

I. Cahier des charges

1. Contexte du projet

Le but de ce projet est de réaliser un robot autonome qui pourra suivre une trajectoire à l'aide de marqueurs au sol et de corriger sa trajectoire selon les obstacles qu'il rencontre ou encore le décalage qu'il a suivant les marqueurs. Ce robot, doit interagir avec l'homme qu'on appelle IHM (Interactions Homme-Machine) :

- Boutons Marche/Arrêt,
- Affichage des informations en temps réel (position par rapport aux marqueurs, vitesse, par exemple)
- Choix d'une fonctionnalité dans un menu

2. Description du matériel

Pour mener à bien ce projet, trois équipes vont travailler sur un sujet différent. Les trois sujets sont les suivant :

- Capteurs
- Moteurs
- IHM

Voici la liste du matériel pour chaque sujet :

Capteurs	Moteurs	IHM
Carte DsPic	Carte DsPic	Carte DsPic
Capteurs de position (photos diodes) IR TCRT 5000		3 boutons
Capteurs d'obstacles (à ultrasons) : SRF 05		Écran LCD : EA DOGM163W-A

Le sujet choisit par mon binôme et moi est celui de l'IHM. Jusqu'à présent, nous avons réalisé les câblages sur la plaque à essai, les premiers programmes et la réalisation de la carte schématique sur Eagle. Chacune des réalisations sont présentées dans les pages suivantes.

Dans un premier temps, voici le calendrier d'avancement de notre projet :

Date	Objectifs	Résultats	Membre
31/01	Câblage de l'écran LCD et allumage du rétroéclairage.	Réussite.	C / M ¹
02/02	Affichage d'un message sur l'écran LCD.	Réussite.	C/M
07/02	Branchement des boutons et résistances.	Branchement non convenable.	M
14/02	Programmation des boutons et affichage de leurs états (appuyé ou relâché) sur l'écran.	Réussite + Correction du branchement des Boutons.	C/M
21/02	Début sur Eagle : réalisation du schéma de la carte électronique.	Réussite sans capacités de découplage.	C/M
24/02 au 02/03	Réalisation d'un programme « Menu » permettant de choisir une action à l'aide des boutons. (Vacances)	Erreur de construction du projet.	C
05/03	Test du code « Menu » et correction.	Réussite.	C/M
06/03	Correction du schéma de la carte de la carte et début de réalisation de la carte.	Réussite + en cours de réalisation de la carte.	C/M

