

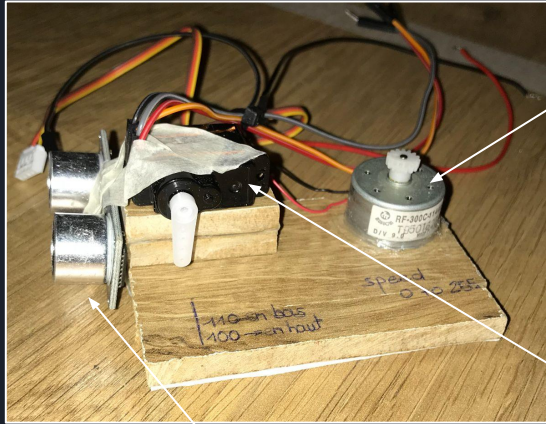


AUTONOMAX

Rapport du projet de NSI

Paolo Fersino
29/05/2021

Le prototype



Moteur à courant continu

Servomoteur

Capteur de distance

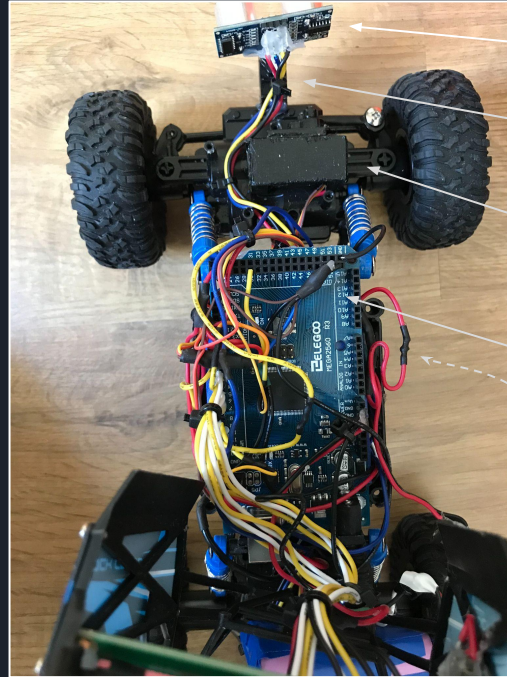
Le prototype a servi à:

- régler les angles du servomoteur
- régler la vitesse du moteur DC
- faire des test avec un code basique

Construction de la voiture - Partie électronique



Avant



Après

Capteur de distance

Servomoteur(du capteur de distance)

Servomoteur(roues)

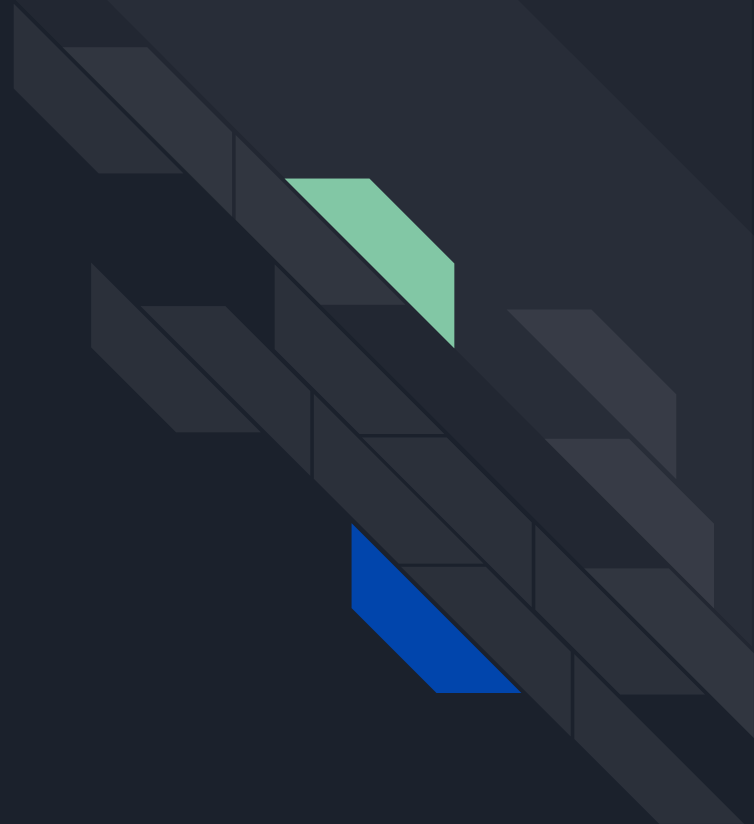
Carte Arduino Mega

LEGENDE CÂBLES:

