

Rapport EMBI4

Exercice 1 ; Manipulation roue servomoteur :

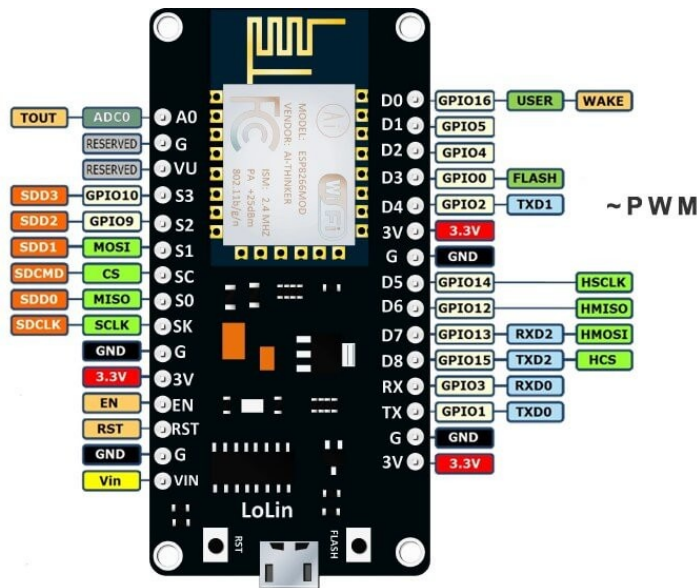
Le but de ce code est de faire tourner un servomoteur de manière circulaire de sorte à actionnée une roue.

Pendant mes recherches de la datasheet et d'information sur les servomoteurs. J'ai vu qu'il y en avait de deux types. Les servos avec une limite d'angle de 180 degrés et ceux qui ont 360° degrés. Ça veut dire quoi ? Quand on écrit le programme, on a des servomoteur qui on un point de repère zéro. Et qui savent accepter des valeurs de 0 à 360. et il y en a d'autres qui ne sont pas synchrone et qui ne savent que faire de petits mouvement compris entre 0 et 90° dans un sens et 91 à 180° dans le sens opposé.

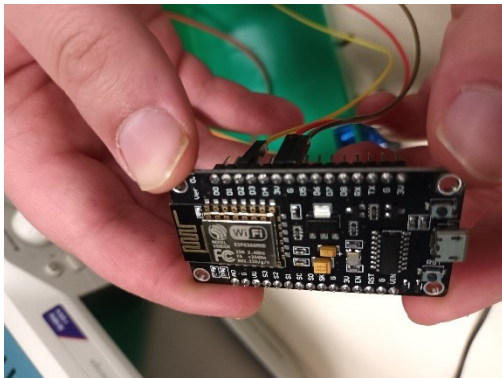
```
-----  
#include <Servo.h> // on inclut la bibliothèque servo  
Servo servoMoteur; // on crée un objet servo appelé servoMoteur  
void setup(){  
  // on associe le servo à la broche  
  //5 d'Arduino  
  servoMoteur.attach(5); // 5 = D2  
}  
  
void loop(){  
  // on déplace le servo à la position 0°  
  servoMoteur.write(10);  
  delay(1000); // pause 1 seconde  
  servoMoteur.write(40);  
  delay(1000); // pause 1 seconde  
  // on déplace le servo à la position 90° // on déplace le servo à la position 180°  
  servoMoteur.write(60);  
  delay(1000); // pause 1 seconde  
  
  /* la partie ci-dessous ne fait pas partie de la version final du code*/  
  // servoMoteur.write(270);  
  //delay(1000); // pause 1 seconde  
  // servoMoteur.write(360);  
  //delay(1000); // pause 1 seconde  
}
```

Les étapes manuels

- 1 On a du connecté les deux pin d'alimentation. (généralement le ground qui est le cable de sortie du courant est noir et celui de l'alimentation est rouge) celui du signal prends une autre couleur.
- 2 Puis on a du trouvé un pin de sortie du signal.
- 3 Nous nous sommes basée sur le schéma ci-dessous

