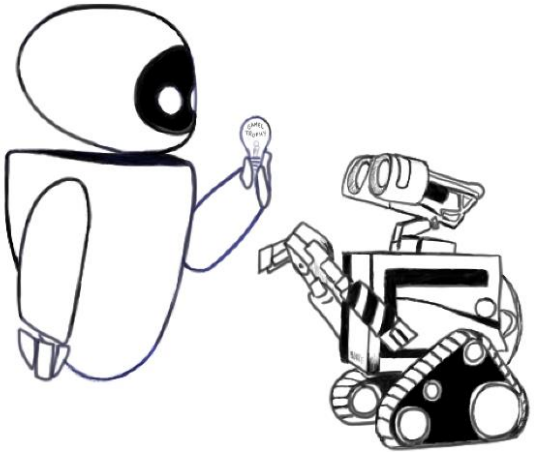
UNIVERSITÉ PARIS SUD

IUT DE CACHAN

université PARIS-SACLAY

Gamel Trophy



Rapport de projet

Robot suiveur de ligne WALL-E

Etude & Réalisation DUT GEII

2018 - 2019

Auteurs :

Lory-Stan Chamsoudine
Ghassen Lamloum
Bilal Marecar
Stéphane Yang

Dirigés par :

Nathalie Brissard
Remy Ghislain
Fabien Parrain

Sommaire

I. Présentation

1. Le « Gamel Trophy » (Ghassen, Stan)

Dans le cadre d'un projet universitaire au sein de l'IUT de Cachan, en filière GEII, une équipe de quatre personnes a été composée dans le but de réaliser un robot suiveur de ligne. Mise à part le châssis du robot qui est fourni, nous devons créer par nous-mêmes une carte puissance qui permet le contrôle des moteurs.

Nous devons également réaliser une carte capteur composée de plusieurs capteurs permettant la détection de la ligne blanche, et écrire un programme qui permettra la coordination de tout cela.

A la fin du premier semestre, un concours de rapidité et un concours d'esthétique aura lieu, le "Gamel Trophy", au cours duquel s'affronteront les robots réalisés par des étudiants de première année des deux départements (GEII 1 et GEII 2). Le robot doit être capable de suivre une piste le plus rapidement possible de façon autonome et de s'arrêter à la fin du parcours sans intervention humaine.

Figure 1 : Le robot

Figure 2 : La piste Gamel Trophy

2. Cahier des charges du projet (Bilal, Stéphane)

2.1 Aspects techniques

- Le robot doit être autonome en énergie.
- Support matériel imposé (châssis, roues, moteurs, batterie, coque, capteur fin de course, jack, carte électronique de commande)
- Le robot doit faire tomber la première barre située à la fin de la piste et de laisser en place une seconde barre, distante de 20 cm de la première.
- Le robot doit évoluer sans aucune aide extérieure.

2.2 Analyse fonctionnelle

La bête à corne désigne une méthode d'analyse fonctionnelle du besoin. Il est représenté sous forme de de fonctions simples que devra remplir le produit ou le service innovant.

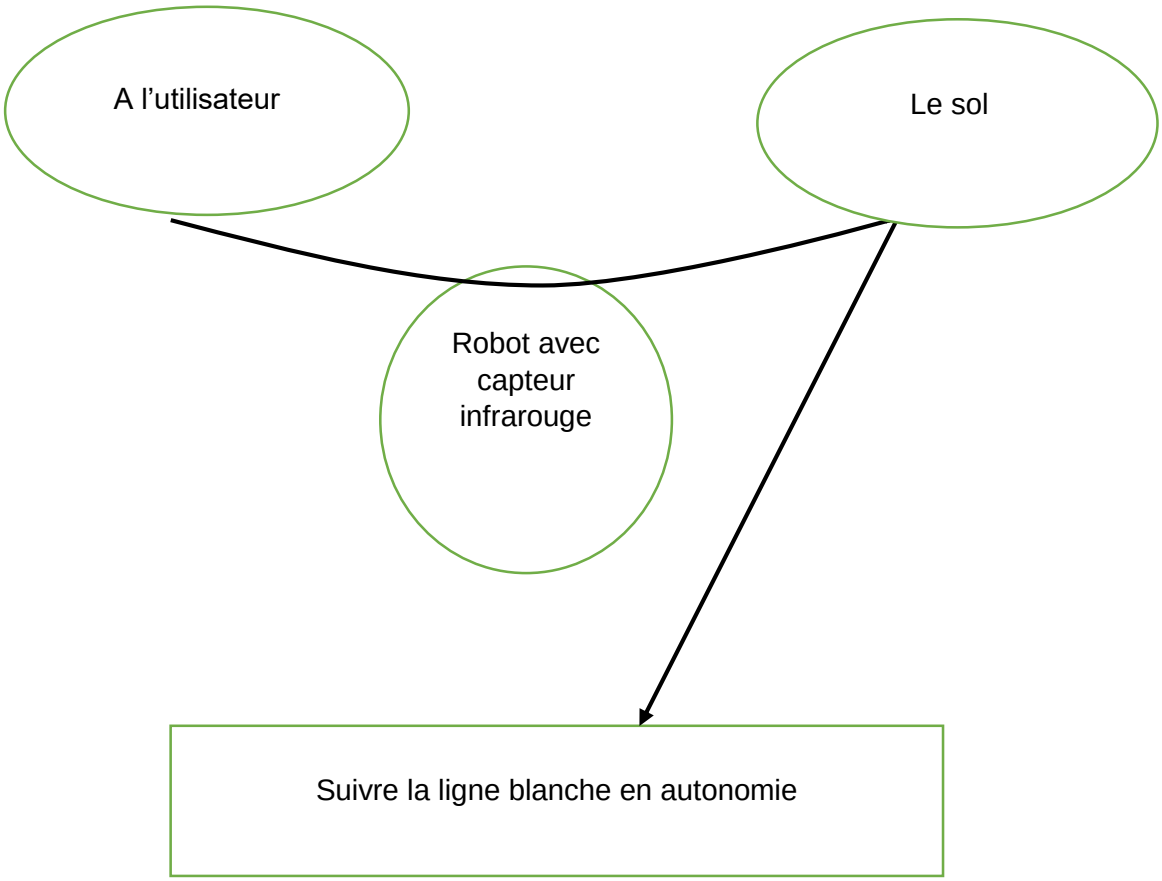


Figure 3 : Diagramme pieuvre

Le diagramme FAST présente une décomposition hiérarchisée des fonctions du système allant des fonctions de service et passant par les fonctions techniques jusqu'à l'énoncé des solutions technologique employées ou prévues pour remplir les fonctions techniques.

