

RAPPORT PROJET ROBOT

|  |
| --- |
|  |
| Première partie – Assemblage de la carte | |
| 16 décembre  GROUPE III  LORRIAUX Tristan  MAISONNAVE Lucas |



Table des matières

1. Introduction
2. Alimentation
3. Capteurs
4. Interface Homme/Machine
5. Pilotage rotation roues-chenilles
6. Protocole communication série I2C
7. Conclusion

1. 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 premier rapport rend compte des avancements sur la partie *Hardware* 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.

Le déplacement aura lieu à l’aide de chenilles. Il interagira avec le monde grâce à un sonar, des capteurs infrarouges et une télécommande. L’utilisateur sera informé des différents états du robot avec les LEDs, et chaque fonction sera alimenté par des batteries d’accumulateurs.

Notons que pour faciliter l’examen, la simulation et l’exécution de la carte, nous avons sectionné celle-ci en blocs fonctionnels tels que l’alimentation, les capteurs, la rotation sonar, la surveillance de l’alimentation, la communication liaison série ou encore la rotation chenilles.

Moteurs

Carte électronique

Communication interne (Bus I2C …)

Alimentation

Ordinateur

Affichage LEDs

Témoin batterie

Mémorisation (facultative)

Réception

Capteurs IR

Sonar

Télécommande

Rotation chenilles

*Ce présent rapport résume la simulation, le montage, la programmation ainsi que les tests de chacune des fonctions utiles pour notre robot et présentes sur la carte électronique.*

*Le câblage et les simulations de ces fonctions ont été réalisés conjointement pour chacune de ces fonctions. Des tests ont été réalisés régulièrement afin de s’assurer du fonctionnement de celles-ci.*

*Pour réaliser ces simulations, nous avons utilisé le module ISIS du logiciel Proteus.*

1. Alimentation

*Ce bloc est à la base de toute l’alimentation de la carte, adapte la tension de la batterie et s’assure que en cas de déchargement de la batterie, le système nous informe de celle-ci.*

1. Régulateur de tension

Le bloc inclut un fusible, une diode D9 permettant d’éviter une inversion de polarité, un transistor U5 (régulateur de tension) et des capacités afin de stabiliser la tension à 5𝑉. La LED D10 sert de témoin d’alimentation. Par suite de nos tests, nous relevons une tension de sortie VDD stable de 4,95𝑉 et observons l’allumage du témoin d’alimentation.

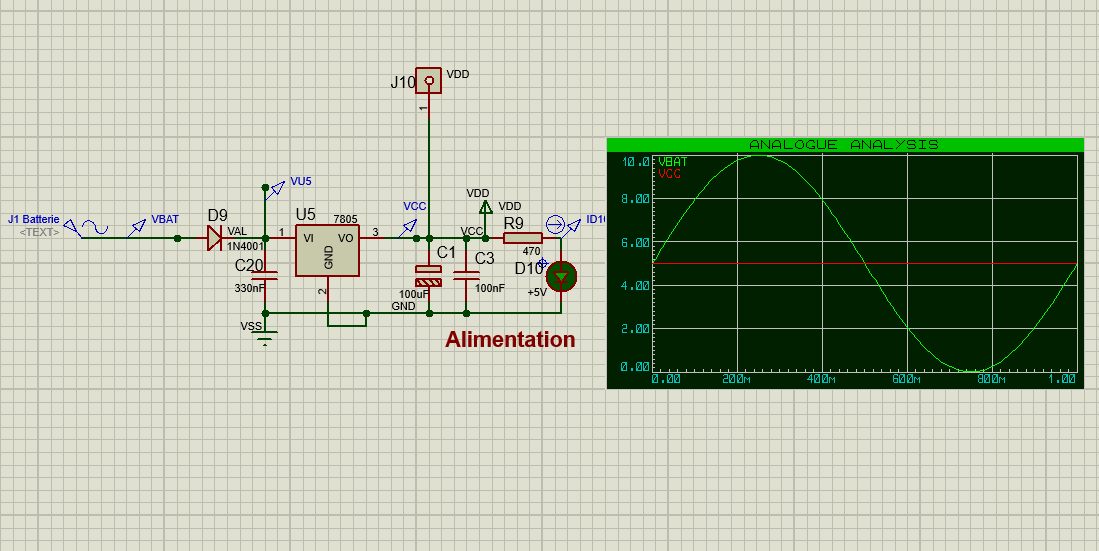


Figure 1 - Bloc alimentation : Visualisation UBat et USortie

1. Surveillance de l’alimentation

Cependant, la batterie alimentant le robot se décharge au cours du temps. Si la tension à ses bornes passe en dessous d’un certain seuil, elle peut ne plus être suffisante pour alimenter la carte.

