Rapport Séance 31 Janvier 2022

Travail personnel

Les objectifs à faire en travail personnel étaient :

1. mettre en place le Bluetooth
2. terminer la construction du robot
3. faire un premier câblage complet du robot
4. programmer la télécommande

Pour commencer, je me suis intéressé à la connexion Bluetooth entre le téléphone et le module qui doit recevoir des informations. Il s'agit donc dans un premier temps de mettre en place un premier programme de connexion et d'appareillage du module afin de le configurer.

Bluetooth

```
//bluetooth robot

//Commandes:
// AT -> OK
// AT+NAMExyz
// AT+PINxxxx

//name: robot_Athlete
//Password: 1234

#include<SoftwareSerial.h>
#define RX 13
#define TX 12
SoftwareSerial BlueT(RX,TX);


void setup() {
  Serial.begin(9600);
  delay(500);
  Serial.println("Bonjour-Prête pour les commandes AT");
  BlueT.begin(9600);
  delay(500);

}

void loop() {
  while (BlueT.available()){
    Serial.print(char(BlueT.read()));
  }

  while (Serial.available()){
    BlueT.write(char(Serial.read()));
  }
```

Programme de configuration d'un module Bluetooth avec Arduino

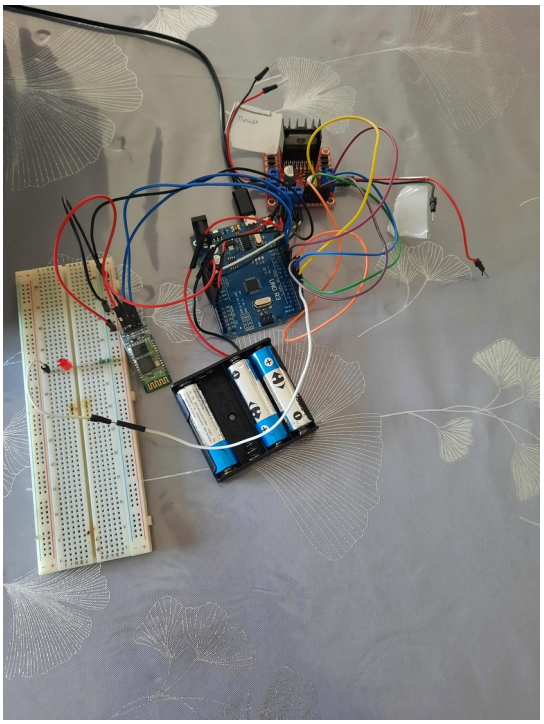
Ce programme permet de nommer le module « robot_Athlete » et de se connecter à lui. Nous pouvons à présent créer un programme principal.

Pour commencer à utiliser le Bluetooth, j'ai créé un programme qui permet d'allumer une led via ce système . J'ai d'abord créé une télécommande avec un bouton sur le téléphone, qui envoie deux type de message selon si l'on appuie ou non sur le bouton. Cela à pour conséquence l'allumage de la led.

Ce système servira par la suite à contrôler les moteurs.

```ino
// TEST Led et télécommande

#include<SoftwareSerial.h>
#define RX 13
#define TX 12
#define LED 2
SoftwareSerial BlueT (RX,TX);

char Data;
boolean etat = false;

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600);
  BlueT.begin(9600);
  pinMode(LED,OUTPUT);

}

void loop() {
  // put your main code here, to run repeatedly:


  if(BlueT.available()){
    Data = BlueT.read();
    Serial.println(Data);


    if (Data == 'C'){
      etat = true;
      Serial.println("UN");
    }

    if (Data == 'c') {
      etat = false;
      Serial.println("DEUX");
    }

    if (etat==true){
      digitalWrite(LED,LOW);
      Serial.println("allumée");
    }

    else {
      digitalWrite(LED,HIGH);
      Serial.println("éteint");
    }
  }
}
```

Programme de contrôle d'une led par Bluetooth avec Arduino



Montage de contrôle d'une led par Bluetooth avec une carte Arduino et un module de connexion

Pour suivre, je me suis occupé de la création de la seconde partie du robot. J'ai tout d'abord recopié un symétrique de la partie avant avec des Legos en essayant d'homogénéiser les couleurs. Ainsi l'avant sera jaune et noir, et l'arrière rouge et noir. Ces couleurs permettrons par la suite de distinguer les différents moteurs, notamment dans le programme final. Je me suis aussi occupé de recréer les blocs moteurs avant et arrière afin de les optimiser au mieux. Une barre Lego a été rajoutée sur toute la longueur afin de solidariser l'ensemble et éviter une destruction dut aux forces d'écartements. Le robot est à présent terminé.

