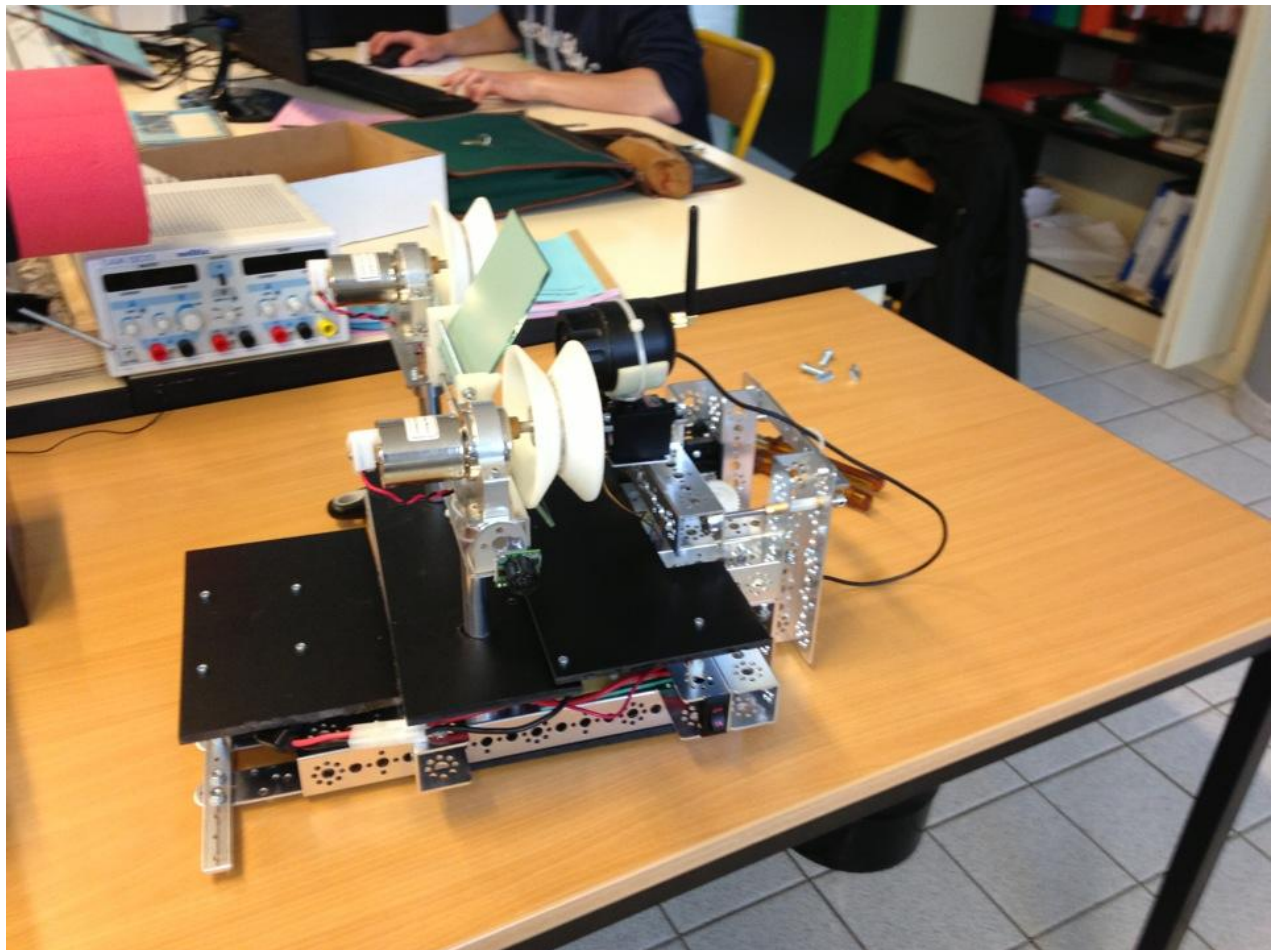


ROBOT INSPECTEUR DE CABLES

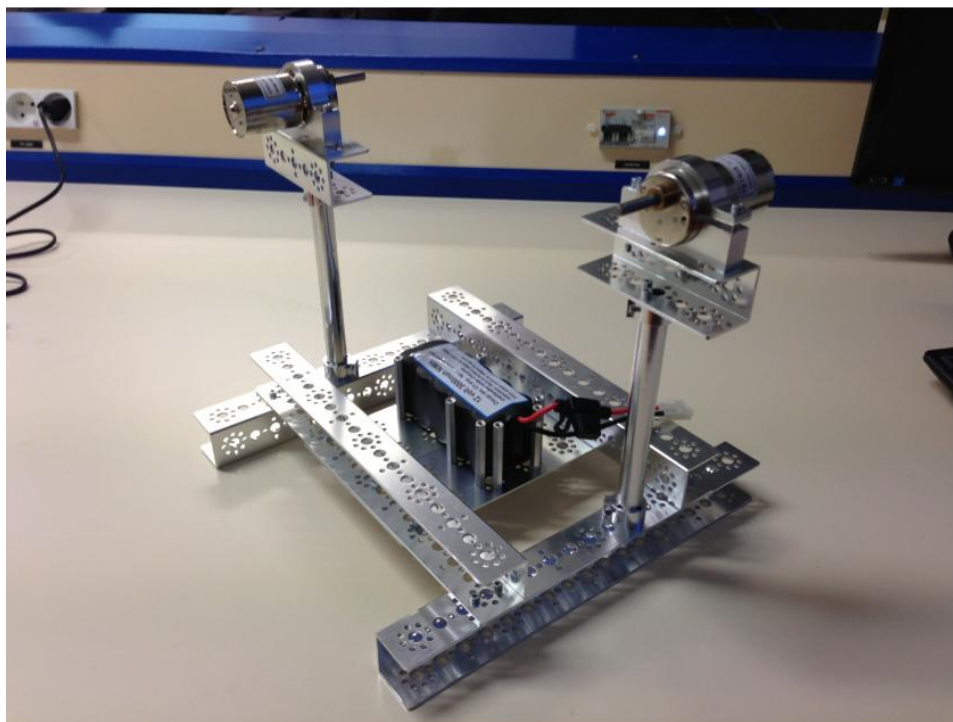


En ce début d'année de terminale S Si, nous ont été présentés différentes idées de projets que l'on pourrait réaliser pour notre projet personnel encadré. Le choix d'un sujet pour notre groupe composé de Robin Bertholet, Samuel Duval, Mathieu Delabays, Jérémy Villeveille, Ludovic Boury et de moi même, Jérémy Barra, s'est porté sur la réalisation d'un robot inspecteur de câbles. Notre projet, nommé « Spybird », a également été motivé par la participation aux Olympiades de Sciences de l'Ingénieur de l'académie de Lyon puis de sa qualification à la finale nationale de ces Olympiades de déroulant près de Paris au sein de l'entreprise ASTRIUM.

Le but de ce projet était donc d'étudier et de concevoir un robot capable d'inspecter visuellement l'état de câbles ou de lignes à haute tension. Nous avons tout d'abord pris connaissance des solutions qui peuvent exister sur le marché ce qui nous a permis de commencer à concevoir ce projet. Le choix s'est porté sur la conception d'un robot tenant en équilibre grâce deux roues motorisées posées sur le câble, une caméra communiquant avec un écran au sol permettant aux utilisateurs d'observer les éventuelles aspérités du câble. Ceux-ci pilotent le Spybird à distance et sans fil à l'aide d'une tablette ou d'un téléphone portable.

Nous avons donc mis en évidence trois grandes parties à réaliser et à se partager entre les six élèves : une première concernant la mécanique et l'équilibre du robot, une seconde sur l'électronique et la programmation des différents composant et enfin une troisième portant sur la communication entre les utilisateurs et le robot.