Il faut donc mettre en place une surveillance de l’alimentation. Le schéma ci-contre permet d’acquérir une tension image de la tension réelle aux bornes de la batterie qui est ensuite transmise au microcontrôleur. La résistance sert de pont diviseur de tension afin d’éviter de dépasser la plage de tension acceptable par le microcontrôleur (0𝑉 − 5𝑉). Cette surveillance de l’alimentation est effectuée via le bloc surveillance batterie, qui renvoie au microcontrôleur U1 une tension image de la tension batterie (il faut rajouter les 0.6 V due à la chute de tension entraînée par la diode empêchant une inversion de polarité D9, et appliquer le rapport du pont diviseur de tension : on obtient alors une loi affine). Celle-ci est alors divisé par 20 mV à la sortie de U1.

On a donc finalement :

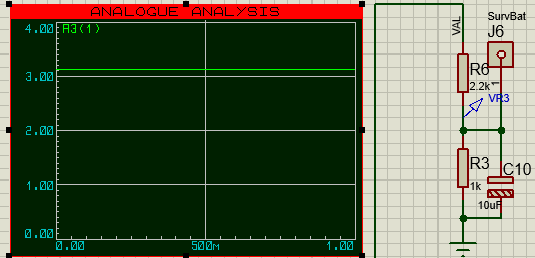


Figure 2 - Bloc surveillance de l'alimentation et tension en sortie du diviseur de tension

1. Reset

Une fonction « reset » est nécessaire sur une telle carte pour permettre son redémarrage en cas de mauvais fonctionnement sans avoir besoin de la débrancher. Nous utilisons alors un bouton poussoir qui déclenche une interruption sur le microcontrôleur U1 : celui-ci est au niveau logique haut lorsque le bouton est relâché et bas lorsque le bouton est enfoncé (niveau atteint après 3,6 ms de pression). L’interruption vient de la capacité C4 qui se voit déchargée lorsque le bouton pressoir est enfoncé : l’entrée 1 de U1 passe à la masse, le niveau logique est bas et l’interruption est déclenchée.

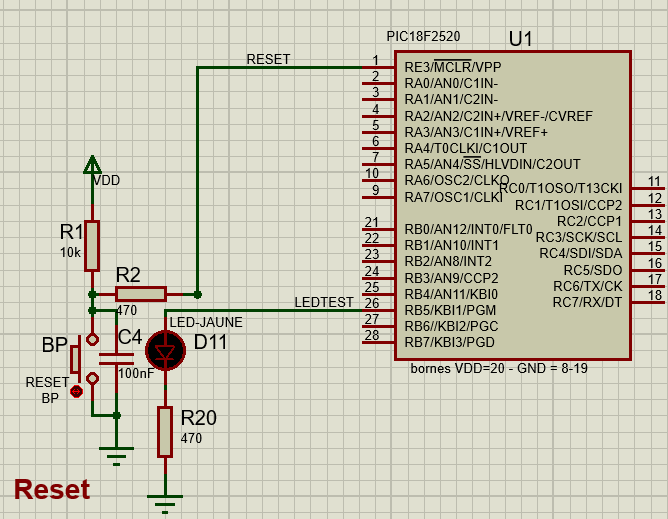


Figure 3 - Bloc Reset

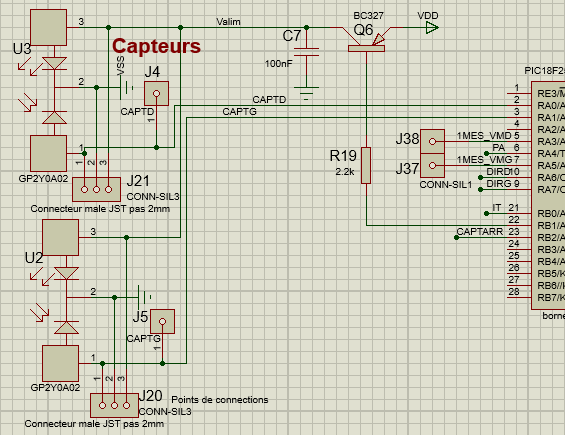
1. Test alimentation et surveillance batterie

Le témoin lumineux D11 est allumé lors de la mise sous tension du bloc alimentation. Notons que la tension minimale de UBat acceptable pour une sortie du régulateur de tension à 5V est de 7,3 V. Pour ce qui est de la formule évoquée précédemment, nous retrouvons expérimentalement :