Lors d'un premier essai, on remarque que le moteur a des difficultés à soulever le poids de l'articulation avant. Il faut donc réadapter l'avant et l'arrière afin d'obtenir plus de puissance. J'opte donc pour l'utilisation d'engrenage. Après un premier test peut concluant avec deux engrenages de respectivement 8 et 24 dents, j'ai conservé deux engrenages de 8 et 40 dents afin d'obtenir une puissance de traction 5 fois plus grande. Mais on remarque une baisse considérable de la vitesse. Mais avec ce procédé, une tension de 3V suffit à faire marcher le mécanisme. Il ne devrait donc pas y avoir de problème avec les 6V du montage final.

Lors de cette finalisation j'ai aussi mis en place le dernier servomoteur. Avec l'aide du programme déjà établi lors des séances précédentes et de la même démarche de résolution (test des différents angles), j'ai rechercher les angles limites de ce servomoteur. De part la symétrie du robot on remarque que pour un même angle, l'un des servomoteur est dirigé vers le bas alors que l'autre est dirigé vers le haut. Il faut donc anticiper ce problème afin que les servomoteurs puissent marcher dans le même sens et rester au même niveau.

Enfin, j'ai mis en place un premier programme destiné à contrôler simultanément les deux servomoteurs. L'idée et qu'ils se positionnent systématiquement au même niveau en montant et descendant en même temps. Ce programme et destiné à permettre à l'utilisateur de contrôler en même temps l'intégralité des moteurs.

2Servo

```
// 2 Servo

#include <Servo.h>

Servo monServoRouge;
Servo monServoJaune;



void setup() {
  Serial.begin(9600);

  monServoRouge.attach(2);
  monServoJaune.attach(14);

  /*
  int positionServo =  monServoRouge.read();
  monServoJaune.write(120 - positionServo); // on met ServoRouge au même niveau que ServoJaune


  // MONTER

  for (positionServo ; positionServo<=120 ; positionServo++){
    monServoRouge.write(120 - positionServo);
    monServoJaune.write(positionServo);
    delay(40);
  }



  // DESCENDRE

  for (positionServo ; positionServo >=0 ; positionServo--){
    monServoRouge.write(120 - positionServo);
    monServoJaune.write(positionServo);
    delay(40);
  }
  */

  int positionServo =  monServoRouge.read();
  monServoJaune.write(positionServo); // on met ServoRouge au même niveau que ServoJaune
}


void loop() {

   int positionServo =  monServoRouge.read();


  // MONTER

  for (positionServo ; positionServo<=120 ; positionServo++){
    monServoRouge.write(positionServo);
    monServoJaune.write(positionServo);
    delay(40);
  }


  // DESCENDRE

  for (positionServo ; positionServo >=0 ; positionServo--){
    monServoRouge.write(positionServo);
    monServoJaune.write(positionServo);
    delay(40);
  }

}
```

Programmation de deux servomoteurs avec Arduino

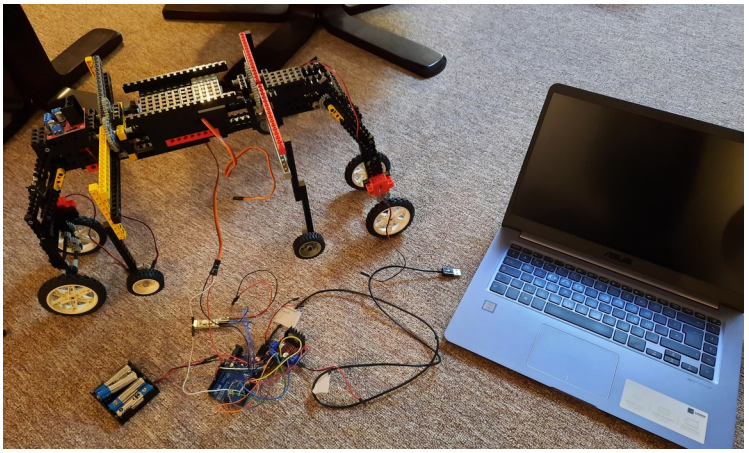


Image du robot grimpeur lors de la première phase de construction complète

L'objectif suivant est de câbler tout le robot. Afin de rendre son utilisation plus simple, j'intercale un bouton classique en sortie des piles. Puis je câble et relie tous les composants entre eux. Malheureusement l'utilisation d'une plaque d'essai est nécessaire afin de brancher tous les câbles. Ce qui alourdi encore le montage. Afin d'alimenter les moteurs de la nouvelle partie du robot je choisis d'utiliser les ports analogiques de la carte. En effet ils se situent au plus proche de ces moteurs et ne serviront probablement pas à la récupération d'informations analogiques dans ce projet. On remarque que par manque de matériel, des câbles de couleurs rouges et noirs sont utilisés

pour relier la carte au pont en H. Il faudra donc modifier cela à l'avenir. De plus certains câbles sont de trop petites longueurs.

