

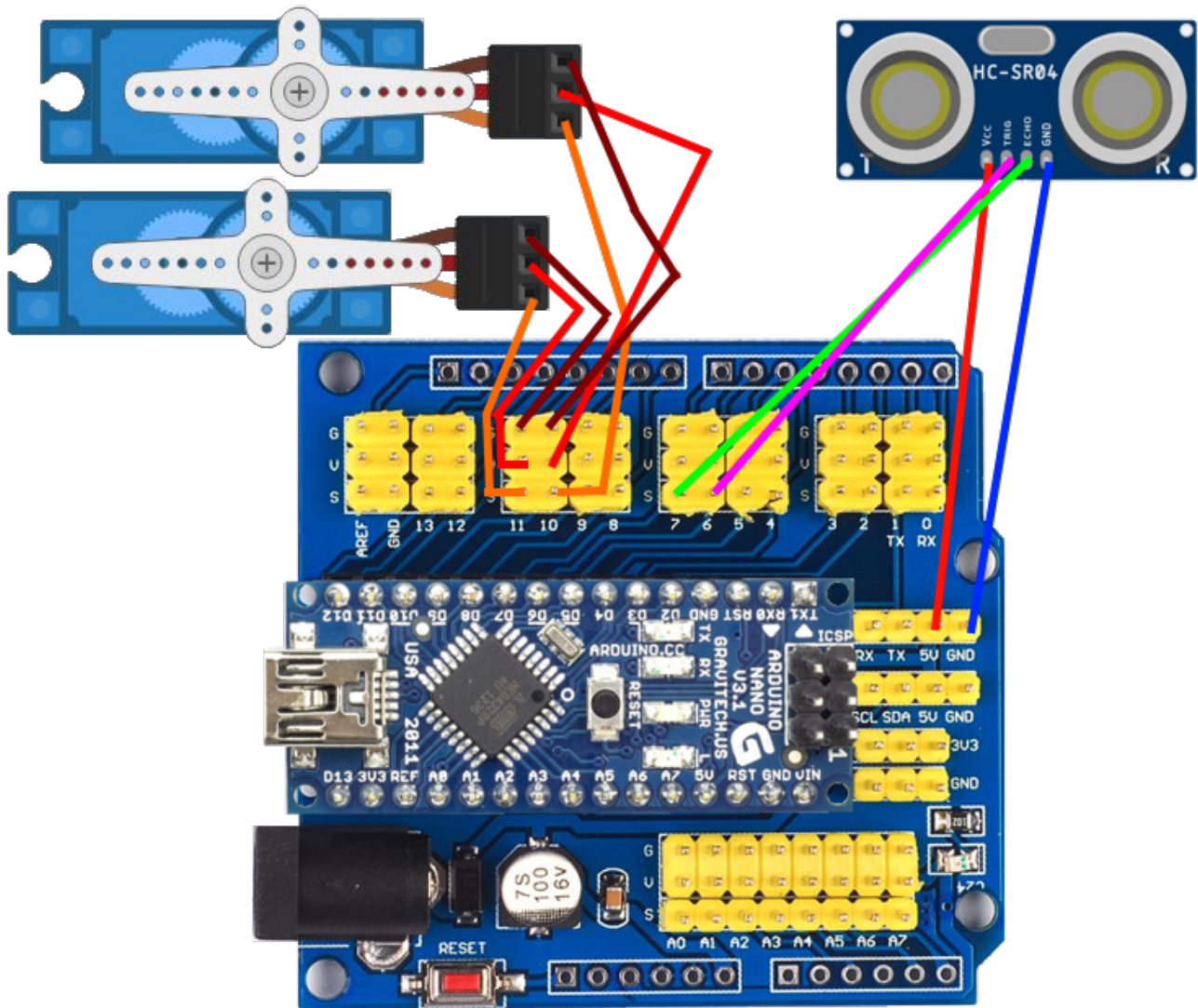
# Câblage et programmation du LittleBot

## Le câblage :

Pour le câblage et la programmation du LittleBot, il nous faut :

- Un Arduino Nano (ou équivalent)
- Un Sensor Shield
- Un capteur à ultrason (HC-SR04)
- 2 Servomoteur / Moteur (DM-S0090D)

Le branchement se présente comme ceci :



1. Tout d'abord nous branchons l'Arduino Nano sur notre Sensor Shield. Attention, il y a un sens. Le port de charge doit être sur l'extérieur de votre Shield.