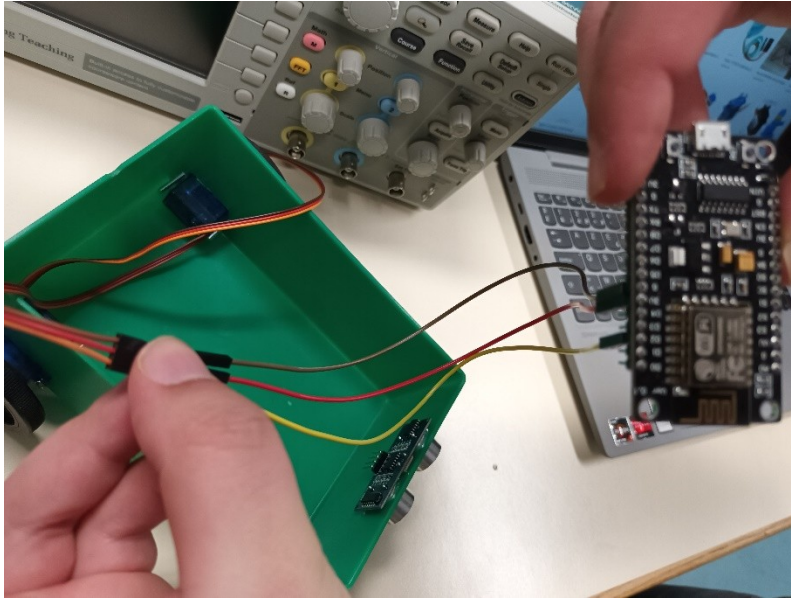


Ensuite on a vu qu'aucune réaction ne pouvait être observée. Pourtant le code compile. Donc on a exploré la piste d'un éventuelle faux-contact.



au finale ce qu'il s'est passé, c'est qu'on ne connaissait pas la correspondance entre les numéros de pins Arduino et ceux du NodeMCu ESP8266. Nous avons mal nommé nos sorties.

Solution : nous avons acheté notre propre carte Arduino de sorte à ce que chaque entré ou sortie noté sur la carte soit accepté à l'implémentation.



Exercice 2 : ampèremètre

le but de ce code est de capturer la valeur d'`analogRead()` (entre 0 et 1024) qui s'exprime en millivolt/ par unité selon l'alimentation de votre MCU

→ Arduino 5V = 4,9mV/unité

→ Arduino 3,3V = 3,2mV/unité

donc le voltage est égale à l'unité * 0,0049

ensuite, quand on a le voltage, on peut avoir l'intensité en ampère en soustrayant la tension à la tension à vide et en divisant le résultat de la soustraction par la « sensibilidad » qui est simplement le facteur de conversion.

```
float Sensibilidad = 0.066; //Sensibilité en Volts/Ampère pour capteur de 5A à 185mV/A -  
> ,0.185
```

```
//Sensibilité pour 30 Amp -> 66mV/A
```

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

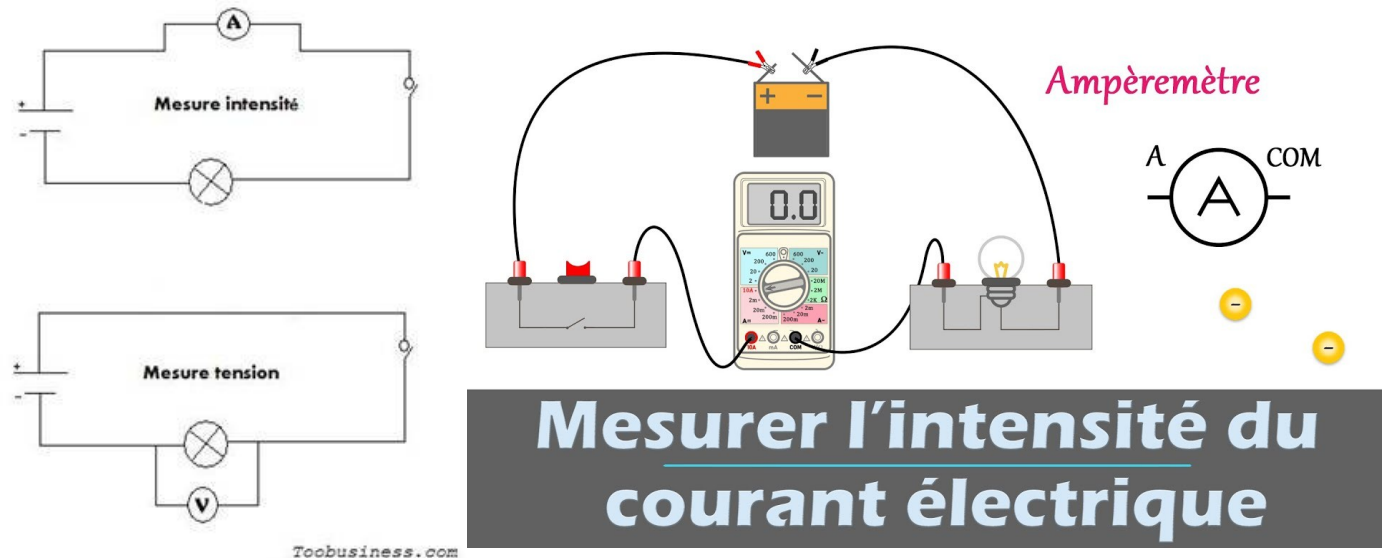
```
void loop() {  
    double voltajeSensor = analogRead(A0)*0.0048828125;  
    //(5.0 / 1023.0); //Pour la lecture du capteur  
    double I=((voltajeSensor - 2.49999) / Sensibilidad); //Formule pour obtenir le courant ou  
    l'intensité mesuré avec les cordons connectés au module ACS712
```

```

/*5000mV (tension de référence analogique - c'est la valeur par défaut (30.0 ->
30000Mv)*/
/*adcValue = analogRead(currentPin);
adcVoltage = (adcValue / 1024.0) * 5000;
currentValue = ((adcVoltage - offsetVoltage) / sensitivity);*/
Serial.print(voltajeSensor);
Serial.print(" ");
Serial.print(analogRead(A0));
Serial.print(" ");
Serial.print("La intensidad en Amperios es de: ");
Serial.println(I,3);
delay(200);
}