1. Capteurs

*Le robot aura besoin de se déplacer dans son environnement. Il faut donc qu’il puisse appréhender*

*Celui-ci, ce qui est rendu possible par le biais de plusieurs capteurs.*

1. Capteurs infrarouges

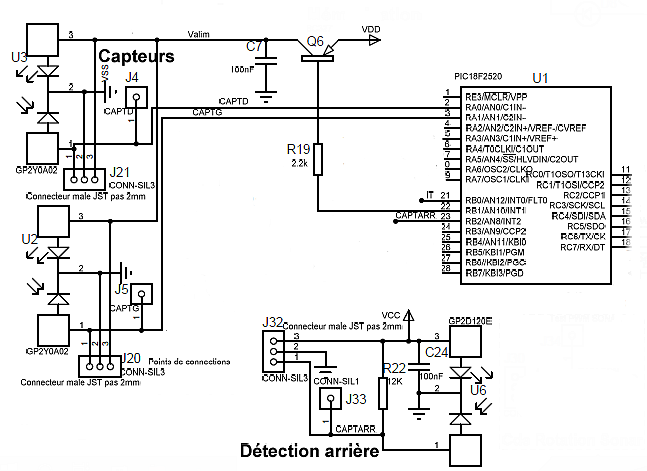


Figure 4 - Bloc Capteur IR

Le bloc capteurs IR assure au microcontrôleur U1 de connaitre la distance relative du robot aux obstacles.

Les capteurs IR discutent avec U1 sur les ports 2 et 3. La tension image fournie par ces capteurs est relative à la distance et semble suivre une loi hyperbolique. Notons la chute de tension imposée par le constructeur pour un objet à moins de 5cm pour éviter d’endommager les circuits.

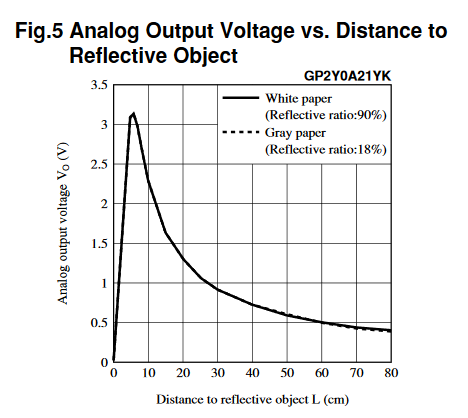


Figure 5 - Tension de sortie du capteur IR en fonction de la distance de l'objet

Le transistor Q6, qui voit sa grille reliée à U1, joue le rôle de commande de l’alimentation des capteurs : U1 agit sur la grille et peut faire commuter Q6, décidant d’alimenter ou non les capteurs à une tension VDD dans une optique d’économie d’énergie.

1. Test capteur infrarouges

Ce test nous a permis de vérifier la caractéristique du capteur IR, on remarque bien que la tension délivrer par le capteur décroit avec l’augmentation de la distance de l’objet, et on observe aussi un maximum pour une distance inférieur à 10cm

1. Le sonar

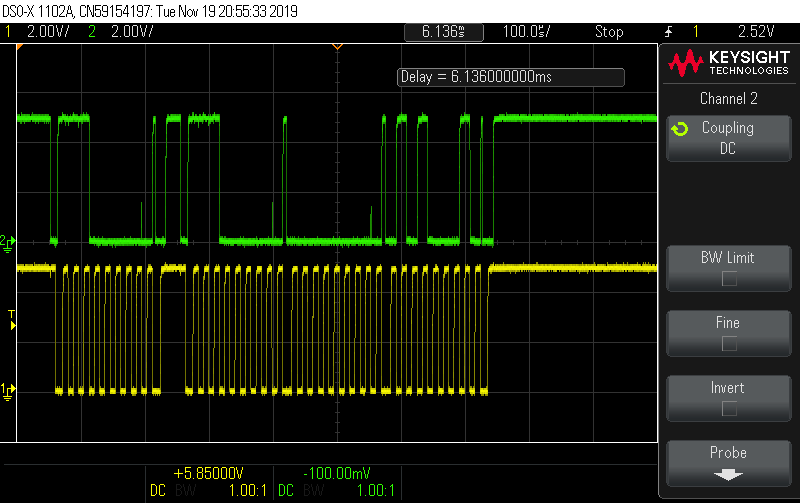


Figure 6 - Observations sur oscilloscope du bus I2C en sortie du sonar

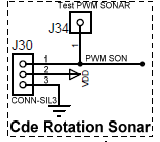
Ce bloc permet de détecter la présence d’obstacle autour du robot par échos sonores.