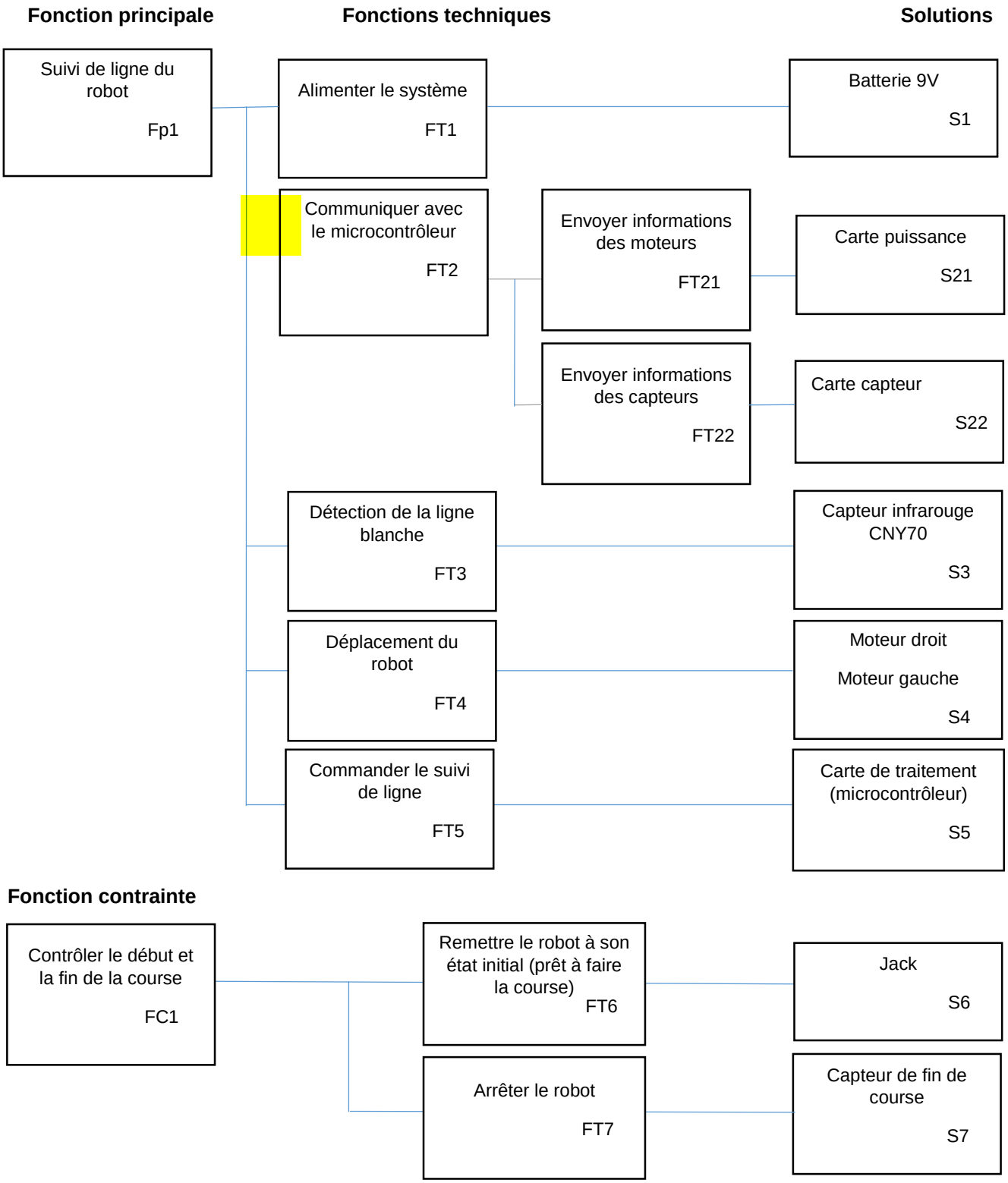
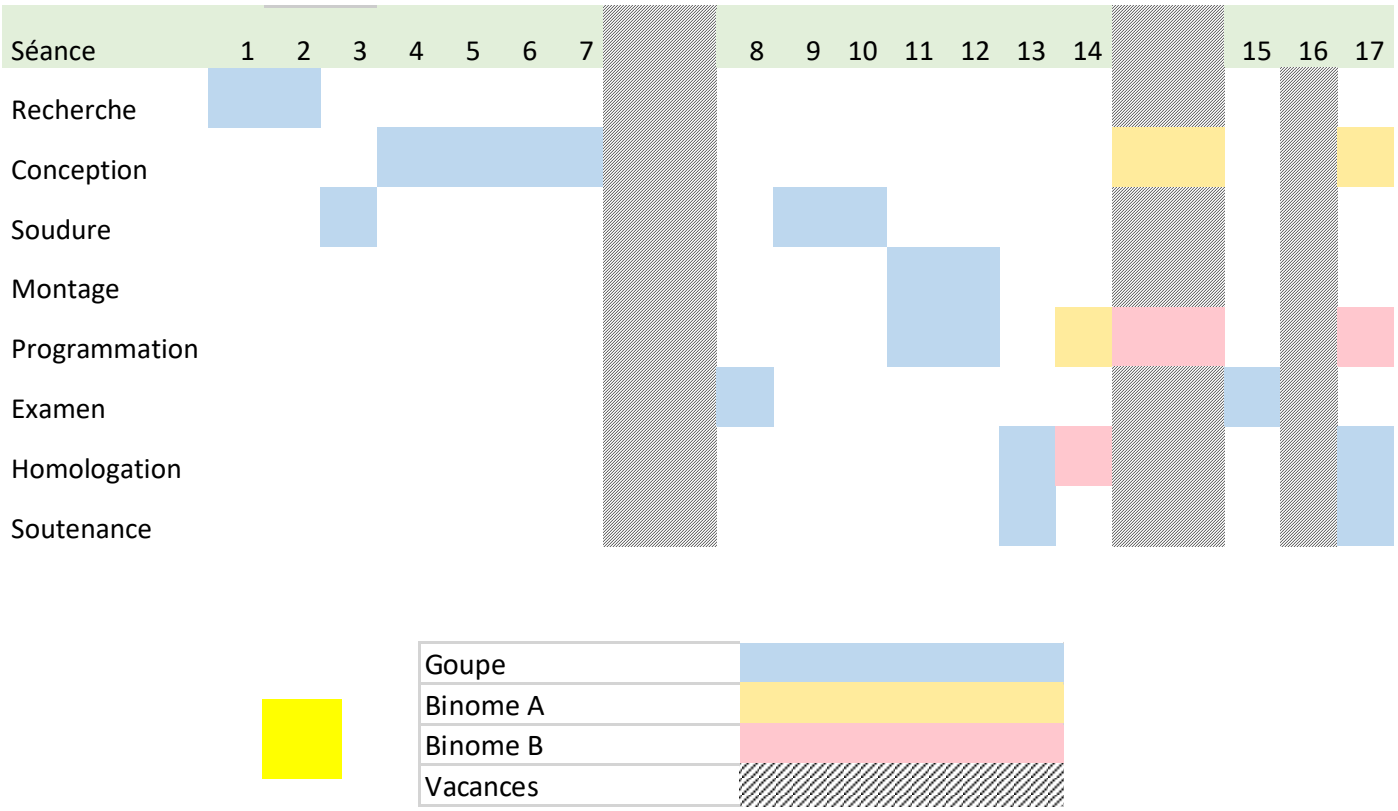