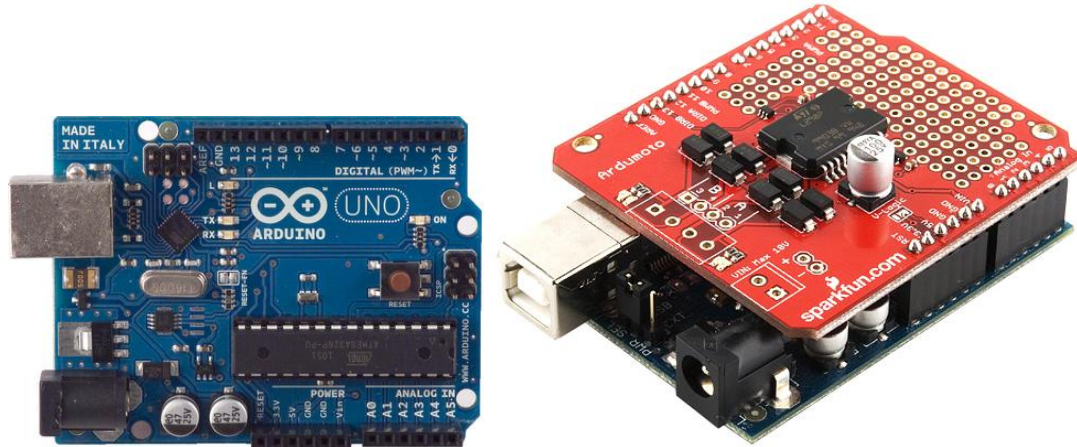
Voici ci-dessous une première ébauche de la forme qu'allait prendre le robot par la suite :



Nous nous sommes chacun attribués une partie du projet afin d'être plus efficace. J'ai donc travaillé en majorité sur la partie électronique avec Jérémy Villeveille.

I-Choix du matériel

Nous avons choisi d'utiliser une carte Arduino Uno pour programmer le robot. En effet, son prix assez bas, ses nombreux modules adaptables disponibles sur le marché et sa facilité d'utilisation en font un très bon choix. Pour piloter des moteurs nous avons utilisé un « shield », un module appelé Ardumoto, couplé à l'Arduino Uno il permet de diriger des moteurs en plus des entrées/sorties déjà présentes sur l'Uno.



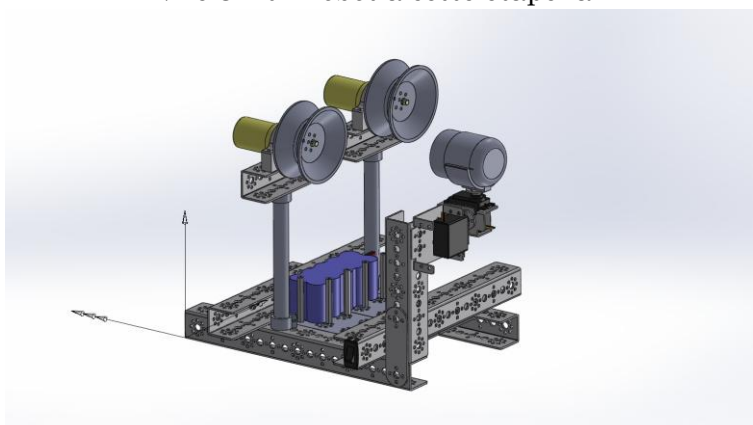
Nous avons utilisé deux capteurs à ultrasons, fixés de chaque côté du Spybird pour lui permettre de s'arrêter en cas de rencontre avec un obstacle trop important. Ils nous délivrent une tension analogique correspondante à un rapport de 5V/512 par pouce.



