(0,return_home4);
break;


}
}

void loading(int fois)
{
        int i;
        for (i=0;i<fois;i++)
        {
        ecrire_lcd(1,'|');
        delay_xms(200);
        ecrire_lcd(0,shift_cur_left);
```

```
        ecrire_lcd(1,'/');
        delay_xms(200);
        ecrire_lcd(0,shift_cur_left);
        ecrire_lcd(1,'-');
        delay_xms(200);
        ecrire_lcd(0,shift_cur_left);
        ecrire_lcd(1,0); //contrabarra cgram 64
        delay_xms(200);
        ecrire_lcd(0,shift_cur_left);
        }
        ecrire_lcd(1,'*');
}

void stop(int pos)
{
        int i;
        ecrire_lcd(0,return_home2);
        cursor_right(pos);
        ecrire_lcd(1,2);
        for (i=0;i<4;i++)
                {
                ecrire_lcd(1,0b11111111);
                }
        ecrire_lcd(1,1);
        ecrire_lcd(0,return_home3);
        cursor_right(pos);
        ecrire_lcd(1,0b11111111);
        ecrire_lcd(1,'S');
        ecrire_lcd(1,'T');
        ecrire_lcd(1,'O');
        ecrire_lcd(1,'P');
        ecrire_lcd(1,0b11111111);
        ecrire_lcd(0,return_home4);
        cursor_right(pos);
        ecrire_lcd(1,3);
        for (i=0;i<4;i++)
                {
                ecrire_lcd(1,0b11111111);
                }
        ecrire_lcd(1,4);
}
void coche(int pos)
{
        int i;
        ecrire_lcd(0,return_home2);
        cursor_right(pos);
        ecrire_lcd(1,0b11111111);
        ecrire_lcd(0,return_home3);
        cursor_right(pos);
        ecrire_lcd(1,0b11111111);
        cursor_right(1);
        ecrire_lcd(1,0b10110010);
        ecrire_lcd(1,1);
        ecrire_lcd(0,return_home4);
        cursor_right(pos);
        for (i=0;i<4;i++)
                {
                ecrire_lcd(1,0b11111111);
                }

}

void coche2(void)
{
}
void coche3(void)
{
}

#endif
```

5)Librairie control 44

```
// control44PAAD.h
// Configuration des pates pour commander le control44
//fonction pour selectioner une voiture
//select_voiture(int);



#ifndef __CONTROL44PAAD_H
#define __CONTROL44PAAD_H

#include <p18f4580.h>
#include <p18cxxx.h>



// LAT pour numero de voiture

#define a0  LATCbits.LATC5
#define a1  LATCbits.LATC4
#define a2  LATDbits.LATD3
#define a3  LATDbits.LATD2

// LAT pour donnée envoyé

#define d0  LATDbits.LATD1
#define d1  LATDbits.LATD0
#define d2  LATCbits.LATC3
#define d3  LATCbits.LATC2


//
void select_voiture(int numero)
{// on peut commander 16 groupes de 4 voitures chaque un.
int groupe;
groupe=numero/4; //on obtient le groupe 0bxxxx
a3=(groupe & (0b1000))>>3;
a2=(groupe & (0b0100))>>2;
a1=(groupe & (0b0010))>>1;
a0=(groupe & (0b0001));
}//fin select_voiture