```

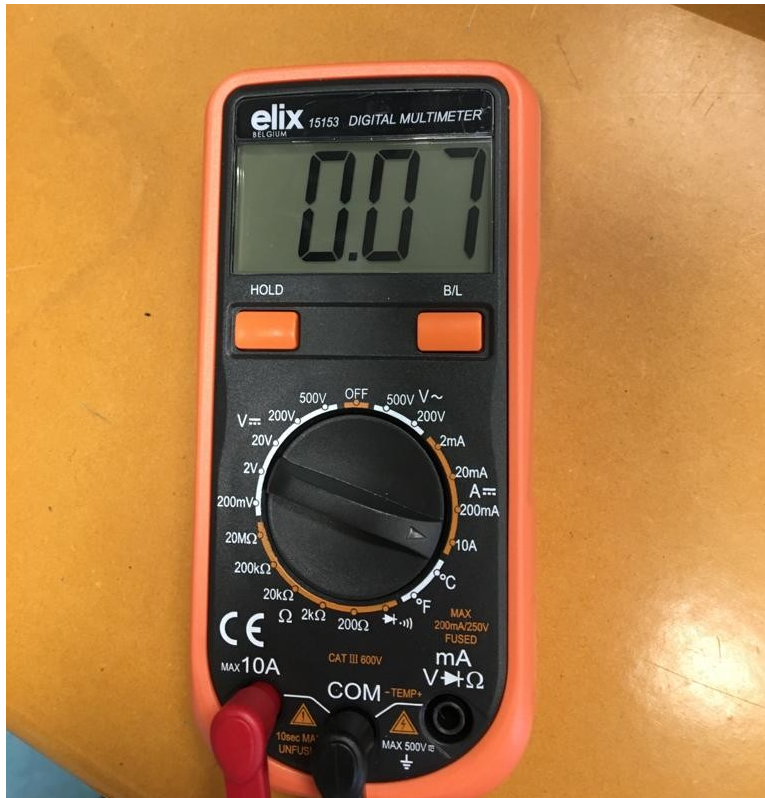
Ampèremètre en pratique



en cherchant sur le net : j'ai trouvé (en comparant plusieurs schéma sur le web) que pour mesurer l'intensité du courant électrique il fallait placé les sondes soit du coté de l'alimentation ou bien du coté du câble ground. Mais pas entre les deux.

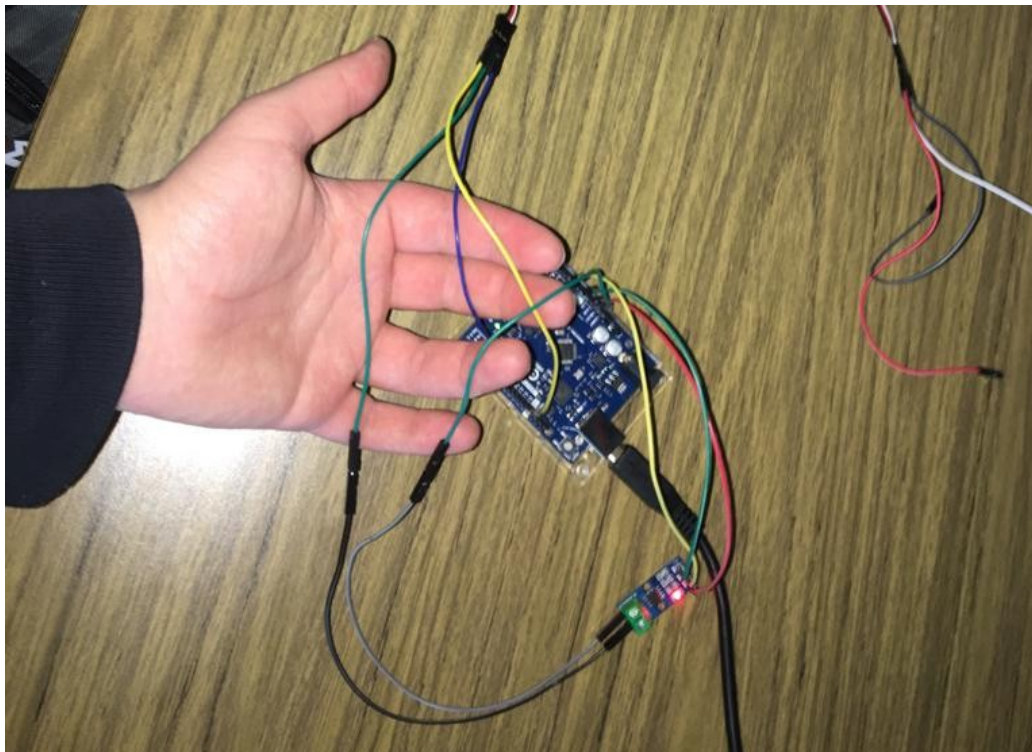
Si on place les sondes en cernant la résistance au milieu, on calcule mesure le voltage. A condition bien évidemment de bien réglé le cadrant. Comme décrit en dessous :