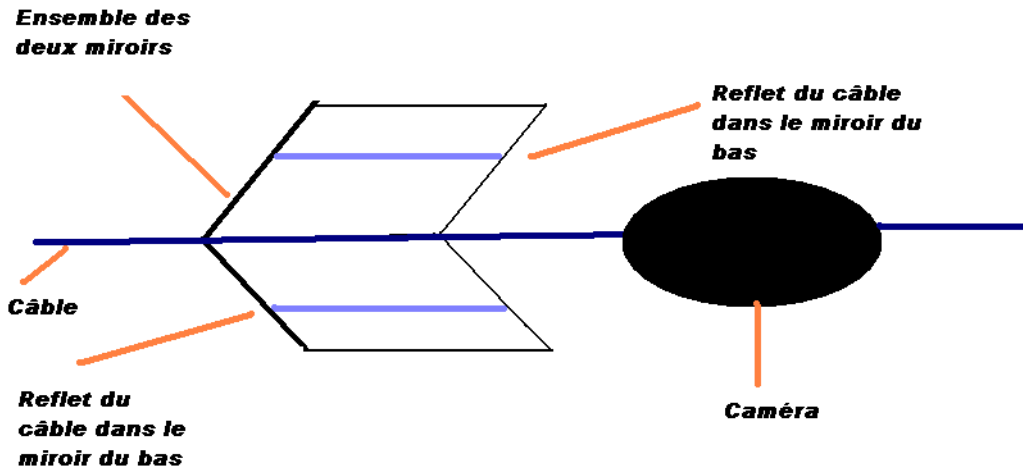
Pour établir une liaison entre l'utilisateur et le robot, la tablette nous offrait deux possibilités qui étaient d'utiliser le wifi ou le bluetooth. Le rendement des modules wifi pour Arduino était assez faible ils consommaient donc beaucoup d'énergie ce qui aurait réduit l'autonomie de la batterie. Nous avons donc choisi d'utiliser un module bluetooth qui nous donne une portée de 100 mètres.

Notre robot comporte également trois servomoteurs destinés à faire pivoter la caméra et à faire une mise au point. Avant la finale nationale des Olympiades il n'y avait que deux servos, le premier servait à orienter la caméra de haut en bas et le second de droite à gauche.

Vue 3D du robot à cette étape là :



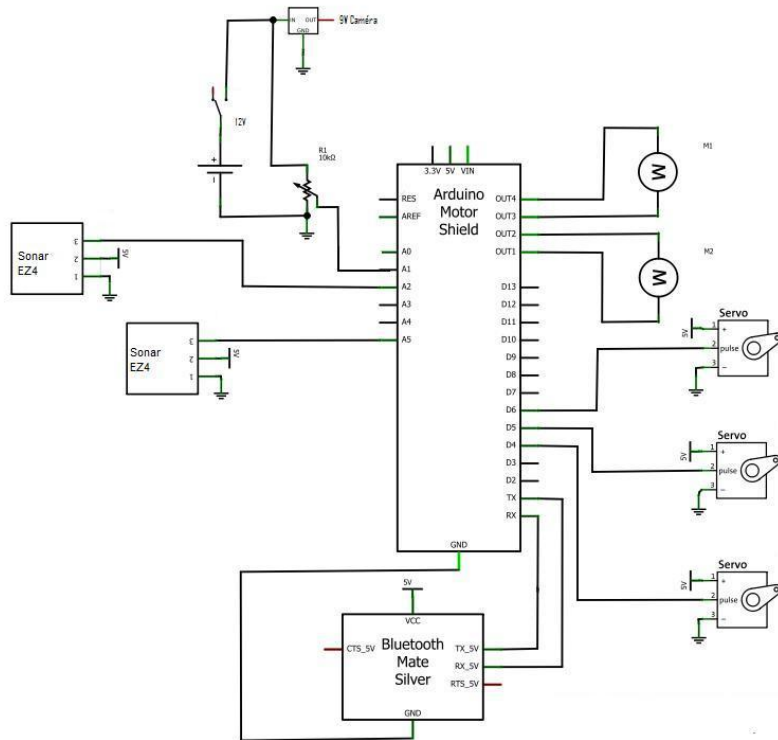
Nous nous sommes ensuite rendu compte que avec ce système nous ne pourrions observer qu'un seul coté du câble à la fois. Nous en avons donc rajouté un servomoteur qui fait avancer la caméra d'avant en arrière, voici ci dessous un schéma permettant de représenter la solution adoptée :



Lorsque ce servo fixé à la caméra est en position butée arrière, on voit le câble directement et ses reflets sont flous. Lorsqu'il est en position butée avant, les reflets deviennent nets et le câble flou. Ce système nous permet ainsi d'observer le câble sous tous les angles.