Pour terminer, j'ai commencé la programmation finale du robot. Après avoir créé une télécommande spécialement pour contrôler le robot (composée d'un bouton de direction, de 4 séries de boutons de contrôle des articulations, et de 2 boutons pour contrôler l'ensemble des moteurs pour un même mouvement de monter ou descente) , je me suis consacré à faire marcher les roues et les moteurs classiques. Il est possible à présent de faire rouler le robot et de faire monter et descendre les trains avant et arrière avec une simple pression sur l'un des boutons de la télécommande. Mais les servomoteurs ne sont pas encore opérationnels. On remarque aussi que le module Bluetooth se déconnecte lorsque l'ordinateur est débranché. De plus les moteurs semblent marcher avec difficulté. Je suppose qu'ils soient sous alimenté du fait de la quantité de composants dans le robot. On remarque aussi que les servomoteurs seuls ne suffisent pas pour maintenir le robot.

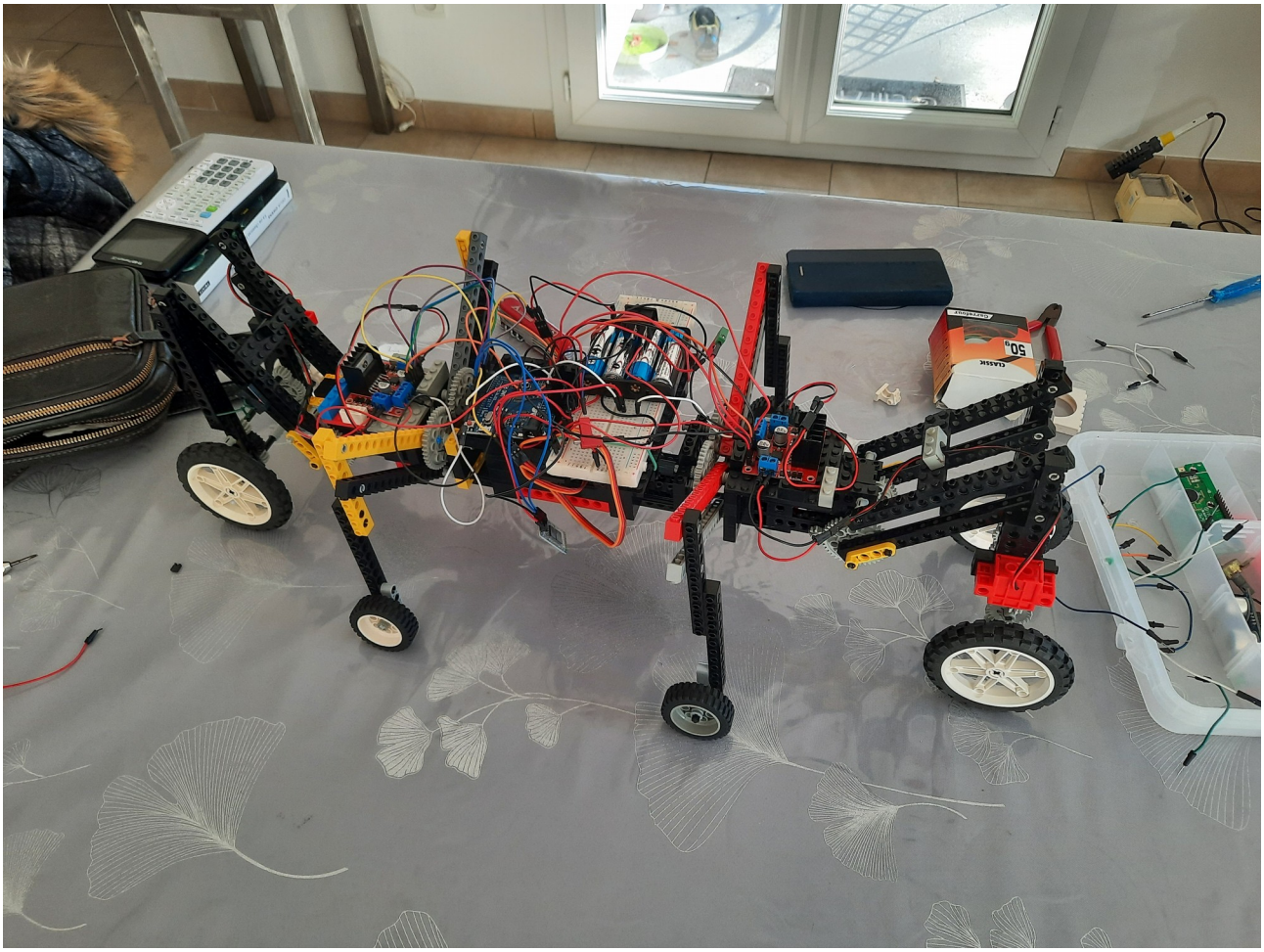


Image du robot et de son équipement

Objectif de la séance

- *Arranger le câblage des moteurs arrières (rouges)*
- *Résoudre le problème de déconnexion du module Bluetooth*