avec une échelle de 10 A

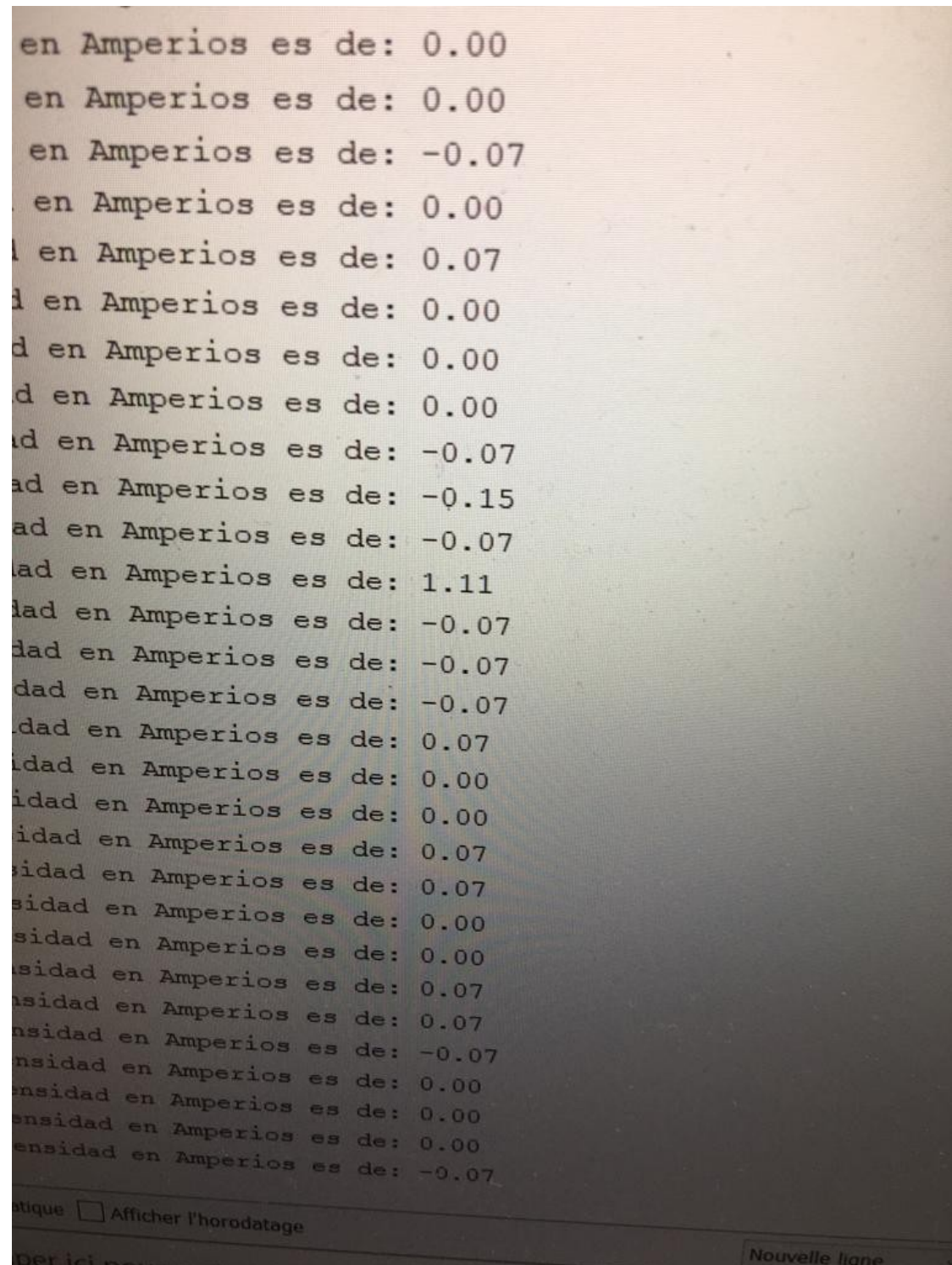


mais étant donné que l'on mesure qu'en ampère : on a pas besoin de ce bouton.

Comme décrit sur la photo joint en dessous. Le câble gris de « l'ampèremètre » est connecté à l'alimentation et le noir au VCC pour délivrer le courant.



Ainsi on trouve des valeur similaire au multimètre



The image shows a screenshot of a software interface, likely a data logger or measurement tool. It displays a list of 25 measurements, each consisting of a label followed by a value in Amperios. The labels are partially cut off on the left side. The values range from -0.15 to 1.11. At the bottom of the interface, there is a checkbox labeled 'Afficher l'horodatage' and a button labeled 'Nouvelle ligne'.

