

RAPPORT PROJET ROBOT 2

|  |
| --- |
|  |
| Seconde partie – Programmes embarqués | |
| 22 juin  GROUPE III  LORRIAUX Tristan  MAISONNAVE Lucas |

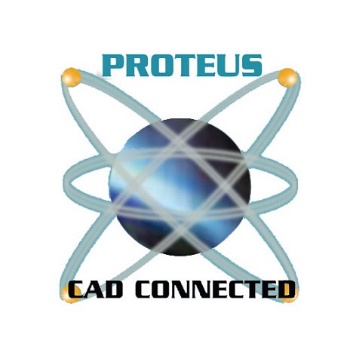


Table des matières

[Introduction 3](#_Toc43762059)

[I°) Le contrat 7 5](#_Toc43762060)

[II°) La synoptique du matériel utilisé 6](#_Toc43762061)

[III°) Algorigrammes 7](#_Toc43762062)

[IV°) Code & Simulation 11](#_Toc43762063)

[V°) Perfectionnements apportés et pistes d’améliorations 14](#_Toc43762064)

[Conclusion 15](#_Toc43762065)

[Annexes 16](#_Toc43762066)

### Introduction

Lors de leur cursus, les étudiants de l’école des Mines de St Etienne cycle ISMIN ont l’occasion de concevoir et prototyper un projet de soudure, de simulation et de test de la carte électronique d’un robot appelé projet Robot. Ce second rapport rend compte des avancements sur la partie *Software* de la carte.

Le robot réalisé est un robot suiveur. Il doit donc voir, être capable de se mouvoir, ou encore de rendre compte de son état. Nous employons des solutions parmi d’autres pour répondre à ces besoins.

La seconde partie du projet a donc pour but de programmer la carte électronique maintenant soudée et de réaliser les tests nécessaires à l’accomplissement d’un contrat qui nous a été attribué. La difficulté de celui-ci contrat dépend du niveau présupposé du binôme.

Notre contrat, le numéro 7 va déjà assez loin dans ce que peut désormais faire la carte soudée. Néanmoins, nous n’irons pas au-delà des fonctions primaires, qui seront quant à elles améliorées (ici pas de Sonar par exemple).

L’étude est passée par la réalisation d’algorigrammes, la programmation de ceux-ci en C embarqué puis la simulation. En raison des récentes conditions sanitaires, l’intégration sur carte électronique n’a pu avoir lieu.

*Ce présent rapport résume l’étude, la programmation ainsi que les tests et la simulation des fonctions de notre contrat pour notre robot et présentes sur la carte électronique.*

*Pour réaliser les simulations, nous avons utilisé le module ISIS du logiciel Proteus. Pour ce qui est de la programmation, elle a été effectuée sur l’IDE de Microchip© .*

### I°) Le contrat 7

Pour ce contrat, il s’agit de détecter un signal à l’aide des capteurs infra-rouge actifs montés à l'avant du robot. Le robot pourra dès lors se diriger vers une cible présente à l'avant de celui-ci. Ainsi , le robot démarre et avance en ligne droite s'il détecte un « obstacle fixe » à moins de 1,50 m et s'arrête si la distance est inférieure à 40 cm. La vitesse d'approche de la cible est fixée à 20% de la vitesse maximale (environ 12 mètres par minute). Par ailleurs les acquisitions des signaux des capteurs se feront toutes les 20ms. La fréquence d’horloge du robot est imposée à 8MHz.