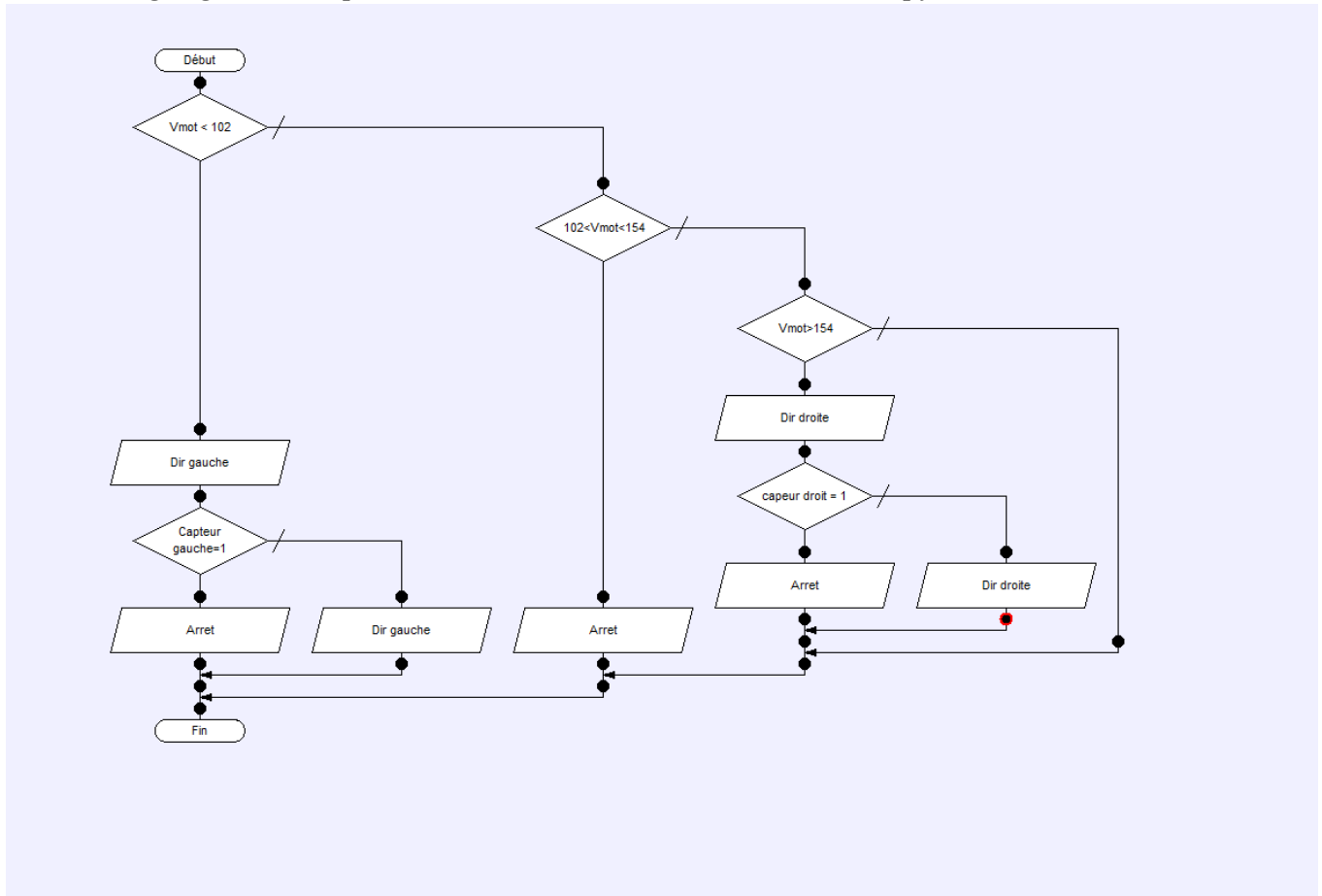
Tous ces composants ne sont pas alimentés par la même tensions (5V, 6V,9V), la seule batterie nous fournissant une tension de 12 V, nous avons utilisé des régulateurs de tensions à découpage. Ils permettent de passer d'une tension d'entrée de 12 Vcc à une tension de sortie de 5 à 9 Vcc.