Label	Value (Amperios)
en Amperios es de:	0.00
en Amperios es de:	0.00
en Amperios es de:	-0.07
en Amperios es de:	0.00
en Amperios es de:	0.07
d en Amperios es de:	0.00
d en Amperios es de:	0.00
d en Amperios es de:	0.00
d en Amperios es de:	-0.07
ad en Amperios es de:	-0.15
ad en Amperios es de:	-0.07
ad en Amperios es de:	1.11
ad en Amperios es de:	-0.07
ad en Amperios es de:	-0.07
ad en Amperios es de:	-0.07
ad en Amperios es de:	0.07
idad en Amperios es de:	0.00
idad en Amperios es de:	0.00
idad en Amperios es de:	0.07
idad en Amperios es de:	0.07
idad en Amperios es de:	0.00
idad en Amperios es de:	0.00
idad en Amperios es de:	0.07
idad en Amperios es de:	0.07
idad en Amperios es de:	-0.07
idad en Amperios es de:	0.00
idad en Amperios es de:	0.00
idad en Amperios es de:	-0.07

atique ☐ Afficher l'horodatage

per ici pour

Nouvelle ligne

Exercice 3 : voltmètre

le but de cette exercice est de codé un voltmètre. Pour cette exercice on utilise un voltage sensor 25 volt qui contient deux résistances de respectivement 30kohm et 7,5kohm.

Le montage se comporte comme un pont diviseur de tension.par lequel l'Arduino mesure la tension entrante et sortante de la résistance de 7,5kohm. Encore une fois : comme pour l'ampèremètre, la valeur de retour d'analogRead est un entier compris entre 0 et 255 qui a une valeur symbolique en millivolt(1,9mV). Et du coup on applique la formule de la tension au borne de $U_n = U_s / (R_2 / (R_1 + R_2))$

```
int analogInput = A1;
float vout = 0.0;
float vin = 0.0;
float R1 = 30000.0; //
float R2 = 7500.0; //
int value = 0;
```

```
void setup(){
    Serial.begin(9600);
    Serial.print("DC VOLTMETER");
}
```

//dans un circuit, le courant électrique circule de la borne positive vers la borne négative

```
void loop(){
    // read the value at analog input
    value = analogRead(A1);
    vout = value * 0.0048828125; // tension de sortie convertie à l'entrée d'analogRead()
    vin = vout / (R2/(R1+R2)); // tension d'entrée trouvée en convertissant la formule

    Serial.print("INPUT V= ");
    Serial.println(vin,2);
    delay(500);
}
```

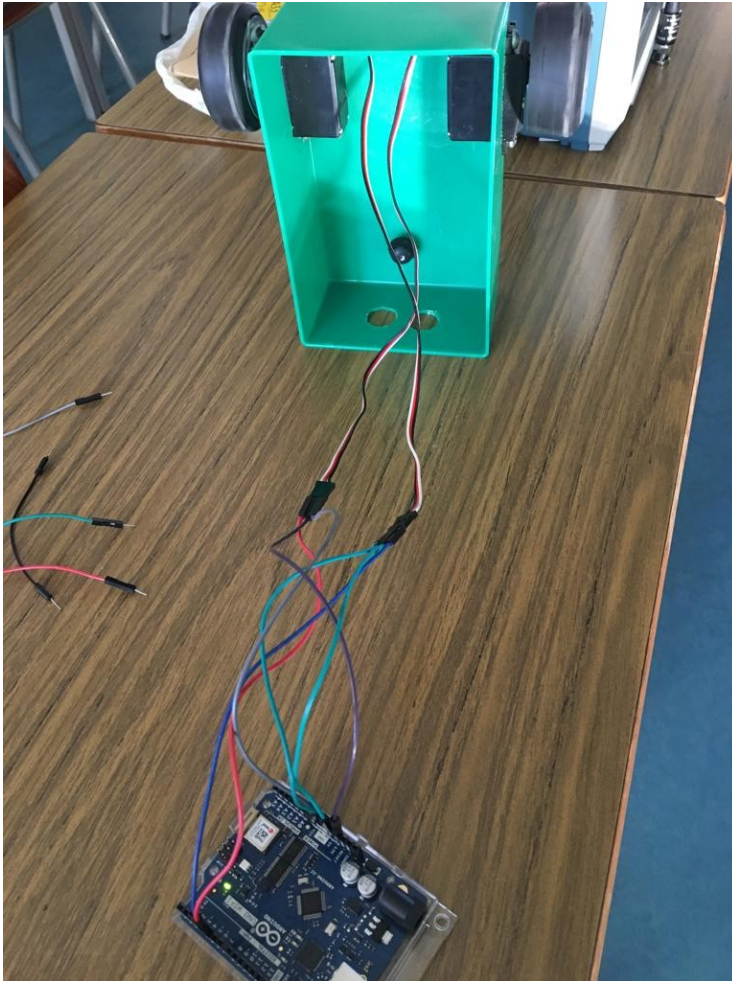
exercice 4 : Projet : Voiture électrique

cable noir = gnd

rouge = tension 3,3 ou 5v

blanc = entrée digitale

Installation/câblage



la terminaison bleu est liée à la sortie numéro 4 et la terminaison rouge, à la 5 en instanciant des Objets de type Servo.h je peux "écrire" des rotations de 90 degrés.