Le contrat peut se diviser en 3 tâches à réaliser :

* Détecter un objet entre 40 cm et 1 m 50
* Avancer dès que la détection est effective
* Avancer d’un mètre et s’arrêter

Au-delà du contrat d’autres tâches devront être réalisées :

* Surveiller la batterie en moyennant sur 4 mesures régulièrement
* Afficher des informations sur les leds (ledtest et pcf)
* Gérer la liaison série RS232 pour le debug
* Optimiser le programme pour limiter la RAM utilisée

### II°) La synoptique du matériel utilisé

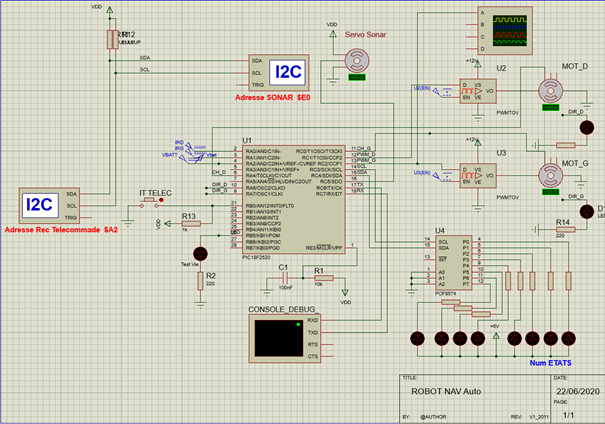
Faisons un bref inventaire des composants présents sur la carte qui nous seront utiles :

- Le PIC18 : microcontrôleur qui exécuter le code que l’on a implémenté.

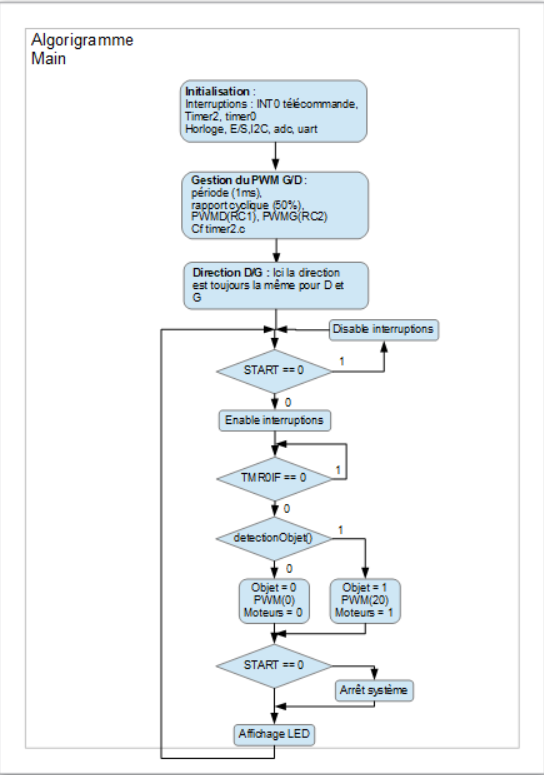
- Les 2 moteurs, droit et gauche, à courant continu pilotés en PWM par le µC.

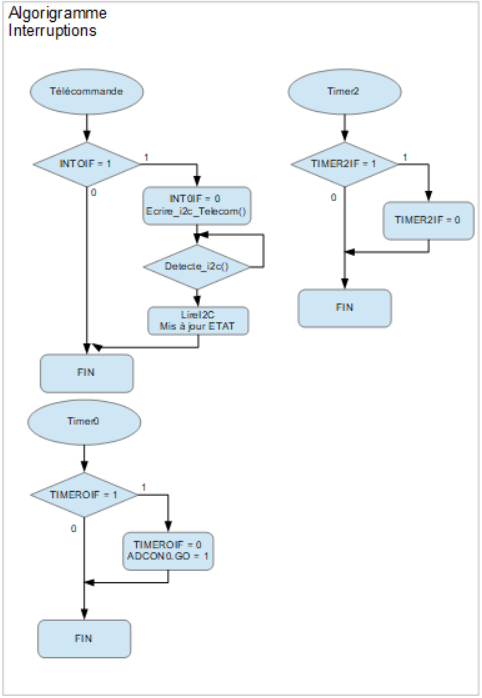
- U4 qui va convertir l’entrée I2C en sortie binaire à afficher sur les LEDs.