Figure 4 : Diagramme FAST

2.3 Planning du projet



Goupe	
Binome A	
Binome B	
Vacances	

I. Réalisation et test des cartes électroniques

1. Carte puissance (Stan)

Afin de fabriquer la carte puissance, nous avons utilisé le logiciel Altium Designer. Ce dernier est un logiciel permettant de réaliser des schémas électriques, les vérifier, les simuler et aller jusqu'à la conception du circuit imprimé.

Dans un premier temps, nous avons effectué une première esquisse, sur papier, des cartes celle-ci permettra de réaliser la partie schématique des cartes sur le logiciel. Puis, nous avons décidé des composants à utiliser. En effet, nous avons décidé des valeurs des résistances et des condensateurs, ou encore des régulateurs de tension, dans le but de répondre au cahier des charges. Voici la liste des composants dont nous avons eu besoin pour cette carte moteur :

- 1 Connecteur HE10 2*8
- 2 Transistor MOS à commande logique
- 2 Résistance 25Ω
- 2 Résistance 100kΩ
- 1 Condensateur PL 0,1 µF
- 1 Connecteur Autocom 2 broches mâles
- 1 Entrelec 4 broches mâles

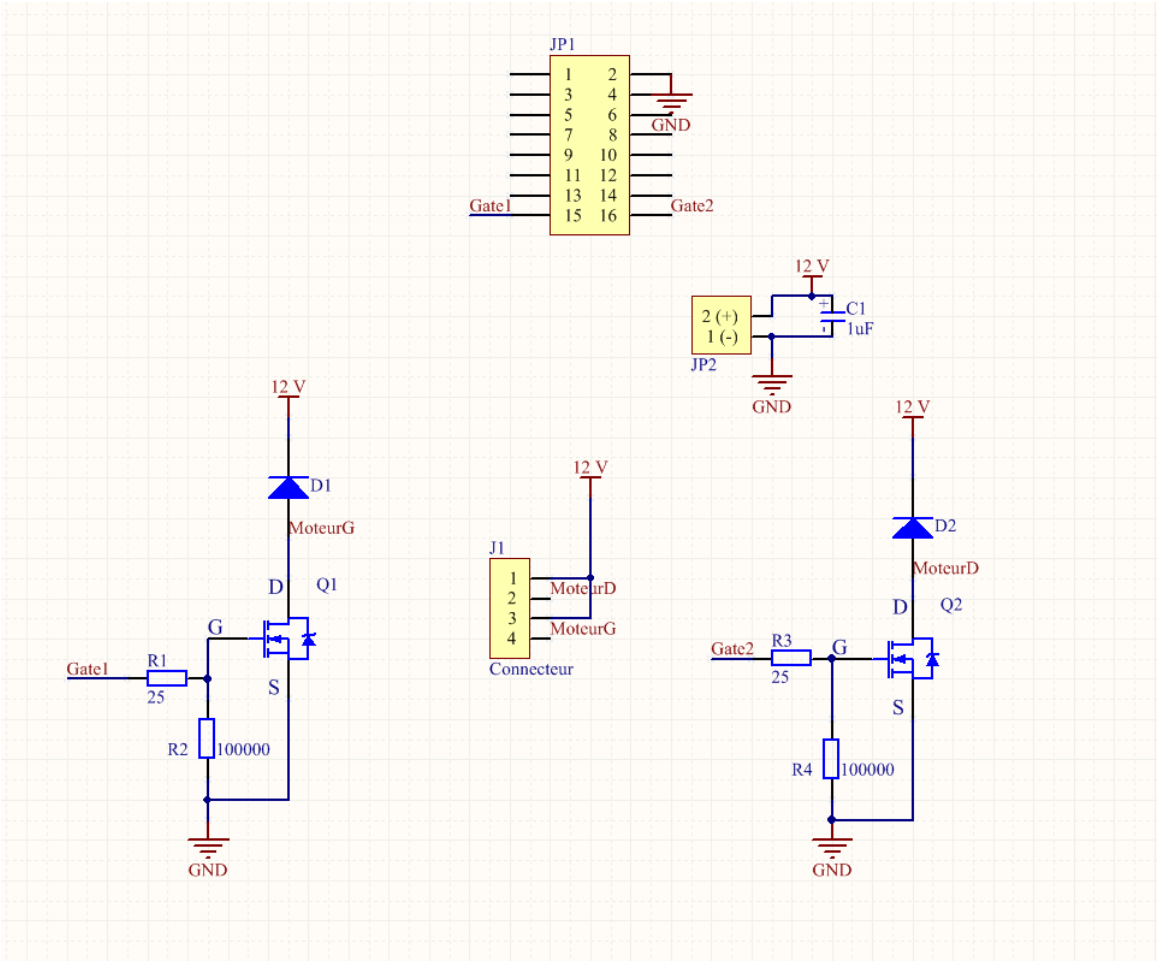


Figure 5 : Schéma carte puissance