2. Nous allons à présent brancher notre capteur à ultrason :

- VCC sur une broche 5V.
- Trig sur la broche 6.
- Echo sur la broche 7.
- GND sur une broche GND

3. Nous allons brancher nos Servomoteurs, les câbles de nos servos sont tous reliés à un raccord. Celui ci ne peut être branché que dans un seul sens. Nous brancherons donc un servo sur la broche

10 et un servo sur la broche 11.

- Le fil marron sur la broche G
- Le fil rouge sur la broche V
- Le fil orange sur la broche S

Ainsi le servo qui est sur la broche 10 sera notre roue droite et le Servo sur la pin 11 sera notre roue gauche.

Votre câblage est terminé.

Passons maintenant à la programmation.

## Le programme :

Ici nous décomposerons notre programme pour bien l'écrire.

Tout d'abord, nous déclarons la librairie et les servo que nous utiliserons :

```
#include <Servo.h>
#define trigPin 6
#define echoPin 7
Servo servo1;
Servo servo2;
```

Puis nous déclarons sur quelles pins sont branchés notre capteur et nos servos :

```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  servo1.attach(11);
  servo2.attach(10);
}
```

Rentrons dans le vif du sujet :

```
void loop() {
  long duration, distance;      // Nous déclarons notre variable que nous retrouverons plus tard
  digitalWrite(trigPin, LOW);  // Ici notre capteur à ultrason est en "position 0"
  delayMicroseconds(2);        // Pendant 2 Microsecondes
  digitalWrite(trigPin, HIGH); // Ici notre capteur à ultrason est "activé"
  delayMicroseconds(10);       // Pendant 10 Microsecondes
```

```

digitalWrite(trigPin, LOW);    //Puis nous le retournons en position "0"
duration = pulseIn(echoPin, HIGH); // Nous déclarons notre variable "duration" qui est la
durée du trajet du son.
distance = (duration*0.034) / 2;    // Nous déclarons notre variables "distance" par la durée
multiplié par la vitesse du son le tout divisé par 2.
if (distance < 20) {              // Nos déplacement commence ici, "Si la distance est
inférieur à 20cm alors..."
    servo1.writeMicroseconds(1000); //Servo Gauche tourne à l' envers
    servo2.writeMicroseconds(2000); //Servo Droit tourne à l' envers
    delay (2000); // pendant 2 sec
    servo1.writeMicroseconds(1000); //Servo Gauche tourne à l' envers
    servo2.writeMicroseconds(1500); //Arrêt du Servo Droit
    delay (2000); // pendant 2 sec
}

else {                            //Sinon...
    servo1.writeMicroseconds(2000); //Servo Gauche tourne
    servo2.writeMicroseconds(1000); //Servo Droit tourne
    delay (2000); // pendant 2 sec
}
}

```

Puis nous assemblons le tout, voici à quoi cela devrait ressembler :

```

#include <Servo.h>
#define trigPin 6
#define echoPin 7
Servo servo1;
Servo servo2;

void setup() {
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    servo1.attach(11);
    servo2.attach(10);
}

void loop() {
    long duration, distance;

```

```
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration*0.034) / 2;
if (distance < 20) {
    servo1.writeMicroseconds(1000);
    servo2.writeMicroseconds(2000);
    delay (2000);
    servo1.writeMicroseconds(1000);
    servo2.writeMicroseconds(1500);
    delay (2000);
}

else {
    servo1.writeMicroseconds(2000);
    servo2.writeMicroseconds(1000);
    delay (2000);
}
}
```

---

Revision #12

Created 18 September 2023 10:12:49 by Mathis Simoen

Updated 25 March 2025 09:39:30 by Quentin Petrazoller