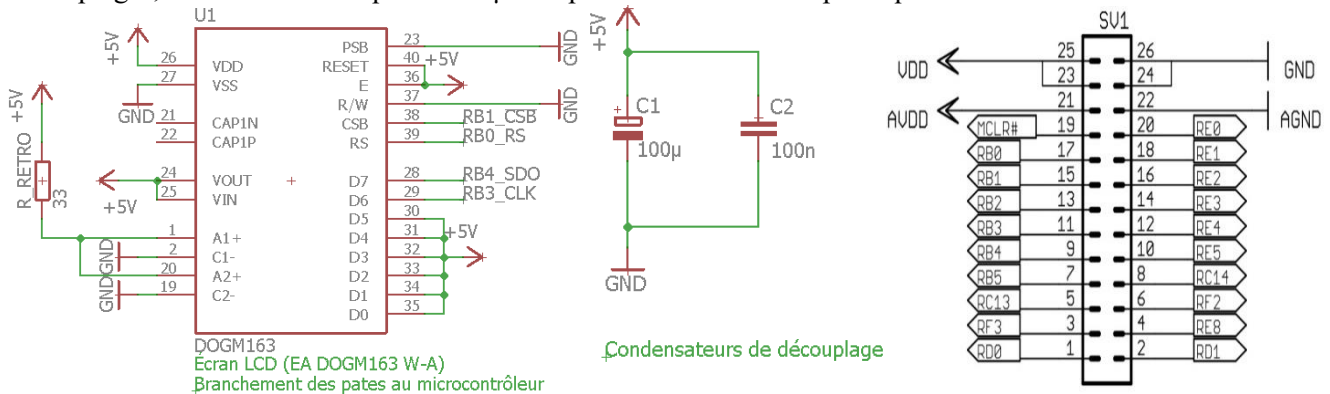
¹ C : Christopher / M : Mathéo

II. Câblage électronique et raccordement au Micro-contrôleur

1. Connexion Ecran/Micro-contrôleur (μC)

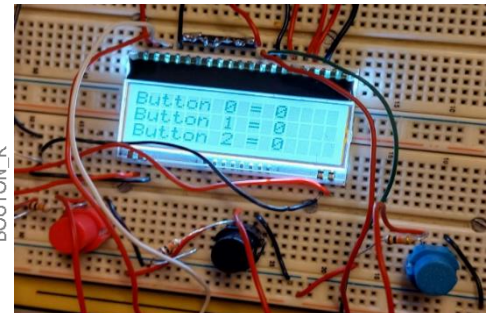
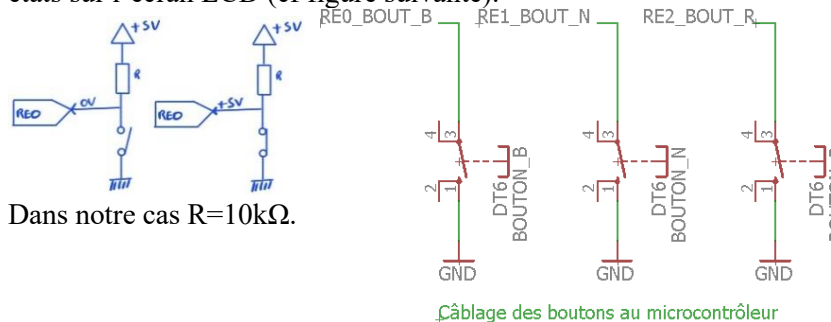
Pour ce projet, l'écran LCD utilisé est le EA DOGM163W-A. Il est composé de 40 pins mais seulement 22 sont utilisés.

On peut lire sur la figure ci-contre : \overline{CSB} , RS, SDO et CLK qui correspondent à des pins du μC . Ces pins ont été choisis vu leur fonction de contrôle d'affichage. Afin d'éviter tout problème avec le RESET, le pin 40 a été mis au +5V et non relié au RB2 du μC . Afin d'éviter un tirage trop important en courant, le +5V nécessaire à l'éclairage de l'écran a été pris par la plaque à essai. Après lecture de la datasheet de cet écran, on y lit qu'il faut placer entre les pins 1, 2, 3, 4 une résistance de 30Ω . De plus, pour avoir une entrée non bruitée, on a branché 2 capacités de découplages, une électrochimique de $100\mu F$ en parallèle avec une en plastique de $100nF$.



2. Connexion Boutons/Micro-contrôleur

Comme premier code avons afficher sur l'écran « Bonjour le monde ». Nous avons ensuite câbler les boutons sur les pins RE0, RE1, et RE2 du micro-contrôleur. Après câblage, nous avons coder les boutons afin d'afficher leurs états sur l'écran LCD (cf figure suivante).



Nous avons choisi les pins RE0, RE1 et RE2 vu que ses pattes ont une fonction d'interruption à des événements externes. Dans le câblage des boutons nous avons retiré les résistances vues la prestance de résistances (Pull-up resistors) dans le micro-contrôleur sur les pattes RE0, RE1 et RE2.

3. Connexion avec les Micro-contrôleur des capteurs et du moteur

Pour finalisé la réalisation du schéma de la carte il nous restait à ajouter des support de pins pour pouvoir brancher et connecté d'autre micro-contrôleurs avec la notre. Ceci a été réalisé à l'aide de 2 connecteurs avec lesquels on a connectés les pattes RC13(UA1TX) et RC14(U1ARX) de notre micro-contrôleur avec le micro-contrôleur des capteurs et celui du moteur. La pte RC13 va nous servir à envoyer des commandes au moteur et la pte RC14 à recevoir des informations des capteurs.