Par la suite, nous avons exporté le schématique sur le PCB. Et nous l'avons routé, ce qui nous a permis de terminer la partie PCB qui représente la partie physique de la carte électronique.

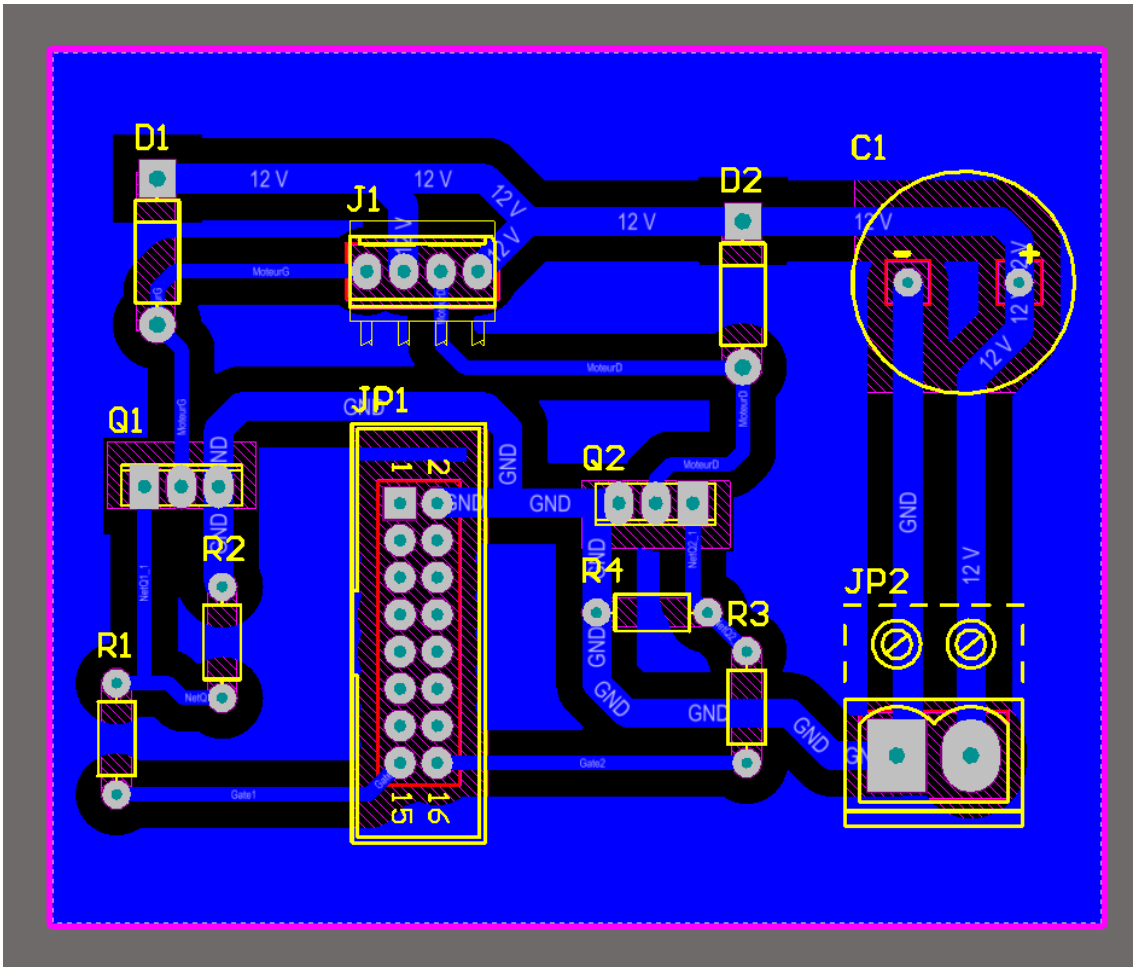


Figure 6 : PCB carte puissance

De plus, nous avons créé des fichiers gerber, étant donné que la machine qui fabrique les cartes ne peut pas lire directement notre PCB et donc ne peut pas directement les imprimés.

Enfin, pour finaliser la conception de la carte nous avons soudés aux deux cartes leurs différents composants cités précédemment. Cette étape nous a permis de découvrir la soudure dans son ensemble et apprendre à souder.

2. Carte capteur (Ghassen)

La carte capteur est un composant primordial de la Gamel puisqu'elle permet la détection de la piste blanche grâce à un ou plusieurs capteurs optiques réflectifs, les CNY70. Ce type de capteur est composé d'un émetteur et d'un récepteur optique mais aussi d'une LED infrarouge et d'un phototransistor. Le phototransistor est un senseur de lumière permettant de détecter la luminosité. Ces capteurs sont aussi constitués de 4 branches dont les deux premières sont l'anode et la cathode de l'émetteur et les deux autres sont le collecteur et l'émetteur du récepteur.