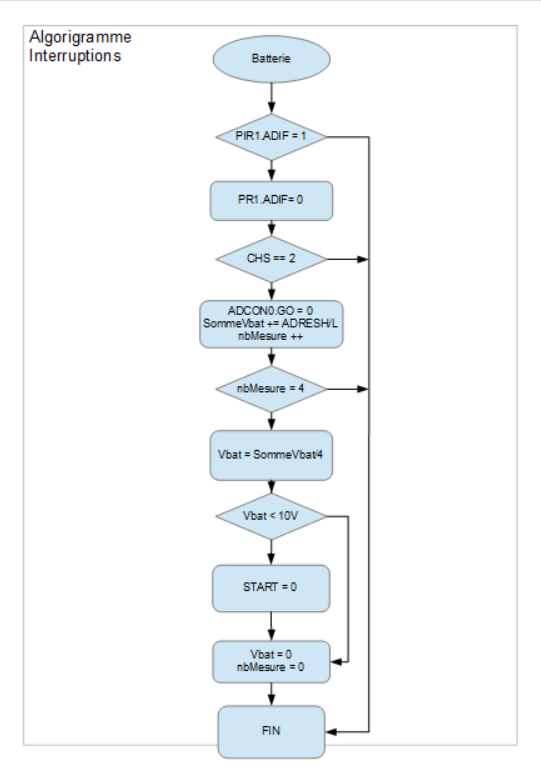
- Les entrées : Vbat qui représente la tension batterie, IRD et IRG qui les tensions en sortie des capteurs IR. IT\_TELEC est l’interruption créée par la télécommande. Et les entrées I2C pour communiquer avec la télécommande et l’affichage LED.

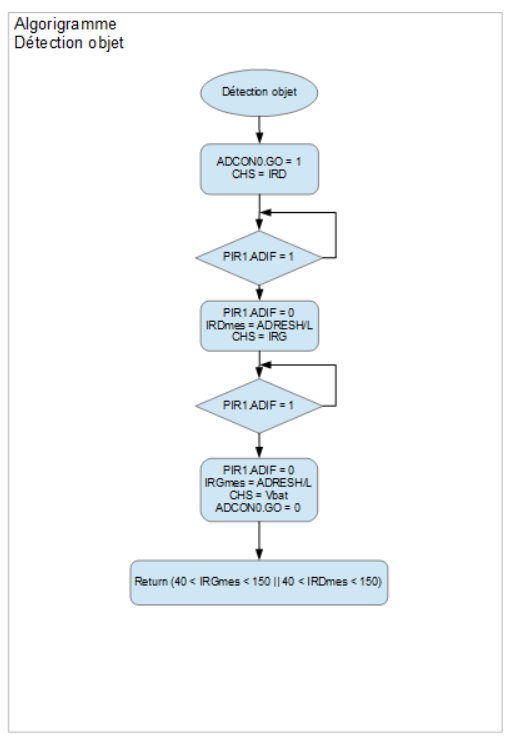
- Les sorties : DIR\_D et DIR\_G qui sont les valeurs digitales de direction des moteurs qui seront constantes, et PWM\_D et PWM\_G qui sont les PWM des moteurs.

### III°) Algorigrammes







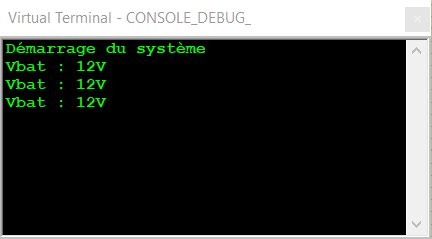


### IV°) Code & Simulation

*Nous mettons à disposition l’ensemble du code disponible en annexe, ainsi que sur GitHub, en suivant le lien :*

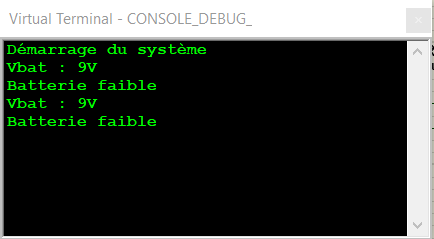
<https://github.com/lucasmaisonnave/ProjetRobot2>