Le sonar possède un microcontrôleur interne, U10 commandé en I2C par U1. La lecture des données est instantanée. Le sonar possède un rayon d’action plus ample que les capteurs infra-rouges

1. Test sonar

On note pendant le test que le sonar est actif de 10 𝑐𝑚 à 1,70 𝑚 (mesuré par l’expérience et directement affiché sur la console). Notons par ailleurs la sensibilité du sonar au bruit.

1. Servomoteur

Pour que le robot puisse voir tout son environnement à 360°, on fixe le sonar sur un servomoteur. Ce dernier est commandé par une consigne d’angle PWM envoyée par le microcontrôleur U1. C’est le rapport cyclique du signal en entrée, qui détermine la rotation angulaire effectuée par le servomoteur.

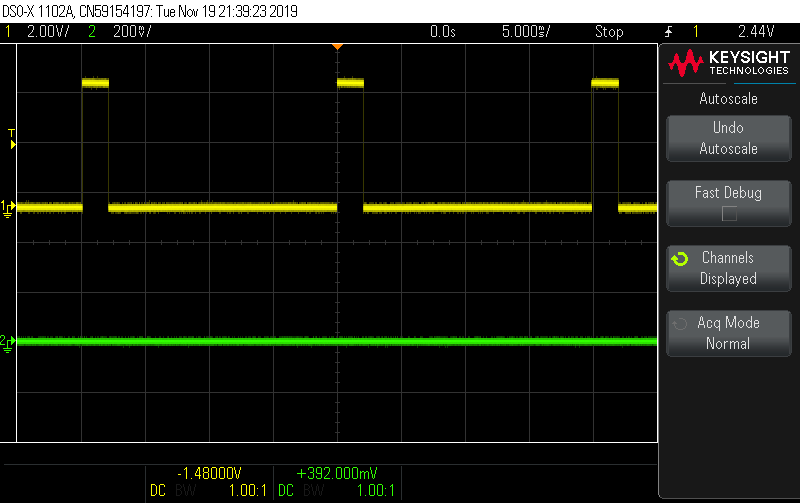
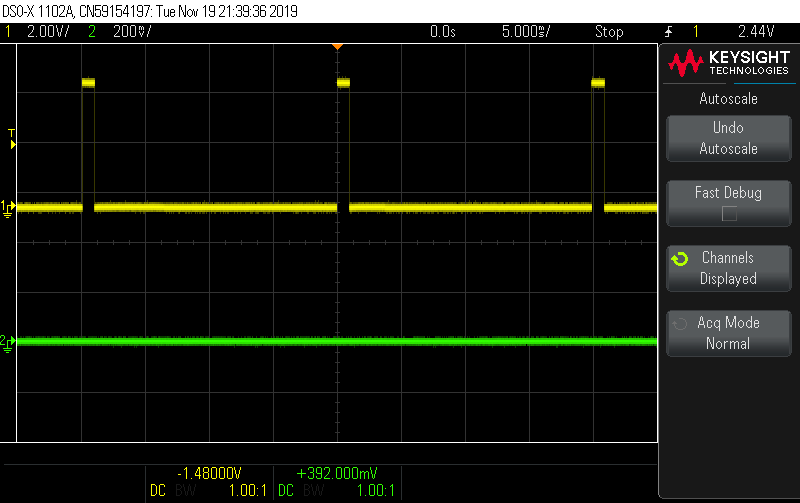


Figure 7 - Signaux PWM issus de U1 en entrée du servomoteur

1. Test servo-moteur

Ce test a permis de mettre en avant la dynamique angulaire du servo-moteur. Pendant le test on observe qu’un rapport cyclique du signal PWM en entrée de 1ms pour correspond à un angle de -90° et qu’un rapport de 2 ms correspond à un angle de +90°

1. Interface Homme/Machine
2. Affichage LEDs

*Durant l’exécution des tâches par le robot, aucun moyen de savoir si la carte a un problème ou si les tâches sont correctement effectuées, la connexion filaire étant possiblement dérangeante. Nous avons donc recours à un affichage LEDs pour prendre connaissance de ces informations.*

1. Principe et activation

8 LEDs composent le système d’affichage et un microcontrôleur U8, esclave du microcontrôleur principal U1. Ce dernier contrôle les 8 LEDs via U8 par bus I2C, évitant ainsi un surplus de fils. Les LED sont activées par les tensions imposées par U8 : 0𝑉 pour allumer une LED et 5𝑉 pour l’éteindre. La LED D1 et la LED D8 constituent respectivement le bit de poids faible (LSB) et le bit de poids fort (MSB) du bus. La LED D11 quant à elle nous informe sur l’état de la batterie.