Le fonctionnement de ce dispositif se fait par réflexion, l'émetteur envoie un signal infrarouge sur un objet et si ce signal est renvoyé au récepteur alors il sera transmis au phototransistor. Le niveau de lumière renvoyé peut être mesuré en relevant la tension aux bornes de la résistance se situant au niveau de l'émetteur. Lorsqu'il fait noir, le phototransistor ne laisse presque pas passer de courant. Par conséquent, il n'y a presque pas de courant non plus dans la résistance qui lui est reliée (il s'en suit que la tension est aussi quasi nulle). Cependant, lorsqu'il fait clair il laisse passer plus de courant, qui circule donc dans la résistance et qui augmente la différence de potentiel aux bornes de celle-ci. La tension mesurée aux bornes de la résistance est donc plus grande quand il y a plus de lumière.

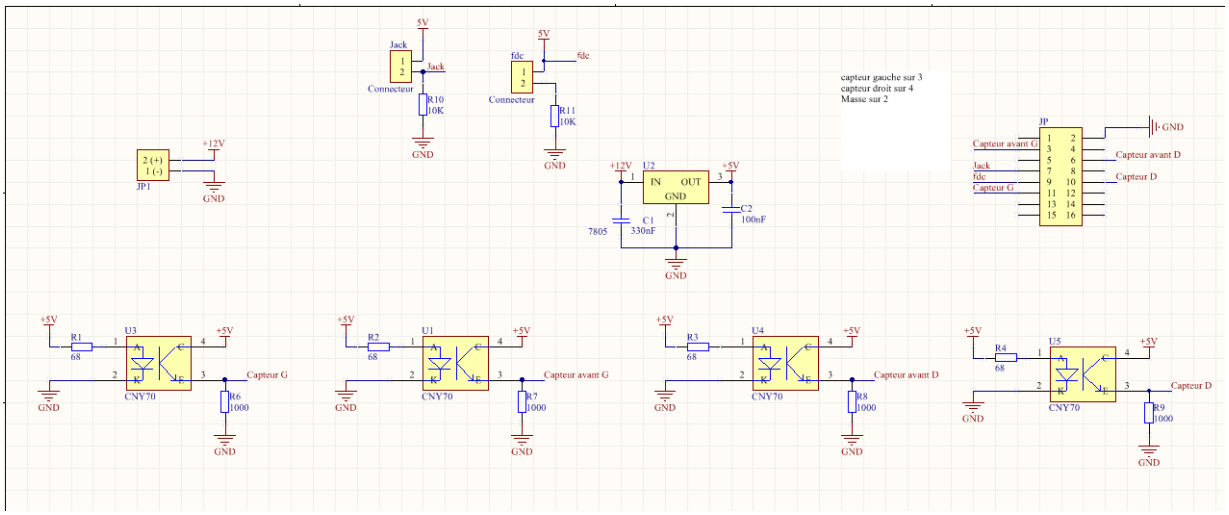


Figure 7 : Schéma carte capteur

L'utilisation d'une résistance adaptée permet d'éviter de « brûler » la LED infrarouge se trouvant dans l'émetteur. Le choix des résistances dépend à la fois de la tension d'alimentation et du niveau de luminosité maximum à détecter. En effet, sachant que nous avons une tension d'entrée de 5V, que la LED de l'émetteur possède une tension de 1.25V et que l'émetteur nécessite un courant de 50 mA nous appliquons la loi d'Ohm pour trouver la valeur de la résistance à utiliser.

Grâce à cette loi nous trouvons une valeur de 75 Ω, nous avons alors choisi une résistance de 68 Ω car c'est la valeur la plus proches de celle calculée. Pour la résistance du récepteur, nous avons choisi la valeur de la résistance en rapport avec le niveau de luminosité maximum du capteur, ici nous avons pris une résistance de 1 KΩ.

Comme expliqué dans la partie précédente, la carte capteur a été conçue de la même manière que la carte puissance, grâce au logiciel Altium Designer.

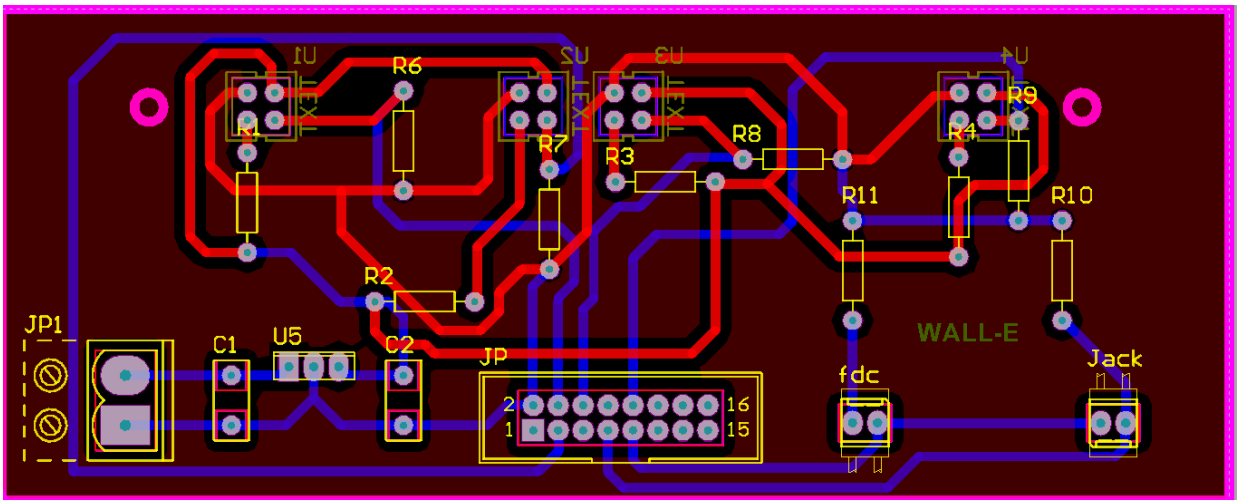


Figure 8 : PCB carte capteur

3. Test du fonctionnement des composants (Bilal, Stéphane)

Avant de monter le robot, nous allons tester le bon fonctionnement des moteurs. Nous allons les soumettre à une tension sinusoïdale (amplitude crête à crête de 5V et un duty cycle de 50%).Avec une fréquence de 10kHz, on perçoit que les moteurs tournent. Ensuite, plus la fréquence est grande plus les moteurs tournent vite.