- *Résoudre le problème de manque de puissance dans les moteurs*
- *Programmer le mouvement des servomoteurs*

Partie 1 : *câblage des moteurs arrières*

Pour commencer la séance, j'ai changé l'intégralité des câbles reliant le pont en H à la carte Arduino afin de ne pas avoir de confusion quand au rôle de ces câbles. J'ai donc retiré tous ceux de couleurs rouges et noires utilisés temporairement lors du premier montage, pour les remplacer par des marrons, oranges, bleus et jaunes. L'objectif ici étant de conserver la convention : rouge pour l'alimentation, noir pour la masse.

Partie 2 : *Résoudre le problème de déconnexion du module Bluetooth*

Afin de résoudre le problème lié au module, j'ai demandé l'assistance de l'un des professeurs. D'après le montage, le module est branché au 5V de la carte. Mais avec l'utilisation d'un voltmètre, on remarque que la tension obtenue par le module est inférieure à 5V. Il y a donc une perte dans le circuit. J'ai donc branché le module directement aux piles afin d'obtenir une tension suffisante pour le maintenir en marche. Une fois cette modification effectuée, il n'y a plus de problème de déconnexion.

Partie 3 : *Résoudre le problème de manque de puissance dans les moteurs*

Pour continuer, j'ai cherché à résoudre le problème de puissance des moteurs en difficultés. Encore une fois avec l'aide du professeur, nous nous sommes aperçu grâce au voltmètre que la tension obtenue au niveau des moteurs était bien plus faible que la tension délivrée en théorie par les piles dans le circuit. En effet, les moteurs reçoivent une tension de 1,7V. Il y a donc bien une perte d'énergie importante. Nous avons aussi remarqué que plusieurs des piles utilisées ne délivrent pas la tension indiquée. Après avoir changé les piles usagées par des nouvelles, le problème reste identique. Nous avons donc substitué le bloc d'alimentation de 6V par une pile de 9V afin d'envoyer plus de tension au circuit. On note que à ce moment là nous avons rebranché le module Bluetooth sur le 5V de la carte car celui ci ne supporte pas des tensions supérieures à 6V. Malgré ce changement radical, les moteurs restent toujours en difficultés. En effet, une baisse de 4V est enregistrée dans le circuit entre la tension de sortie d'alimentation et la tension au niveau des moteurs. Finalement nous utilisons un câble de branchement rattaché au courant du secteur qui permet de compléter la tension des piles afin d'obtenir une tension suffisante au bon fonctionnement du matériel. Avec ce procédé nous obtenons enfin des résultats convaincants. Malgré tout cette solution n'est que temporaire car elle contraint le robot à être rattaché en permanence à une prise extérieure.

Lors de cette étape, on remarque aussi des soucis liés au branchement de certains câbles qui se déconnectent. Il faudra donc utiliser du ruban adhésif afin d'empêcher ce type de problème à l'avenir.

Partie 4 : *Programmer le mouvement des servomoteurs*

Pour finir, je me suis consacré à continuer la programmation des moteurs et de la télécommande. L'idée est d'obtenir un mouvement réactif. Lorsque l'on appui sur un bouton,

l’articulation dirigée par le servomoteur monte ou descend. Lorsqu’on relâche le bouton, le mouvement doit s’arrêter immédiatement. En m’inspirant des programmes établis auparavant lors du travail personnel, et du travail effectué à la première séance, je me suis établis à coder le servomoteur jaune situé à l’avant. Cependant le mouvement obtenu n’est pas encore le mouvement attendu. En effet lorsque j’enclenche mon bouton de montée, l’articulation s’active jusqu’à l’angle limite haut sans tenir compte de l’état du bouton. Il n’est pas suffisamment réactif et doit tenir compte de l’état permanent de ce bouton de pression.

Conclusion

De part le travail personnel et la séance de cours, le robot est à présent proche de son état final. La connexion avec le téléphone s’effectue et il est possible de contrôler certains moteurs avec une télécommande prête à l’emploi. Cependant certains problèmes persistent notamment la question de l’alimentation qui devra être résolue, la question du poids qui devrait pouvoir être encore optimisé en réduisant la taille de la plaque par exemple, la programmation des servomoteurs qui, malgré le fonctionnement actuel, ne correspond pas encore au résultat attendu, et les derniers ajustements concernant le câblage et la disposition des différents éléments.

Pour la prochaine séance :

→ terminer la programmation des servomoteurs
→ rechercher une plaque plus petite
→ résoudre le problème d’alimentation
→ ajustements
→ si possible faire un premier essai du robot pour tester son comportement et sa capacité à se déplacer, se transporter