Voici l'ensemble du schéma électrique du robot :

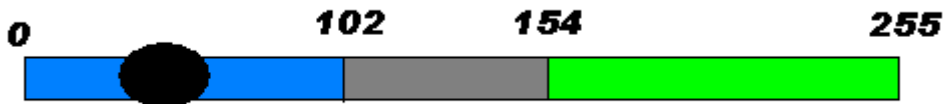


II-Communication avec l'utilisateur

Voici un algorithme représentant la commande des moteurs du Spybird :



Le Spybird est dirigé par l'utilisateur via une application Android. La commande des moteurs est sous cette forme :



Un curseur se déplace et prend une valeur de 0 à 255. Voici un extrait du programme pilotant le robot :

```
void commande_mot()
{
  if ((vmot>102)&&(vmot<154)) // On est ici dans le cas où le curseur est situé dans une plage de valeurs allant de 102 à 154
  {
    analogWrite(pwm_a,0); // pwm_a signifie que l'on dirige le moteur a, 0 signifie que le rapport cyclique du moteur est à 0
  }

  if (vmot>=154) { // Ici le cas où le curseur est situé entre 154 et 255
    int v=(vmot-140)*(255/115); // Expliqué plus bas
    analogWrite(pwm_a,v);
    digitalWrite(dir_a,LOW); // LOW configure le sens dans le quel tourne le moteur

    if (valeurcapteur2<20) // Si la valeur du capteur est inférieure à 20, on arrête le moteur
    {
      analogWrite(pwm_a,0);
    }
  }
}
```

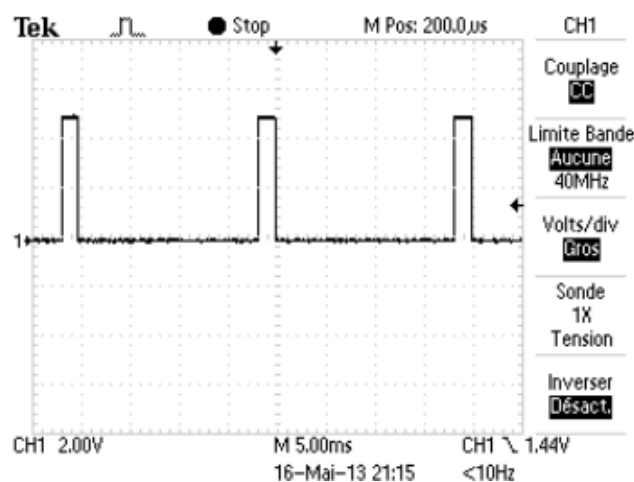
La valeur correspondant au rapport cyclique faisant varier la vitesse du moteur doit se situer entre 0 pour 0% et 255 pour 100%. Vmot étant la valeur envoyée par l'application, en théorie la variable v devrait être égale à $v=(vmot-154)*(255/101)$ où 101 correspond à 255-154. Seulement, nous nous sommes rendu compte que avec ce calcul, la vitesse des moteurs variait trop rapidement donc pour plus de précision nous avons utilisé $v=(vmot-140)*255/155$, comme si la zone d'arrêt du moteur était plus petite et s'arrêtait à 140.

Nous avons utilisé le même raisonnement lorsque le curseur est situé dans la zone gauche de la zone d'arrêt seulement ici, la tablette nous envoyait des valeurs allant de 0 à 102, v est donc égal à $v=(116-vmot)*(255/116)$.