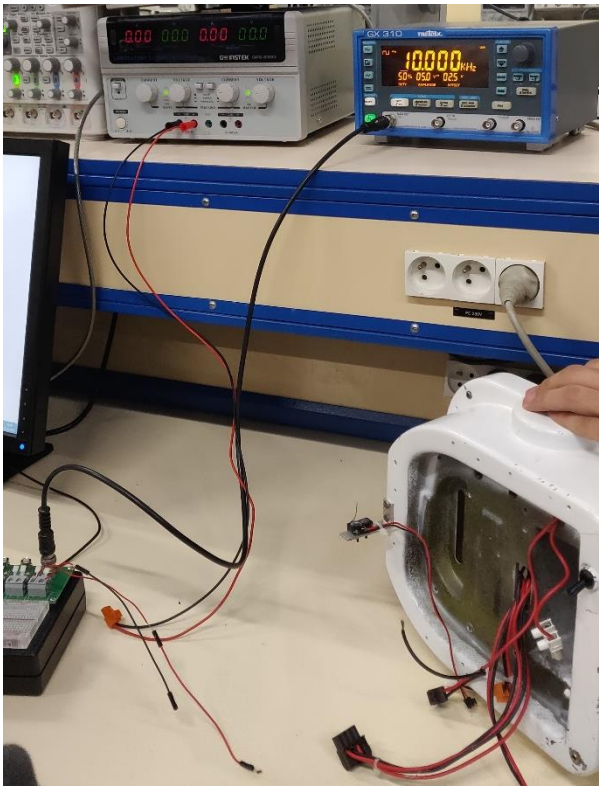


Figure 9 : Test des moteurs

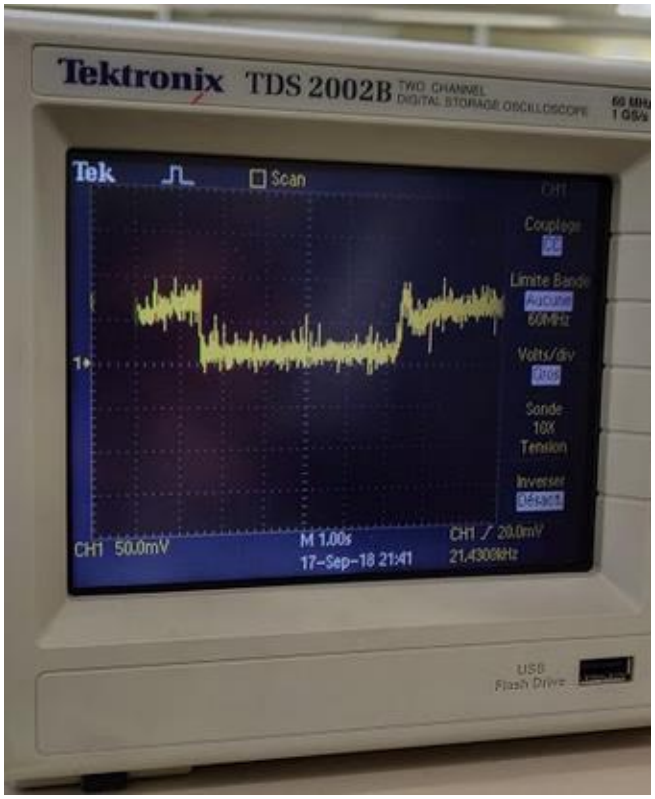


Figure 10 : Test des capteurs

Il est aussi nécessaire de prendre des mesures en temps réelle du capteur CNY 70 à l'oscilloscope pour vérifier les valeurs (si c'est en accord avec ce qui est dit précédemment).
On remarque que lorsqu'on passe une ligne blanche sous le capteur, ce dernier réagit et envoie une tension de plus ou moins 1.25V.
Plus on éloigne la ligne blanche, moins la tension est grande. On peut affirmer qu'à partir de 1 cm de distance, la tension est nulle.
0 volt = ligne NOIRE ou rien
1.25 volts = ligne BLANCHE

A travers ces tests, nous avons pu constater que nos moteurs tournaient dans le sens horaire et que deux nos capteurs semblaient ne plus fonctionner. Actuellement, nous essayons de résoudre ces problèmes en inversant le sens du branchement et en remplaçant nos deux capteurs défectueux.

II. Homologation du robot : suivi de ligne, polygone

1. Structure du programme du robot (Bilal)

Afin de programmer le robot, nous allons tous d'abord représenter une machine à état. Elle sera très utile par la suite pour programmer plus facilement notre système séquentiel.