#endif
```

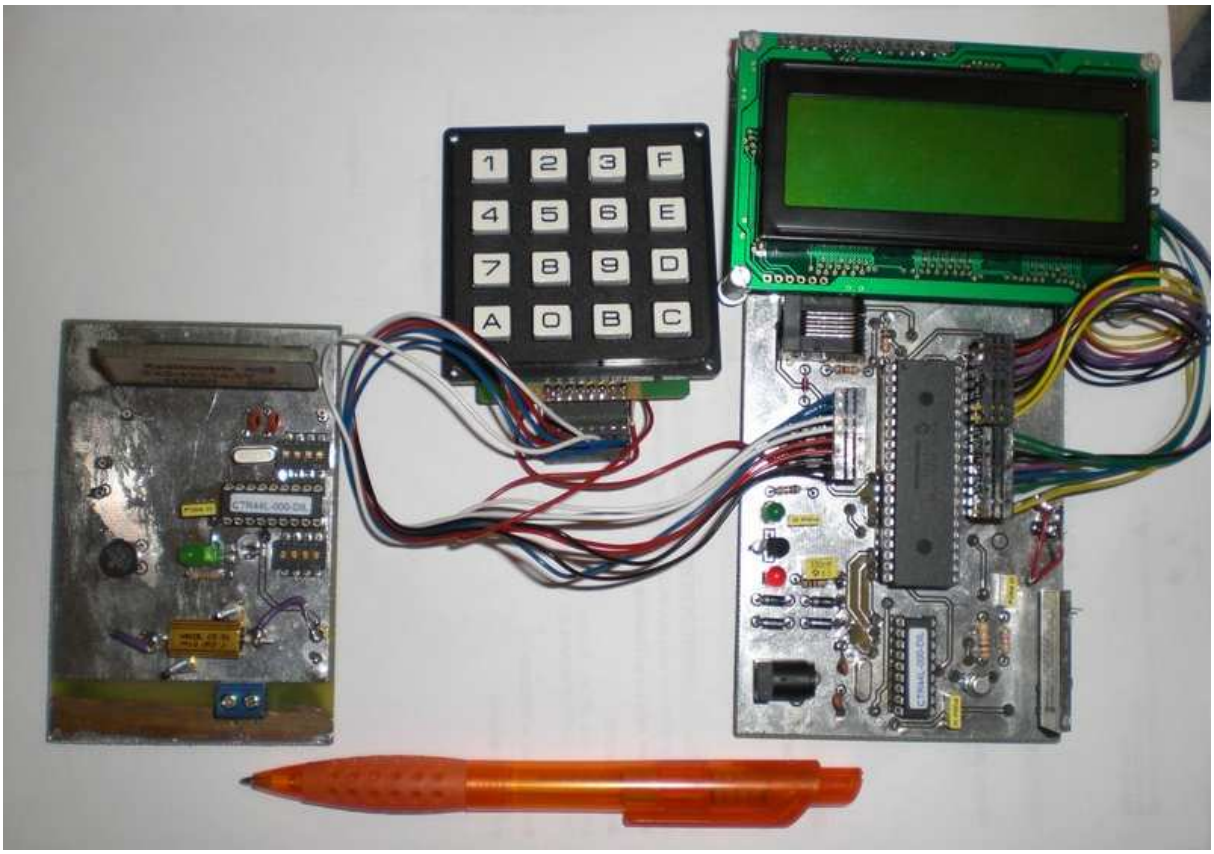
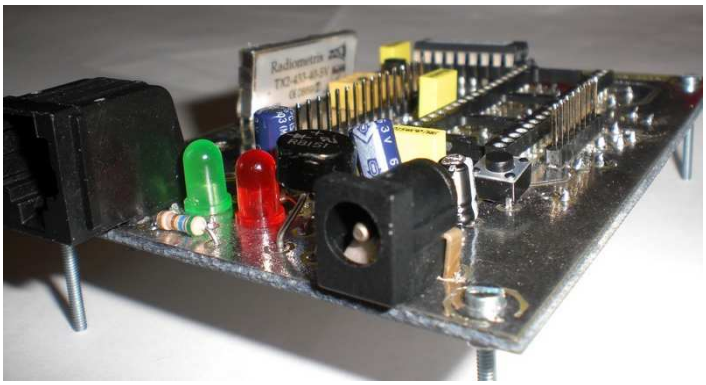
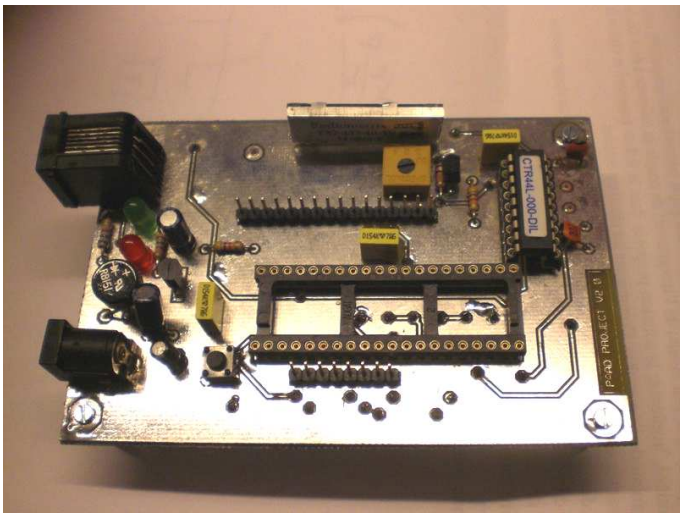
6)Librairie cgram