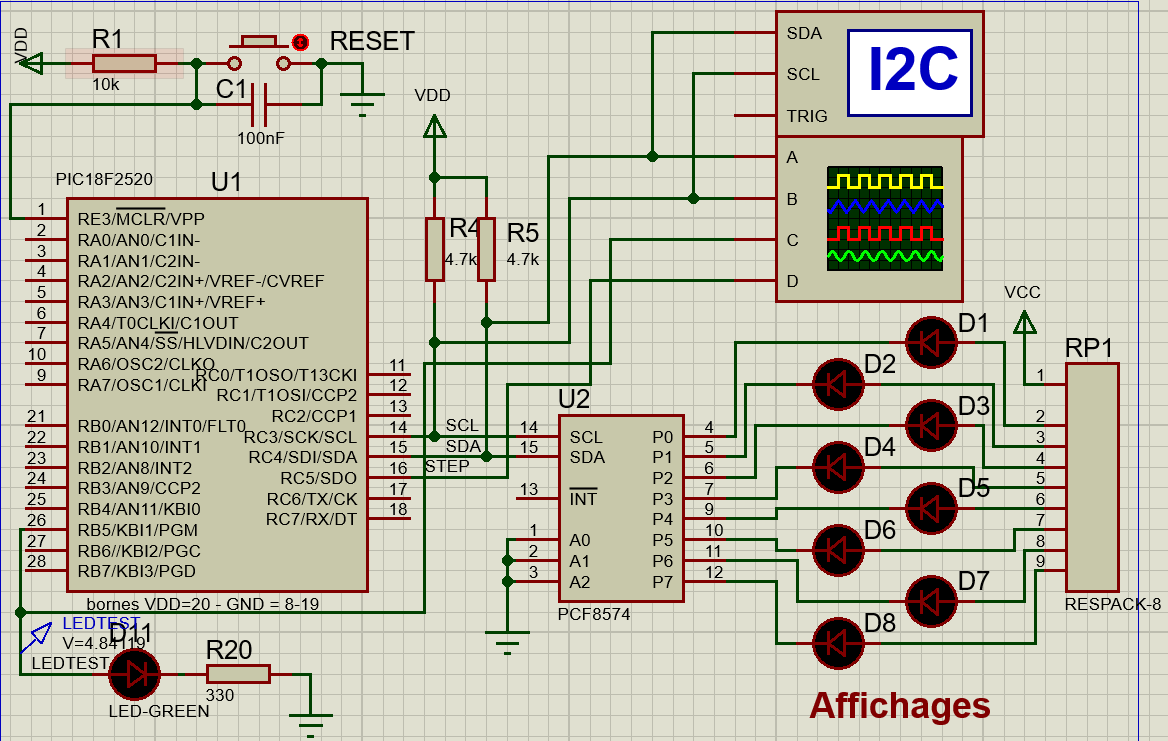
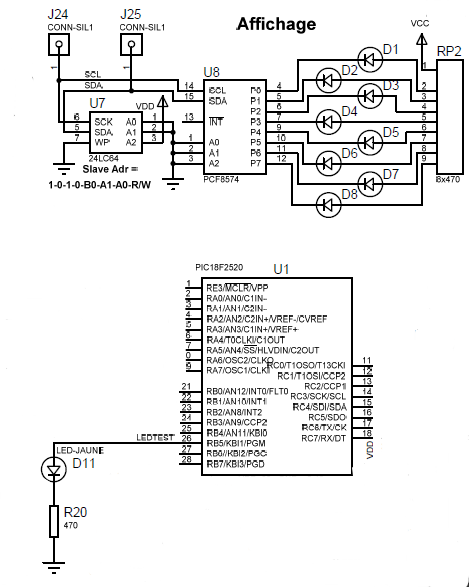