Dans un premier temps j'ai mis 45 degré partout.

Mais ça à engendré le fait que les deux servomoteur roulaient dans des senses opposés.

Après doc le servomoteurs de sortie 4 à reçu une valeur de 180 degrés pour tournées dans le sense inverse de sorte à ce que la voiture puisse rouler tout droit.

Ainsi après avoir écrit ma fonction pour aller tout droit. J'ai pu en déduire logiquement le code à écrire pour aller en arrière.

Pour la gauche et la droite, c'est plus compliqué. Il a fallu faire plusieurs testes pour "calibrer" les mouvements car on peut par exemple tourner à droite en :

- arrêtant la roue droite et en faisant rouler la roue gauche
- en faisant reculer la roue droite et en faisant roulé en avant la roue gauche
- en diminuant la vitesse de la roue droite.

Je suis parvenue à faire une rotation de 90 degrés à gauche et à droite au bout de plusieurs essais.

Pour la marche avant

```
#include <Servo.h> // on inclut la bibliothèque servo

Servo servoMoteur;
Servo servo2;// on crée un objet servo appelé servoMoteur

void setup(){
  // on associe le servo à la broche 2 d'Arduino
  servoMoteur.attach(5);
  servo2.attach(4);
}

void loop(){
  servoMoteur.write(0);
  servo2.write(180);
}
```

code circuit en 8 :

manipulation des servo de sorte à ce que ça fasse viré la voiture à gauche ou à droite.en formant un 8. grace à des appels de fonctions qui utilisent analog write pour pouvoir manipuler les deux servos en jouant sur les intensités de courants de sortes à crée artificiellement les mouvement sans bouton flechés ni souris.

```
#include <Servo.h> // on inclut la bibliothèque servo
Servo servoMoteur;
Servo servo2;// on crée un objet servo appelé servoMoteur
```

```
void setup(){

    servoMoteur.attach(5);
    servo2.attach(4);

}

void loop(){
    toutDroit();
    droite();
    toutDroit();
    droite();
    toutDroit();
    droite();
    toutDroit();
    gauche();
    toutDroit();
    gauche();
    toutDroit();
    gauche();
    toutDroit();
    gauche();
    toutDroit();

}

void Arriere(){
    servoMoteur.write(110);
    servo2.write(80);
    delay(1000);
}

void toutDroit(){
    servoMoteur.write(10);
    servo2.write(100);
    delay(1000);

}

void droite(){

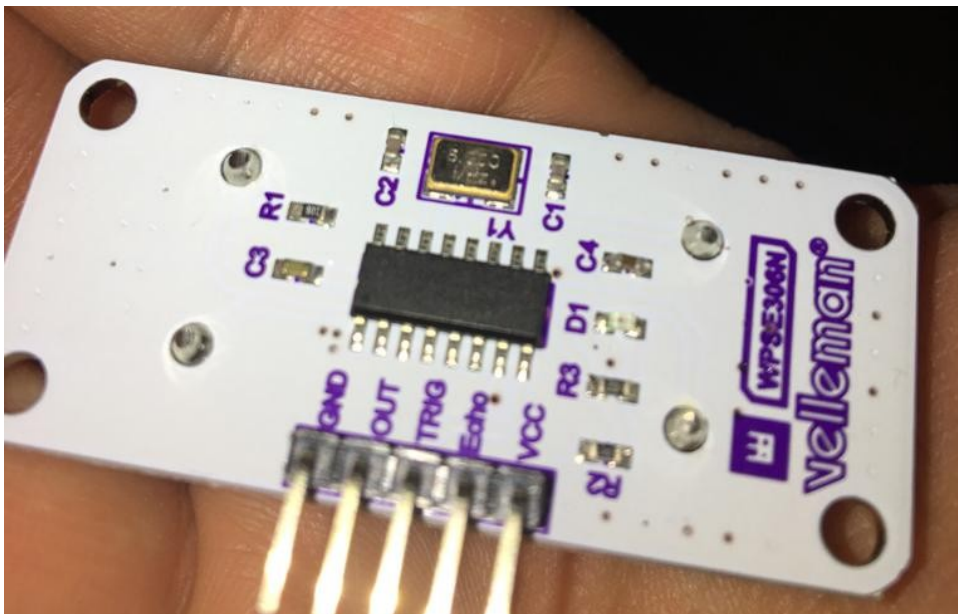
    servo2.write(100);
    servoMoteur.write(150);
    delay(750);
```

```
}  
void gauche(){  
  servo2.write(70);  
  servoMoteur.write(10);  
  delay(680);  
}
```