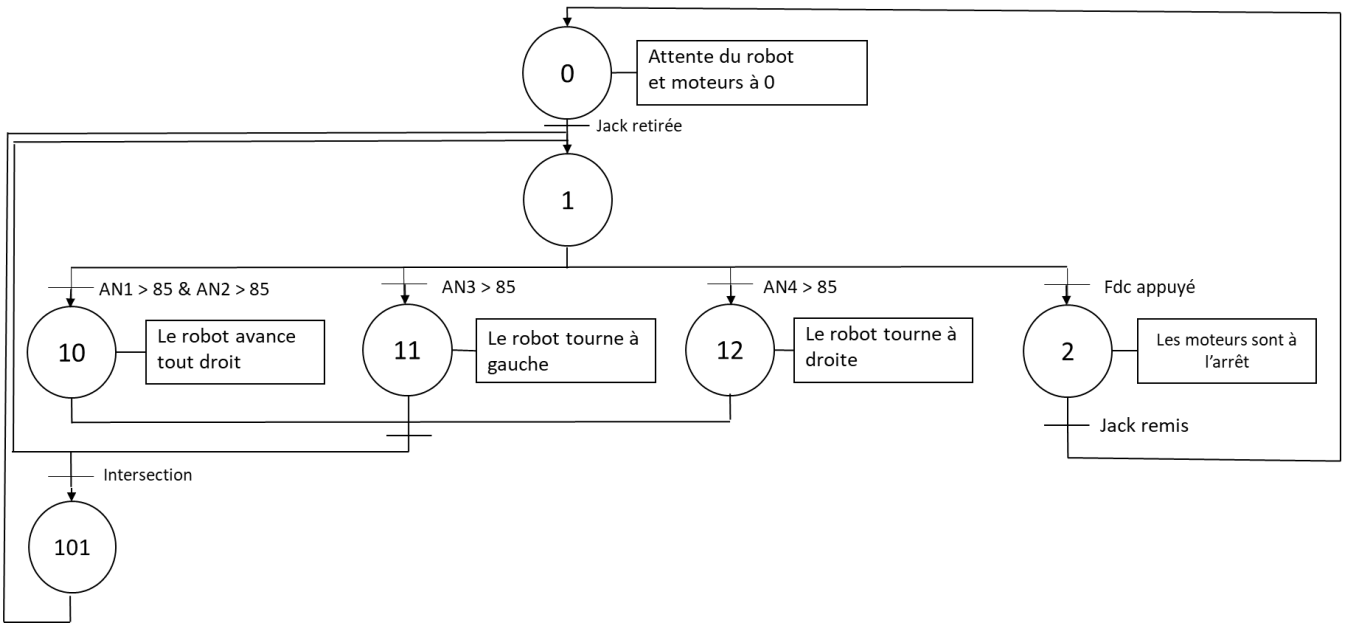


Figure 11 : Machine de Moore

Cependant, d'autres états vont s'ajouter au fur et à mesure de l'avancement du projet. Notamment pour la prise des raccourcis par le robot.

2. Programmation du suivi de ligne (Stéphane)

La programmation est ce qui va réellement assurer le suivi de ligne et donc constitue une part très importante du système.
Dans le cadre de ce projet, nous utiliserons le logiciel MPLAB qui servir à créer des programmes en langage C.

2.1 Déclaration des variables

Dans l'introduction des bibliothèques, il y a ceux présent originellement dans la création d'un nouveau programme et ceux appartenant à l'IUT. Ces dernières nous sont extrêmement utiles dans la mesure où cela va nous permettre prendre des raccourcis pour écrire le programme.

Pour la déclaration des variables, on distingue deux types (pour le projet),

```
#include <stdio.h>
#include <stdlib.h>
#include "iut_pwm.h"
#include "iut_adc.h"
#include "iut_lcd.h"
```

Figure 12 : Bibliothèque

La variable char qui concerne les valeurs allant -128 à 127,
La variable Int qui concerne les entiers relatifs.
Ces deux types permettent d'avoir des valeurs qui peuvent être modifiées au cours de l'exécution du programme, et c'est bien ce que l'on recherche à effectuer sur ces valeurs.
Les nombres d'états de la machine à état, de la prise jack et du capteur de fin de course n'étant pas très grands, il est plus pertinent de les inclure dans un char.

AN1, AN2, AN3, AN4 sont les noms que l'on a donnés aux 4 capteurs :
AN1→Capteur avant gauche
AN2→Capteur avant droit
AN3→Capteur gauche
AN4→Capteur droit

La vitesse maximale est définie à 200 bit des moteurs convertis par le CAN de la carte de traitement.

Le port B (concernant le jack et la fin de course) est déclaré avec tous les bits en entrées.
Les sorties PWMs des moteurs sont initialisés pour 2 compartiments (2 moteurs) avec une fréquence de 20kHz.

Le CAN (PortA) est initialisé pour les 8 premiers canaux pour les capteurs.

```
char etat=1;
char jack, fdc;
int AN1, AN2, AN3, AN4;
int vitesseMaximale=200;

TRISB = 0b11111111;

pwm_init(149, 2);
adc_init(7);
lcd_init();
```

Figure 13 : Déclaration des variables

L'écran LCD est initialisé (important sinon rien ne sera affiché sur l'écran).

2.2 Affectation des variables

On ouvre ensuite une boucle while qui répéter à l'infinie le programme :

```
while (1) {

    jack = PORTB & 0x02;
    fdc = PORTB & 0x04;

    AN1 = adc_read(2);
    AN2 = adc_read(3);
    AN3 = adc_read(6);
    AN4 = adc_read(4);

    lcd_position(0, 0);
    lcd_printf("%4d", AN1);
    lcd_position(0, 6);
    lcd_printf("%4d", AN2);
    lcd_position(1, 0);
    lcd_printf("%4d", AN3);
    lcd_position(1, 6);
    lcd_printf("%4d", AN4);
```

Figure 14 : Boucle while

La prise jack occupera le bit n°3 du port B.
Le capteur de fin de course occupera le bit n°5 du port B.

Le capteur AN1 occupera le bit n°3 du CAN.
Le capteur AN2 occupera le bit n°4 du CAN.
Le capteur AN3 occupera le bit n°7 du CAN.
Le capteur AN4 occupera le bit n°5 du CAN.

On affichera sur l'écran LCD les informations des quatre capteurs AN1(en haut à gauche), AN2(en haut à droite), AN3(en bas à gauche), AN4(en bas à droite). Cela sera utile pour suivre la gamelle à l'exécution du programme.