Figure 8 - Bloc affichage LEDs

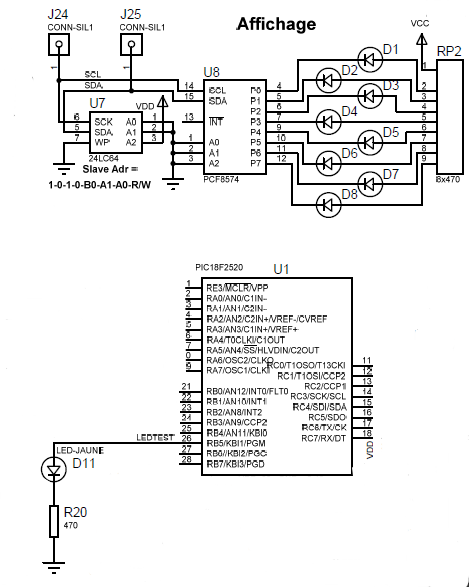


Figure 9 - Le microcontrôleur U8 et les 8 LEDs

RP2 est un réseau de résistances de 2,2kΩ, alimenté en 5V, qui permet d’éviter au LEDs de claquer et U7 est une mémoire programmable qui permet de sauvegarder l’état des LED en cas d’interruption (non monté sur la carte mais possiblement installable). Nous avons noté que le séquence d’activation pour la diode Di est le bit à 0 pendant que les autres sont à 1 sur le port  : D1 s’allume lorsque le code 1111 1110 est lu au port P0 par exemple.

1. Test affichage LED

Pour tester l’affichage LED, un programme a été implémenté dans le microcontrôleur qui permet de faire plusieurs tests sur les différentes fonctions de la carte. Le test d’affichage LED permet d’afficher un chenillard de période 5.7s et ainsi tester toutes les LED.

1. Télécommande :

*La télécommande permet de contrôler à distance sans fil le robot. Elle est capable de transmettre 5 instructions différentes (à l’aide de 5 boutons) que le robot pourra interpréter.*

1. Fonctionnement :

Ce bloc a pour objectif de contrôler les mouvements du robot grâce à une télécommande composé de 5 boutons. Le composant RX1 permet de convertir le signal HF analogique envoyé par la télécommande en un signal de données numériques qui sera communiqué au contrôleur U4.

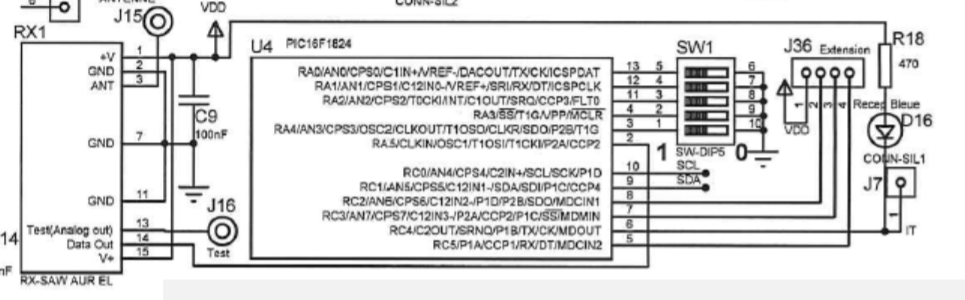
Le signal envoyé par la télécommande contient son adresse qui doit être la même qui le code donné par le switch SW1, ceci permet d’avoir un robot commandé par une seule télécommande, et ainsi éviter les interférences entre plusieurs télécommandes. L’instruction de direction est codée sur 3 bits qui est ensuite transmise par U4 en mode esclave au microcontrôleur U1 via un bus I2C. On note aussi la présence de la diode D16 qui permet de confirmer visuellement la bonne réception de l’information.

Figure 10 - Bloc réception télécommande

1. Test

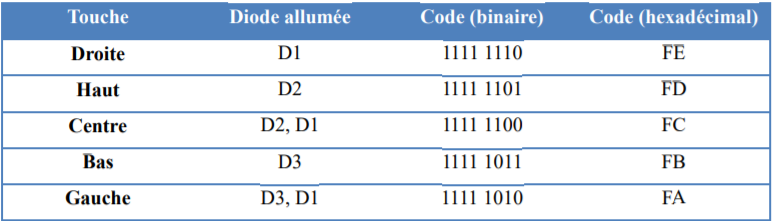
 Pour vérifier le bon fonctionnement de la télécommande le programme que l’on a intégré précédemment au microcontrôleur affiche à l’écran et sur les LED la direction que l’on a demandé. On peut ainsi remplir le tableau suivant :