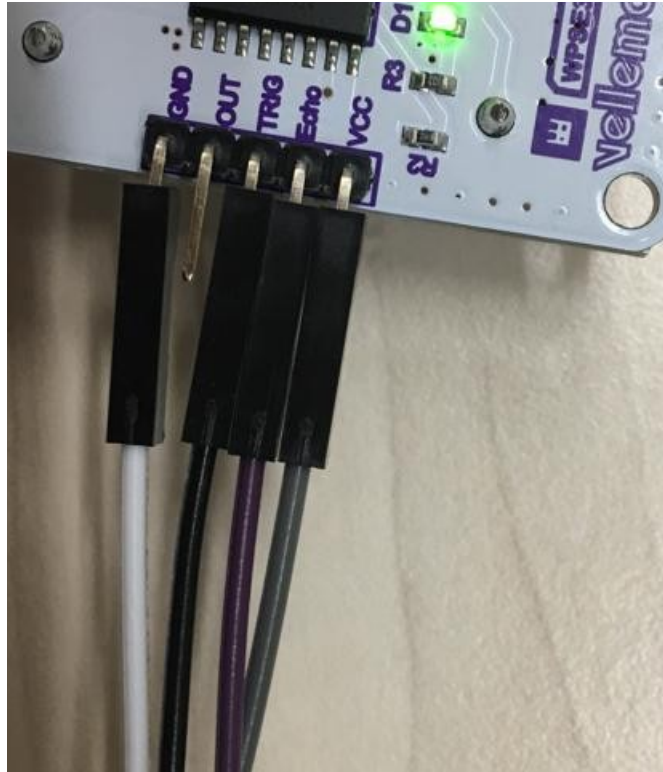
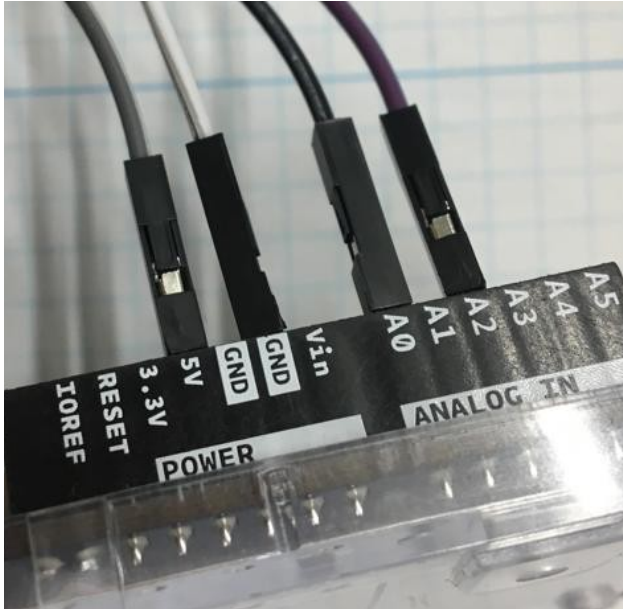
le Capteur Ultra son

Le principe du capteur ultra son est simple. Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence.(c'est à dire qu'elles sont imperceptible à l'oreille humaine). Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur. C'est pour cela qu'on a un pin « trigger » qui en anglais signifie « gâchette, déclencheur ». c'est lui qui envoie le signal ultrasonique. Et le pin echo, reçoit le retour comme un echo. C'est vraiment à prendre au pieds de la lettre.

Le capteur calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho.



Cablage



cable mauve :

echo→A2

cable noir :

trigger→A0

blanc = GND

gris = vcc (alimentation)