1. Initialisation, Uart et Test Batterie



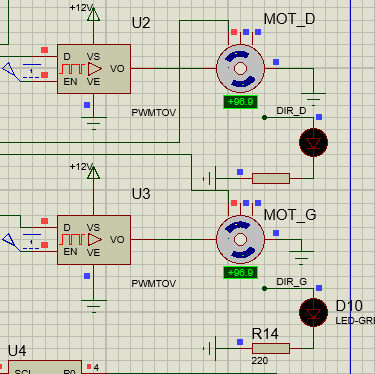
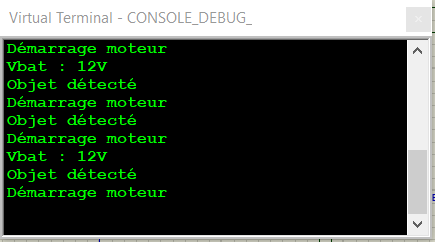
Les premières fonctions programmées sont contenues dans les fichiers d’initialisation (init.c notamment). Basiquement, ces fonctions permettent d’activer les premiers modules du pic18 (Configuration Horloge, Initialisation des entrées/sorties, de l’I2C, de Uart, de l’ADC et des interruptions). Le premier test que nous réalisons donc sous Proteus demeure la vérification de ces protocoles d’initialisation : c’est un succès.

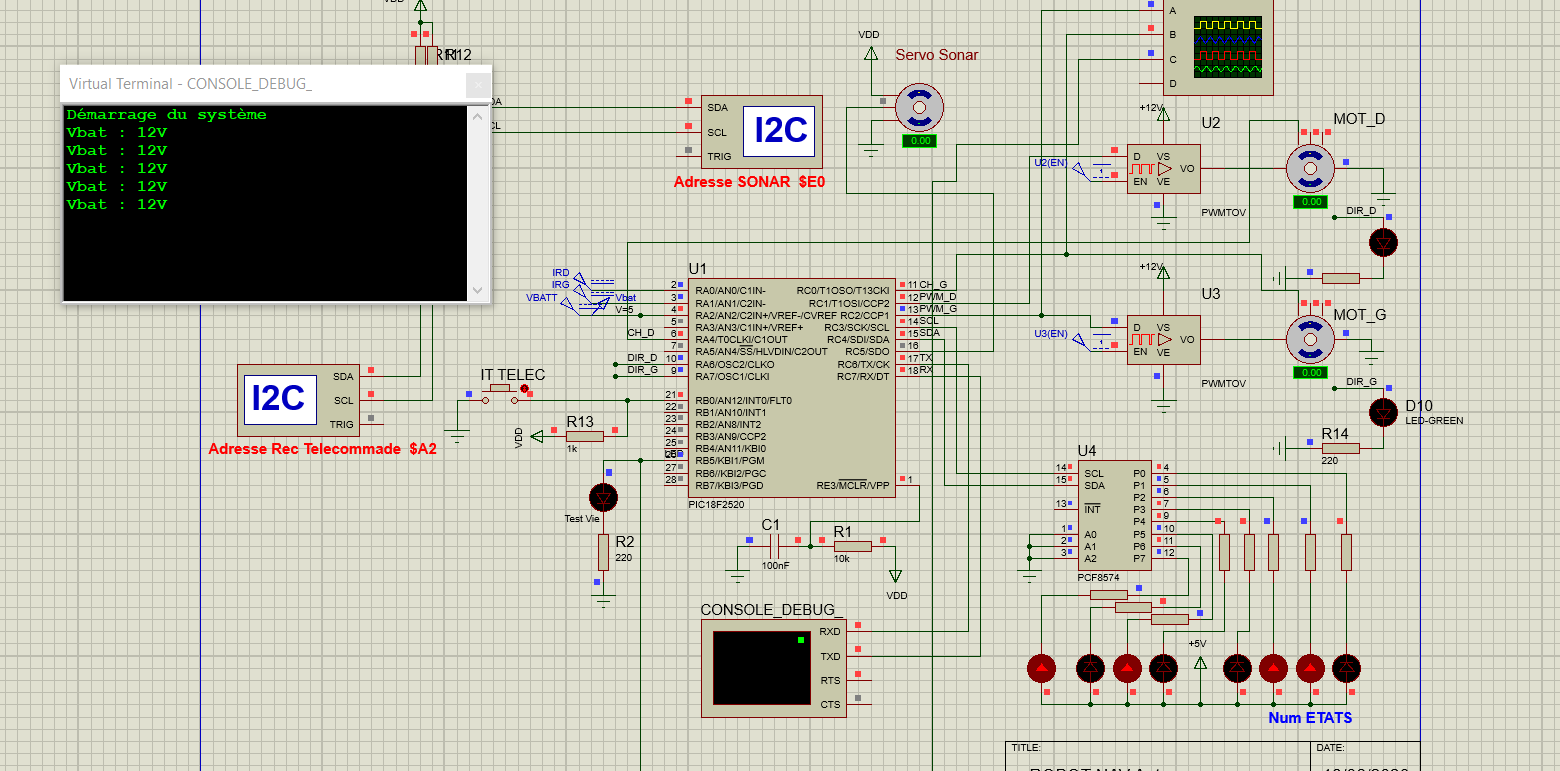
Après l’appui manuel sur l’interruption de télécommande, on lit l’information transmise par la télécommande via un bus I2C et qui envoie l’information d’un appui sur le bouton central : le robot se lance. La LED test s’allume ; pour le programme elle témoigne que le robot est lancé et non en pause ou en arrêt forcé.