Figure 11 - Tableau de correspondance Diode/code

1. Pilotage rotation roues-chenilles
2. Description fonctionnelle

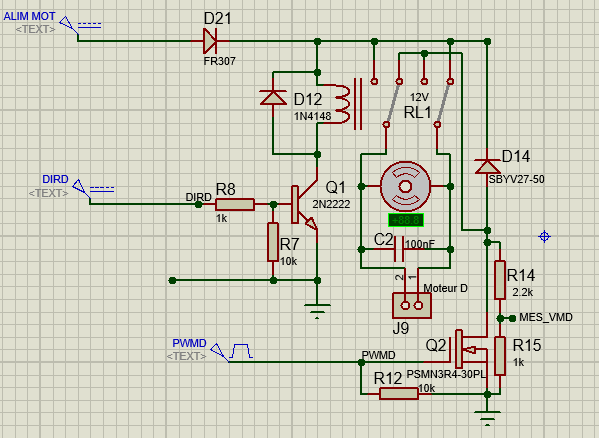
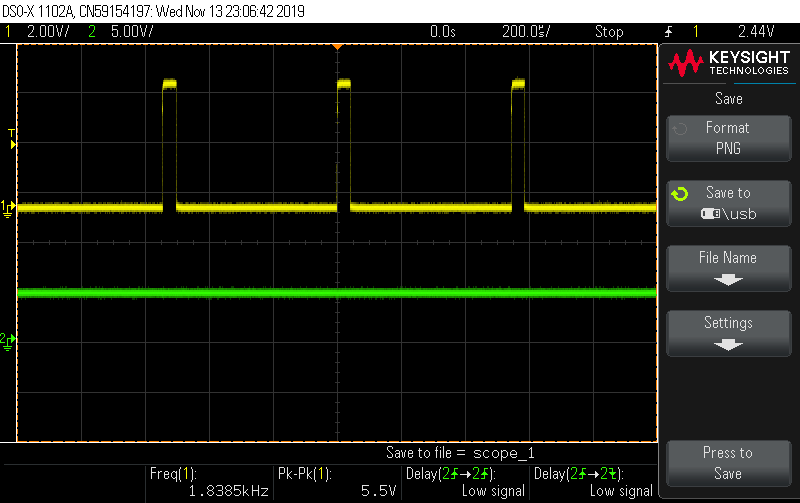
 Ce bloc permet de commander les moteurs droit et gauche indépendamment. Pour cela le moteur nécessite 2 variables de commande : une de direction et une de vitesse de rotation. Un fusible et une diode sont placés au niveau de l’alimentation pour éviter des courants trop élevés et que le courant ne puisse passer que dans un sens. La variable DIR permet de commuter le relais et ainsi changer le sens du courant dans le moteur et donc changer le sens de rotation. La diode associée au relais permet d’éviter les surtensions lors des commutations et on note aussi la présence d’une diode de roulis pour éviter l’apparition d’une tension d’induit créer par le moteur à courant continue. La vitesse de rotation est-elle gérée par la variable PWM qui module la tension moyenne aux bornes du moteur.

Figure 12 - Bloc moteur

1. Test

 Le second test permet de valider la commande moteur envoyé par le microcontrôleur. Pour cela on remplace les moteurs par des résistance d’impédance 330Ω pour modéliser la résistance de l’induit. On peut ainsi relever le PWM des moteurs droit et gauche et la commande de direction et on observe une variation de la tension moteur entre -7V et +7V. Ici on vérifie que la commande de direction fonctionne bien en s’assurant que l’ordinateur renvoie bien le sens de rotation voulu.

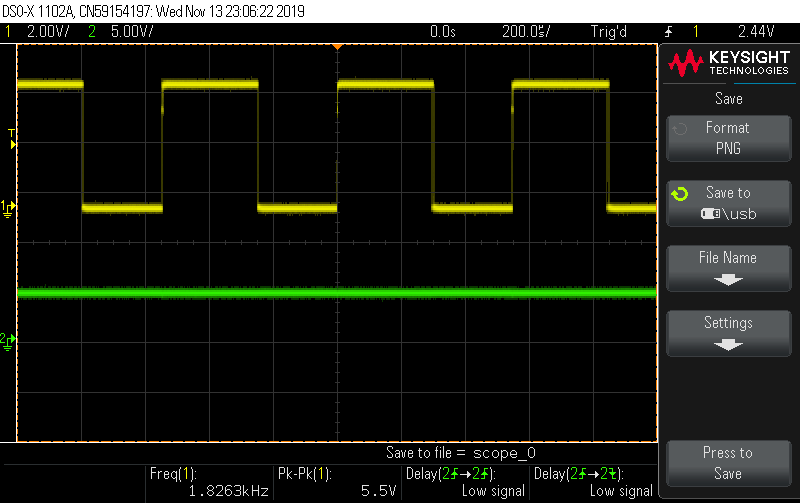
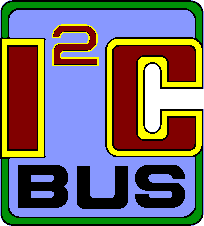