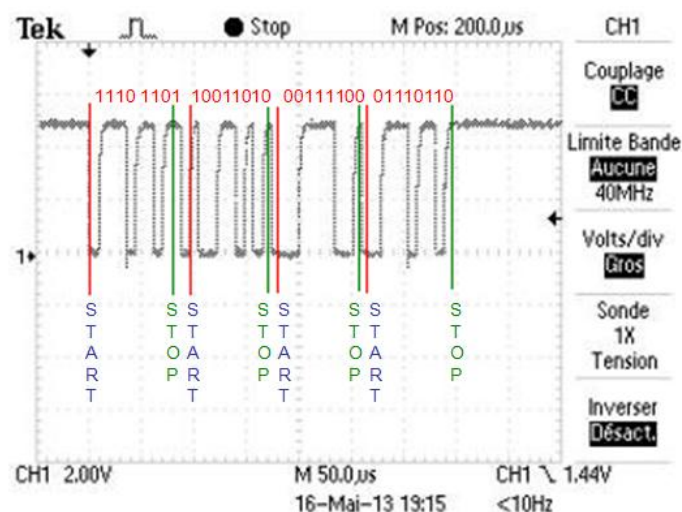
L'application envoie également des valeurs pour les servos allant de 0° à 180°, ce qui correspond à l'angle souhaité. La carte Arduino elle, renvoie la tension de la batterie à la tablette ce qui permet à l'utilisateur de connaître l'autonomie restante.

III-Mesures

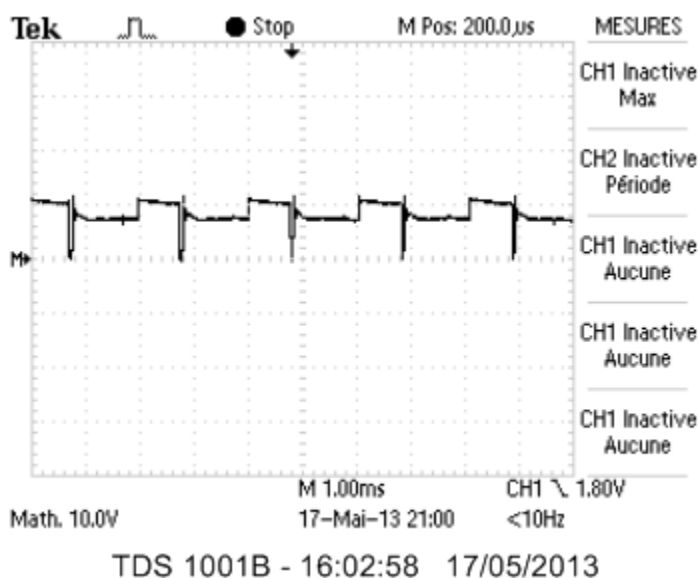
L'angle des servos est géré par la carte Arduino qui envoie des impulsions en permanence en fonction de l'angle demandé. Envoyées à une fréquence de 50Hz, la largeur de ces impulsions varie de 1 à 2ms et correspond à une valeur $L=angle/180+1$ en ms.



Ici par exemple la largeur est de 2ms ce qui correspond à un angle demandé de 180°. Nous avons également observé à l'oscilloscope le signal en sortie du module bluetooth, c'est à dire ce qu'il a reçu de la tablette et qu'il transmet à l'Arduino :



Nous pouvons observer ici le bit de start et celui de stop composant chaque trame de cette liaison série asynchrone. L'ordre des trames est celui-ci : Vitesse moteur, servo droite gauche, servo haut bas, servo mise au point. Nous remarquons aussi que l'on commence par le bit de poids faible et termine par celui de poids fort. Par exemple ici la vitesse du moteur était réglée sur 183 ce qui correspondrait à 1011 0111 en binaire, l'inverse de ce qui est représenté sur cette capture d'écran de l'oscilloscope.