Par ailleurs, nous vérifions aussi l’initialisation du processus de surveillance de Vbat et le bon fonctionnement de l’Uart par ce test.

Ensuite, nous vérifions que la routine de surveillance batterie fonctionne bien. Pour expliquer rapidement son fonctionnement : l’Uart affiche Vbat moyenné par 4 mesures tous les quarts de secondes.

1. Capteurs, moteurs et PWM

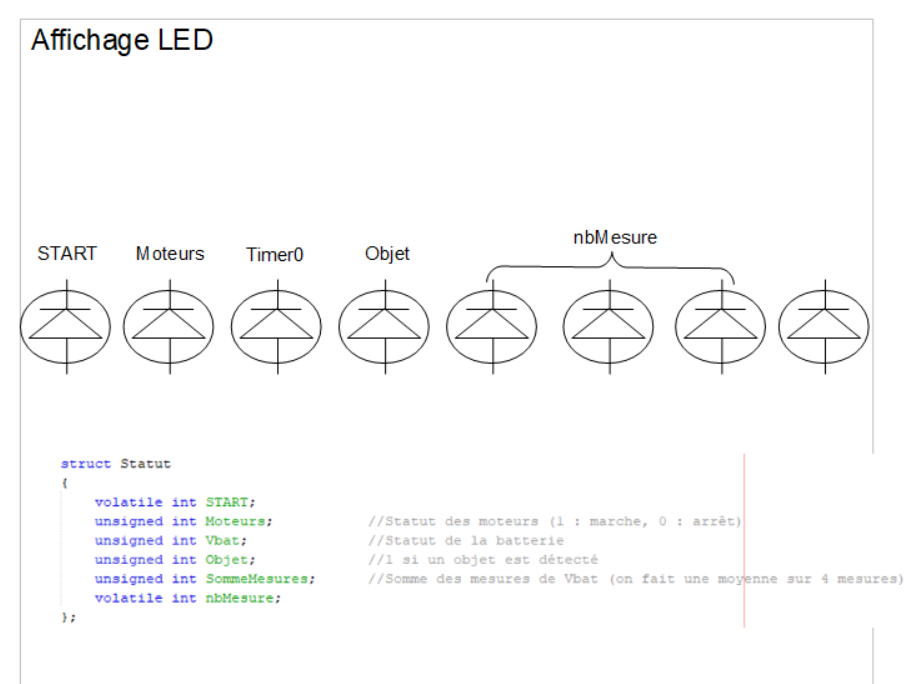
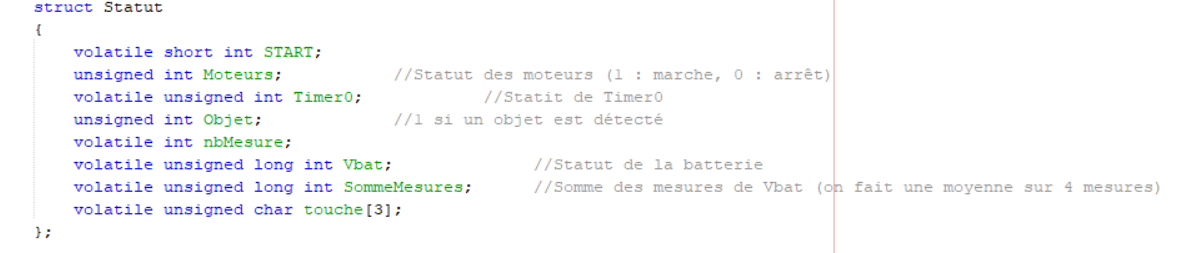
Le second test porte sur les moteurs. Aucun souci n’est noté durant la phase de test. Les moteurs tournent à 97 tours/min, l’acquisition relative au capteurs semble donc fonctionner. Soulignons que leur *duty cycle* se situe aux environs de 20% lorsque le robot avance. La détection a lieu tous les quarts de seconde pour économiser la batterie.