Figure 13 - Signaux PWM en entrée du moteur

1. Bus I2C

*La nécessité de libérer le microcontrôleur central U1 d’un trop gros nombre d’opérations nous impose d’utiliser d’autres microcontrôleurs au sein des diverses fonctions de la carte. Ces microcontrôleurs ne fonctionnant pas indépendamment, il est nécessaire de trouver un moyen pour communiquer les données entre eux et le microcontrôleur central. Un moyen efficace demeure le protocole I2C.*

1. Principe

Pour assurer la communication entre les différentes fonctions de la carte un bus I2C est implémenté au système pour échanger des informations. Ce bus est seulement composé de 3 fils pour faire passer toute l’information, un fil de masse, un fil SCL pour le signal de l’horloge et un fil SDA pour les données et les adresses. Il est utilisé pour l’affichage LED ainsi que pour la communication entre le microcontrôleur et la télécommande mais aussi pour la commande du sonar et du servomoteur. Ce bus I2C doit respecter un protocole dans la structure des données communiquée, en effet le signal démarre toujours par un bit de start, et qu’il soit reçu ou envoyé est décomposé en paquet de 8 bits. Le premier paquet étant l’adresse qui contient aussi le bit d’écriture /lecture, ensuite les autres octets sont des données, puis entre chaque paquet un bit « d’acknowlodge » (ACK) est envoyé par l’esclave pour confirmer qu’il a bien reçu l’information du maître. La trame se finit ensuite par un bit de stop.

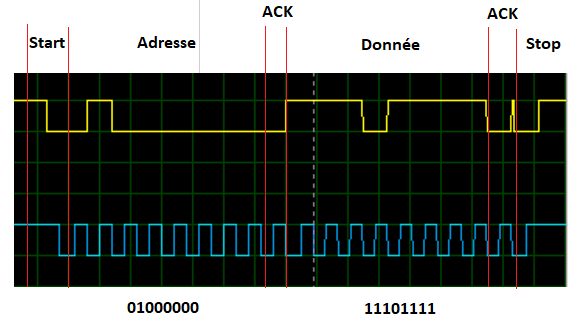


Figure 14 - Le principe I2C

On voit ici une trame envoyée à l’affichage LED qui comprend son adresse : 01000000 et la donnée à afficher sur les LED : 11101111. On remarque dans cette trame le bit ACK qui passe à 1 pour signaler la fin de la trame. Ce protocole est aussi mis en place pour la communication avec la télécommande qui envoie la donnée de direction au microcontrôleur. Notons aussi la présence de résistances de pull-up pour imposer un niveau haut les fils SCL et SDA quand il n’y pas de signal et ainsi éviter tout problème de communication sur un même canal :

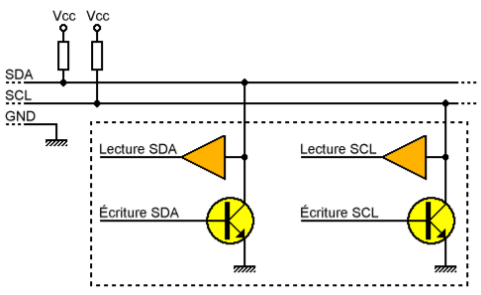
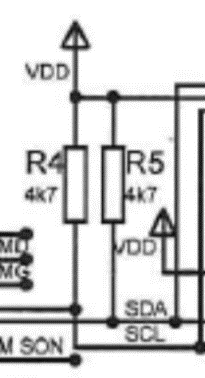


Figure 15 - Les canaux de communications SCL et SDA du I2C



1. Conclusion

La réalisation de cette carte nous a permis de mettre en évidence les différentes fonctions de la carte qui n’apparaissent pas comme explicites à première vue. Ce projet nous a aussi permis de prendre connaissance du protocole I2C et de son efficacité dans la communication entre les fonctions de la carte. Les différents tests qui ont été mis en place ont aussi fait apparaitre des problèmes qu’il a fallu prendre en compte et qui nous ne soupçonnions pas comme les diodes en parallèles des relais par exemple, ou le remplacement de composants détruits comme le PIC18F.

Finalement nous avons compris de la complexité de la carte qui nous paraissait extrêmement compliquée au premier abord mais qui semble aujourd’hui à notre portée.