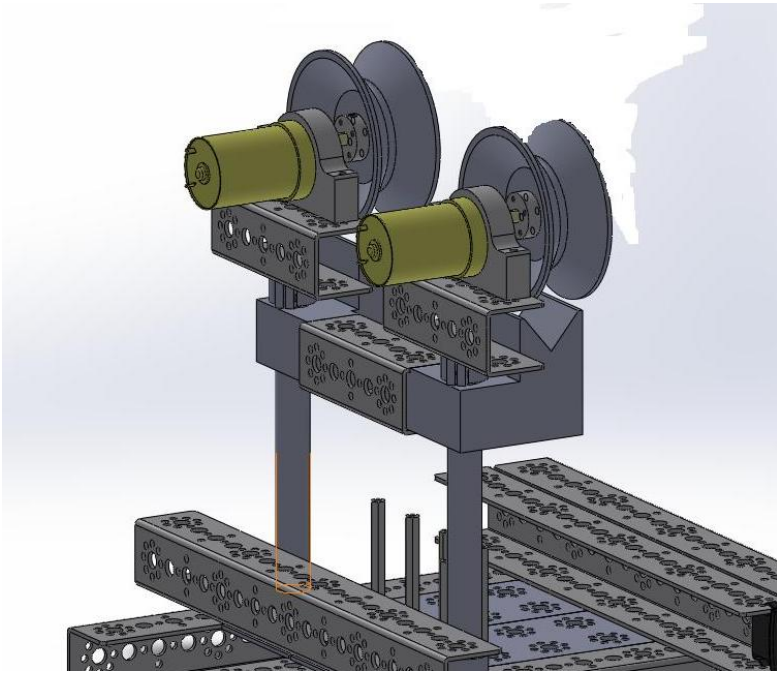
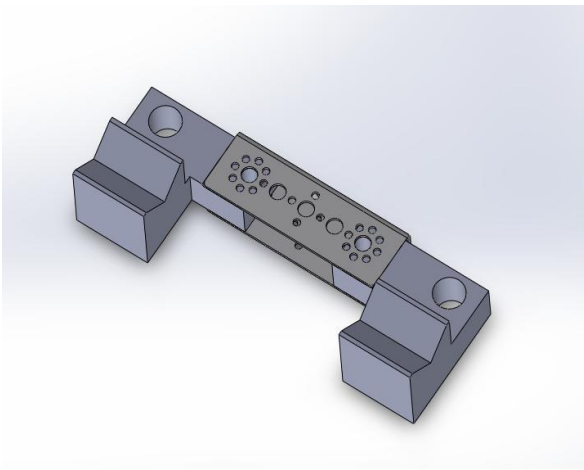
Cette capture d'écran d'oscilloscope représente la commande d'un moteur. Le rapport cyclique détermine la vitesse de rotation. Il correspond au ratio de la durée à l'état haut par la période du signal. Ici il est d'environ 45%.

IV-Perspectives d'évolution

Certains points pourraient encore être améliorés sur le robot. En effet, si celui-ci était destiné à l'inspection de lignes à haute tension, il faudrait adapter les composants électriques pour qu'ils puissent fonctionner avec le rayonnement magnétique et électrique créé par une ligne à haute tension. Celui-ci correspond à une fréquence de 50Hz, notre module bluetooth ayant une fréquence d'émission de 2.4Ghz et la caméra une de 5.8Ghz seraient déjà adaptés à cette utilisation car leur fréquence est bien supérieure à 50Hz.

Un point restant à améliorer est également la portée de notre robot. Le module bluetooth Mat Silver de Sparkfun utilisé sur le Spybird a une portée d'environ 100mètres. Nous avons pensé utiliser une liaison grâce à un module Xbee ayant une portée plus grande mais nous n'avons pas eu le temps de terminer la mise au point.

Enfin, la conception mécanique d'un dispositif permettant le passage d'obstacle et celle d'une sécurité empêchant le robot de se détacher du fil ont été étudiées. Cette sécurité a été dessinée sur Solidworks et nous l'avons imprimé avec l'imprimante 3D. Elle coulisse sur les deux tubes et vient se plaquer à 1mm des roues grâce à des ressorts (qui ne sont pas dessinés sur cette vue). Elle permet d'éviter au câble de sortir de son emplacement.



III-Conclusion

Ce projet a été très intéressant car il m'a permis de mettre en pratique les cours de Sciences de l'Ingénieur et de travailler en groupe avec d'autres élèves de ma classe. L'objectif du projet a été revu à la hausse au fil de l'année et surtout après notre qualification à la finale nationale des Olympiades. Je pense que l'ensemble du groupe a trouvé que cette expérience était stimulante et enrichissante.