II. Programmation

1. Affichage du message d'allumage

Afin d'avoir une interface simple on a programmé l'affichage d'un message de bien venu qui dure 5 secondes afin de vérifier d'un premier temps que le programme fonctionne bien et d'informer l'utilisateur du démarrage de l'écran. Ce code, ci-dessous, situé dans la fonction main, va nous permettre d'afficher le message du démarrage du programme.

```
init_pins();
InitLCD();

int choix_menu=0;

LCDDisplayOn();
LCDContrastSet(contrast);
LCDGoto(0,0);
LCDWriteStr("XXXXXXXXXXXXXXXXXX");
LCDGoto(1,0);
LCDWriteStr("XX Demarrage! XX");
LCDGoto(2,0);
LCDWriteStr("XXXXXXXXXXXXXXXXXX");
delay_en_s(5);
```

2. Initialisation du menu

Nous avons créé une fonction menu que l'on a mis dans un fichier nommé « fonctions_sup.c ». Ensuite nous avons créé le fichier nommé « fonctions_sup.h » pour les prototypes des fonctions créés dans « fonctions_sup.c ». La première fonction que l'on a codée est la fonction 'delay_en_s(temps)'. Cette fonction nous permet de créer un délai en seconde pour pouvoir bien visualiser les messages sur l'écran. Ensuite on a créé la fonction « menu_principale() » qui va nous permettre de faire la sélection des différentes options dans le menu. Lors du déclenchement de la fonction « menu_principale() », un message s'affichera pour indiquer à l'utilisateur qu'il est dans le menu. Ensuite l'utilisateur aura la possibilité de sélectionner une des 2 options dans le menu en utilisant le bouton rouge pour monter, le bouton noir pour descendre et le bouton bleu pour confirmer la sélection.

```
#ifndef FONCTIONS_SUP_H
#define FONCTIONS_SUP_H

void menu_principale();
int detect_button_press(int button);
void loading_X_simulator(int *contrast);
void delay_en_s(int time);

#endif
```

Fichier « fonction_sup.h »

```
void delay_en_s(int time) {
    int loop;
    loop = time*1000;
    for (int i = 0; i < loop; i++) {
        _wait10mus(100);
    }
}
```

Fonction délai en secondes dans
« fonctions_sup.c »

```
int exit_menu=0;
int ptrcontrast = *contrast;
int choix=0;

int presser=100;

int operation1 = presser%2;

LCDClearDisplay();
LCDWriteStr("XXXXXXXXXXXXXXXXXX");
LCDGoto(1,0);
LCDWriteStr("XXXXXX MENU XXXX");
LCDGoto(2,0);
LCDWriteStr("XXXXXXXXXXXXXXXXXX");
delay_en_s(1);

LCDClearDisplay();
```

Code dans « menu_principale() »
qui affiche le message dans
« fonctions_sup.c »

```
do
{
    if (detect_button_press(2)&&detect_button_press(3)) {
        choix = 3;
        break;
    }
    if(detect_button_press(1))
    {
        count_bout_1 += 1;
    }
    if(detect_button_press(2))
    {
        LCDClearDisplay();
        presser = presser - 1;
    }
    if(detect_button_press(3))
    {
        LCDClearDisplay();
        presser += 1;
    }
    if((presser%2)==0)
    {
        LCDGoto(0,0);
        LCDWriteStr("1.Luminosite X");
        LCDGoto(1,0);
        LCDWriteStr("2.Test Bouton ");
        LCDGoto(2,0);
        LCDWriteStr("B1=OK/B2=-/B3+= ");
        choix = 1;
    }
    if((presser%2)==1)
    {
        LCDGoto(0,0);
        LCDWriteStr("1.Luminosite ");
        LCDGoto(1,0);
        LCDWriteStr("2.Test Bouton X");
        LCDGoto(2,0);
        LCDWriteStr("B1=OK/B2=-/B3+= ");
        choix = 2;
    }
}
}while(count_bout_1 < 1);
```

Code dans « menu_principale() » qui
nous permet de sélectionner une option
dans « fonctions_sup.c »

III. Conclusion

En conclusion, la réalisation de notre partie (IHM) est sur la bonne voie. Nous n'avons pas de retard dans notre planning. Actuellement, nous sommes en train de finaliser les dernières optimisations nécessaires dans la partie programmation, tandis que la carte est pratiquement achevée. De plus, le menu de l'IHM est en phase finale de développement. Nous sommes confiants que ces dernières étapes seront menées à bien dans les délais impartis. Nous sommes impatients de présenter un produit final de qualité, prêt à être intégré harmonieusement dans le projet global du robot suiveur de ligne.