```
//cgramPADD.h

/******************DELAYS************************/
#ifndef __CGRAMPAAD_H
#define __CGRAMPAAD_H

#include <p18f4580.h>
#include <p18cxxx.h>
#include <commandsLCDPAADx4.h>

// cgram 64,72,80,88,96,104,112,120/ASCII 0 1 2 3 4 5 6 7
//ecrire_lcd(0,0b0.XXXXXXX);
//generateur de charactere contrabarra
//
void genchar(void)
{
//generateur de charactere contrabarra
ecrire_lcd(0,0b01000000); //set cgram 64 / ecrire_lcd(1,0)
ecrire_lcd(1,0b00000000); //file 0
ecrire_lcd(1,0b00010000); //file 1
ecrire_lcd(1,0b00001000); //file 2
ecrire_lcd(1,0b00000100); //file 3
ecrire_lcd(1,0b00000010); //file 4
ecrire_lcd(1,0b00000001); //file 5
ecrire_lcd(1,0b00000000); //file 6
ecrire_lcd(1,0b00000000); //file 7
ecrire_lcd(0,return_home1);

//coin superiur droite

ecrire_lcd(0,0b01001000); //set cgram 72 / ecrire_lcd(1,1)
ecrire_lcd(1,0b00010000); //file 0
ecrire_lcd(1,0b00011000); //file 1
ecrire_lcd(1,0b00011100); //file 2
ecrire_lcd(1,0b00011110); //file 3
ecrire_lcd(1,0b00011111); //file 4
ecrire_lcd(1,0b00011111); //file 5
ecrire_lcd(1,0b00011111); //file 6
ecrire_lcd(1,0b00011111); //file 7
ecrire_lcd(0,return_home1);

//coin superiur gauche

ecrire_lcd(0,0b01010000); //set cgram 80 / ecrire_lcd(1,2)
ecrire_lcd(1,0b00000001); //file 0
ecrire_lcd(1,0b00000011); //file 1
ecrire_lcd(1,0b00000111); //file 2
ecrire_lcd(1,0b00001111); //file 3
ecrire_lcd(1,0b00011111); //file 4
ecrire_lcd(1,0b00011111); //file 5
ecrire_lcd(1,0b00011111); //file 6
ecrire_lcd(1,0b00011111); //file 7
ecrire_lcd(0,return_home1);

//coin inferieur gauche

ecrire_lcd(0,0b01011000); //set cgram 88 / ecrire_lcd(1,3)
ecrire_lcd(1,0b00011111); //file 0
ecrire_lcd(1,0b00011111); //file 1
ecrire_lcd(1,0b00011111); //file 2
ecrire_lcd(1,0b00011111); //file 3
ecrire_lcd(1,0b00001111); //file 4
ecrire_lcd(1,0b00000111); //file 5
ecrire_lcd(1,0b00000011); //file 6
ecrire_lcd(1,0b00000001); //file 7
ecrire_lcd(0,return_home1);
```

```
//coin inferieur droite

ecrire_lcd(0,0b01100000); //set cgram 96 / ecrire_lcd(1,4)
ecrire_lcd(1,0b00011111); //file 0
ecrire_lcd(1,0b00011111); //file 1
ecrire_lcd(1,0b00011111); //file 2
ecrire_lcd(1,0b00011111); //file 3
ecrire_lcd(1,0b00011110); //file 4
ecrire_lcd(1,0b00011100); //file 5
ecrire_lcd(1,0b00011000); //file 6
ecrire_lcd(1,0b00010000); //file 7
ecrire_lcd(0,return_home1);
}
#endif
```

F) Photos

1) Version 1.0



2)Version 2.0