Nous réalisons donc un second test : on appuie sur start, la batterie est valide, le système est actif mais pas d'objet détecté donc le moteur est à l’arrêt. Tout semble donc fonctionner.

1. Affichage

Nous utilisons la technologie I2C pour contrôler les LEDs, ce qui se fait via MI2C.h/MI2C.c .

Les LEDS suivent ce plan dans notre projet (une LED reste inutilisée, on peut imaginer l’utiliser par exemple pour le sonar) :

Le test LEDs semble donc fonctionner : que ce soit au niveau de la détection objet, du START, du démarrage moteurs, du timer ou encore du nombre de mesure tout est en ordre.

### V°) Perfectionnements apportés et pistes d’améliorations

Le contrat a été amélioré sur plusieurs points.

En premier lieu, la divisions des mesures effectuées de Vbat : l’Uart affiche une Vbat moyenné par 4 mesures toutes les quarts de seconde, ce qui permet une économie notoire de l’énergie et donc une meilleur durée de vie de la batterie.

Les premières améliorations que nous pouvons imaginer sont évidemment liées aux contrats supérieurs : acquisition de la rotation des chenilles, ajout d’un sonar (y compris son acquisition), pilotage par la télécommande, etc…

Mais nous pouvons aussi imaginer :

Commander le robot avec le portable, changer d’un mode suiveur en un mode détection d’obstacles par pressage sur le bouton START, …

### Conclusion

Malgré les conditions sanitaires qui ont compliqué la réalisation de ce projet, nous avons mené à bien la réalisation de ce robot de A à Z. Certes, nous n’avons pas pu voir le fruit de notre travail du fait que nous n’avons pas pu intégrer notre code. Mais nous avons pris beaucoup de plaisir durant cette première année à souder ou à programmer sur ce projet robot, qui nous a notamment permis de mettre en pratique des connaissances théoriques apprises dans le reste de notre cursus

L’architecture du Pic18 nous semble nettement moins ténébreuse aujourd’hui et nous ressortons de ce projet avec la volonté d’en apprendre plus sur d’autres microcontrôleurs.

### Annexes

*Nous n’avons pas inclus MI2C.c/.h et p182520.h qui sont des librairies importées et que nous n’avons pas écrites. Cependant, elles sont nécessaires à la bonne exécution du code.*