2.3 Mise en place du programme : La machine à état

On attaque enfin la partie la plus difficile du programme et du projet : la machine à état. Voici un aperçu de la fonction switch :

```
switch (etat) {
    case 0: // Etat 0: La gamelle attend une action
        pwm_setdc1(0);
        pwm_setdc2(0);
        if (jack !=0)
            etat = 1;
        break;
    case 1:
        if ((AN1>85 && AN2>85) || (AN3==85 && AN4==85))
            etat = 10;
        if (AN3>85)
            etat = 11;
        if (AN4>85)
            etat = 12;
        if (fdc ==0)
            etat = 2;
        break;
    case 10: // Etat 10: La gamelle avance tout droit
        if (AN1>85 && AN2>85 && AN3>85 && AN4>85)
        etat = 101;

        pwm_setdc1(vitesseMaximale);
        pwm_setdc2(vitesseMaximale);
        etat = 1;
        break;
    case 101: // Etat 101: La gamelle avance tout droite (lors d'une intersection)
        pwm_setdc1(vitesseMaximale);
        pwm_setdc2(vitesseMaximale);
        etat = 1;
    case 11: // Etat 11: La gamelle tourne à gauche
        pwm_setdc1(vitesseMaximale/2);
        pwm_setdc2(vitesseMaximale);
        etat = 1;
        break;
    case 12: // Etat 12: La gamelle tourne à droite
        pwm_setdc1(vitesseMaximale);
        pwm_setdc2(vitesseMaximale/2);
        etat = 1;
        break;
    case 2:
        if (jack ==0)
            etat = 0;
        break;
    default:
        etat = 0;
```

Figure 15 : Fonction switch

3. Bonus : programmation des raccourcis et du polygone (Stan, Ghassen)

En fonction du temps restant, on prévoit la prise de raccourcis. Nous remarquons que raccourcis sont définis par une petite ligne blanche qui se trouve à l'entrée du raccourci. Cette petite ligne blanche est situé d'un côté ou de l'autre de la piste, et c'est de ce côté que doit partir le robot pour emprunter le raccourci.

A l'aide du logiciel MPLAB, nous devons écrire une partie de programme qui permettra au robot d'aller d'un côté ou de l'autre de la piste lorsque les capteurs perçoivent une ligne blanche situé que d'un seul côté. Ainsi nous devons donc créer un nouvel état, qui se trouvera dans la machine à état déjà écrite, où celui-ci sera la prise en compte des raccourcis.

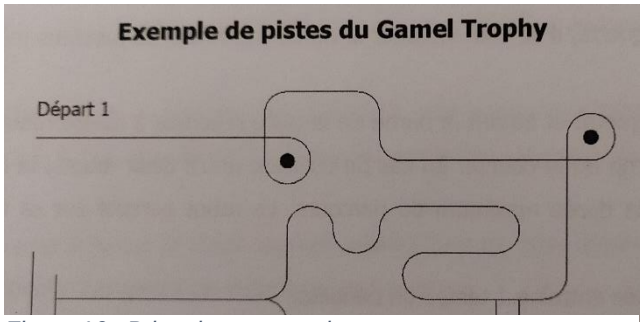


Figure 16 : Prise des raccourcis

III. Design du robot « WALL-E » (Stéphane)

La modélisation est réalisée sous le logiciel SolidWorks. Elle consiste en une sorte de “masque” qu’on met par-dessus la Gamel originale.

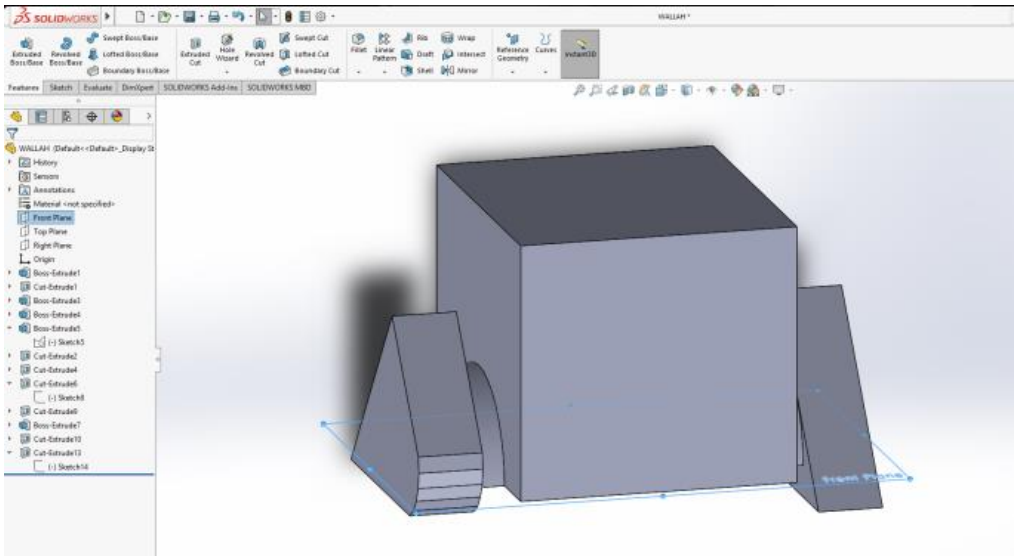


Figure 17 : Modélisation

Les principaux outils utilisés sont polygone, cercle, cotation intelligente, rectangle et trait.
Les principaux outils utilisés sont bossage extrudé, enlèvement de matière extrudé, congé pour les fonctions.
WALL-E est encore en cours de construction...

Figure 18 : WALL-E

IV. Conclusion (Stéphane)

Ce projet de première année a été un véritable défi pour nous mai aussi construit des bases solides en génie électrique. En effet ce projet demandait des notions à la fois d'électronique, d'informatique, et de mécanique, nous avons donc dû faire appel aux expériences du lycée et faire preuve d'efficacité dans l'acquisition des nouvelles compétences. Le projet nous a également permis de découvrir de nouveaux outils tel que Altium Designer que nous n'avions pas encore eu à exploiter. A noter que c'est la première fois que l'on doit réaliser par nous-même des cartes électroniques.
Nous avons été confronté à un planning très serré au regard de la charge de travail, mais c'est ce qui permet justement de se rapprocher d'un cadre professionnel.

Enfin, après plusieurs échecs notamment au niveau matériel (soudure, mesures), on a réussi à obtenir un résultat plutôt satisfaisant. Cela nous fait comprendre comment la cohésion d'équipe est importante et la plus sage utilisation du temps constituent des points essentiels à une expérience à la fois motivante et enrichissante.

V. Bibliographies

Carnet de projet Etudes & Réalisation 2018 - 2019
Cours Informatique Industrielle IA1
Cours Informatique Industrielle IA2
Fiches références sous Altium Designer
Mise en page par Bilal Marecar
Dessin page de garde réalisé par Stéphane Yang