Voiçi ci-dessous le code qui me permet de calculer la distance grâce au capteur

la suite de mes explication après le code

```
/* Initialise le port série */
Serial.begin(115200);

/* Initialise les broches */
pinMode(TRIGGER_PIN, OUTPUT);
digitalWrite(TRIGGER_PIN, LOW); // La broche TRIGGER doit être à LOW au repos
pinMode(ECHO_PIN, INPUT);
}

void loop() {

  /* 1. Lance une mesure de distance en envoyant une impulsion HIGH de 10µs sur la broche
  TRIGGER */
  digitalWrite(TRIGGER_PIN, HIGH);
```



```

delayMicroseconds(10);
digitalWrite(TRIGGER_PIN, LOW);

/* 2. Mesure le temps entre l'envoi de l'impulsion ultrasonique et son écho (si il existe) */
long measure = pulseIn(ECHO_PIN, HIGH, MEASURE_TIMEOUT);

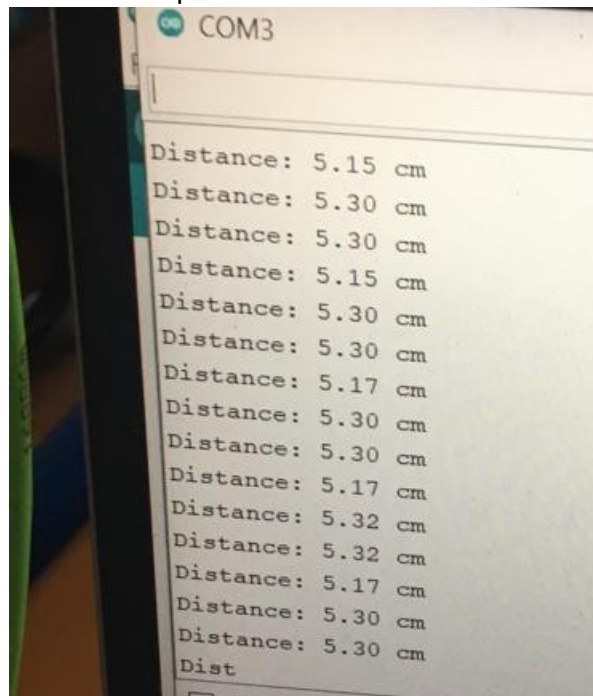
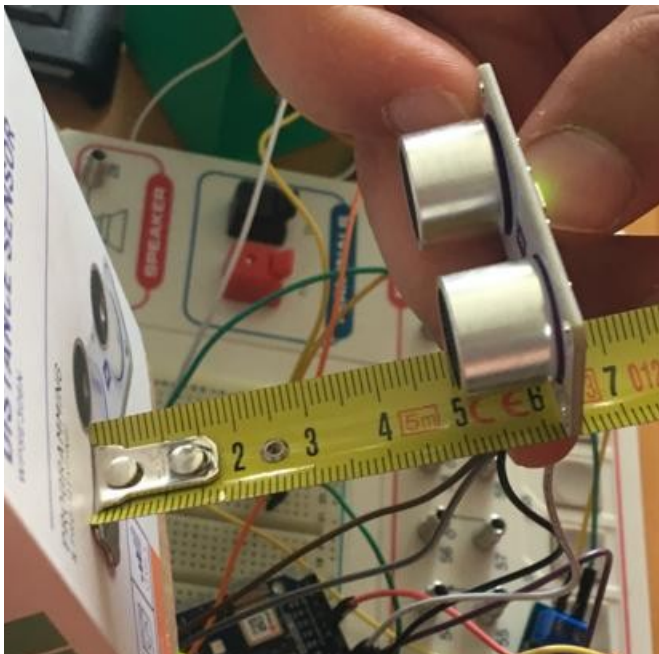
/* 3. Calcul la distance à partir du temps mesuré */
float distance_mm = measure / 2.0 * SOUND_SPEED;

/* Affiche les résultats en mm, cm et m */
Serial.print(F("Distance: "));
Serial.print(distance_mm);
Serial.print(F("mm ("));
Serial.print(distance_mm / 10.0, 2);
Serial.print(F("cm, "));
Serial.print(distance_mm / 1000.0, 2);
Serial.println(F("m)"));

/* Délai d'attente pour éviter d'afficher trop de résultats à la seconde */
delay(500);
}

```

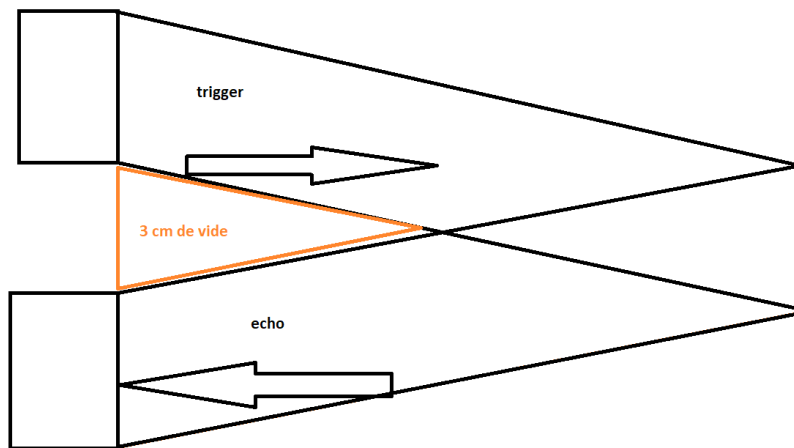
voilà ce que ça donne, nous avons pas précisément 5cm car je ne tenait probablement pas la boîte de manière perpendiculaire au champs de vision du capteurs.



soucis

le soucis c'est qu'en dessous de 3 cm le capteur indique 556 cm. C'est pour un raison très simple. Le constructeur donne dans la data sheet , une plage de distance qui peut être capté. Cela est du au fait qu'il faut une certaine distance pour avoir un écho. Si je me ma main précisément devant les petits tubes. Rien ne peut être envoyé. Et donc rien ne peut être reçu.

Petit schéma :



Solution

faire une condition qui ignore les valeur inférieur à 3 cm.

Projet

grâce à ce capteur ultra son, j'ai pu émettre la condition que si on était inférieur à 40 cm, la voiture recule et tourne à gauche. Le seul couac la dedans c'est que les obstacles sont détecté de manière rectiligne. Du coups si j'ai un obstacle devant la voiture mais qui ne passe pas dans le champs de vision des capteurs. Il fonce dedans.

Donc une solution est de bricoler deux prismes concaves pour pouvoir crée une vision périphérique. À l'aide de plusieurs capteurs.comme le fait bien la vue humaine.