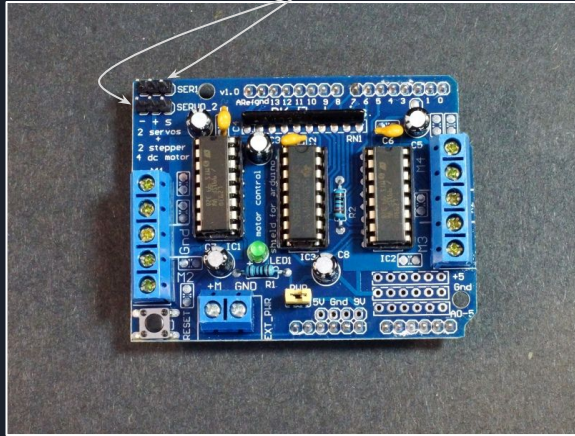
-  5v(+)
-  Ground(-)
-   Data
- 

*Tous les câbles sont isolés avec des gaines thermorétractables

PROBLÈMES



Problème de puissance carte moteurs

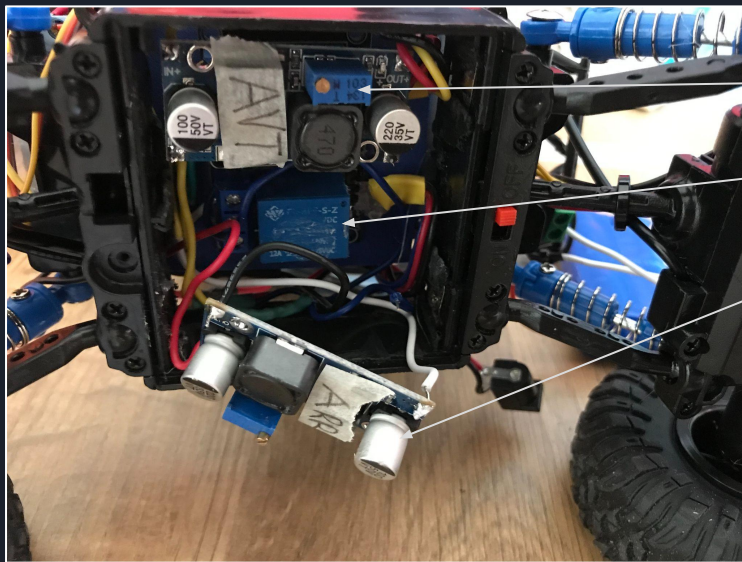


motor control shield

- La carte ne fournissait pas assez de couple aux moteurs.
- La tension du courant était bonne mais l'intensité était insuffisante.
- Les servomoteurs marchaient

Problème de puissance carte moteurs

Solution

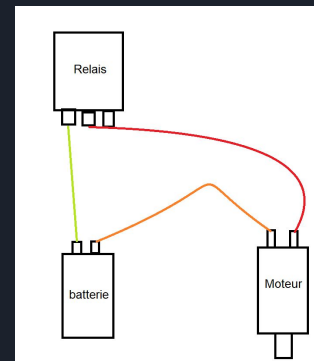
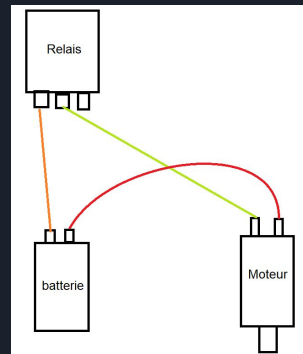


Vue du dessous de la voiture

Régulateur de tension moteur avant

Relais 1

Régulateur de tension moteur arrière



Problème de puissance Arduino

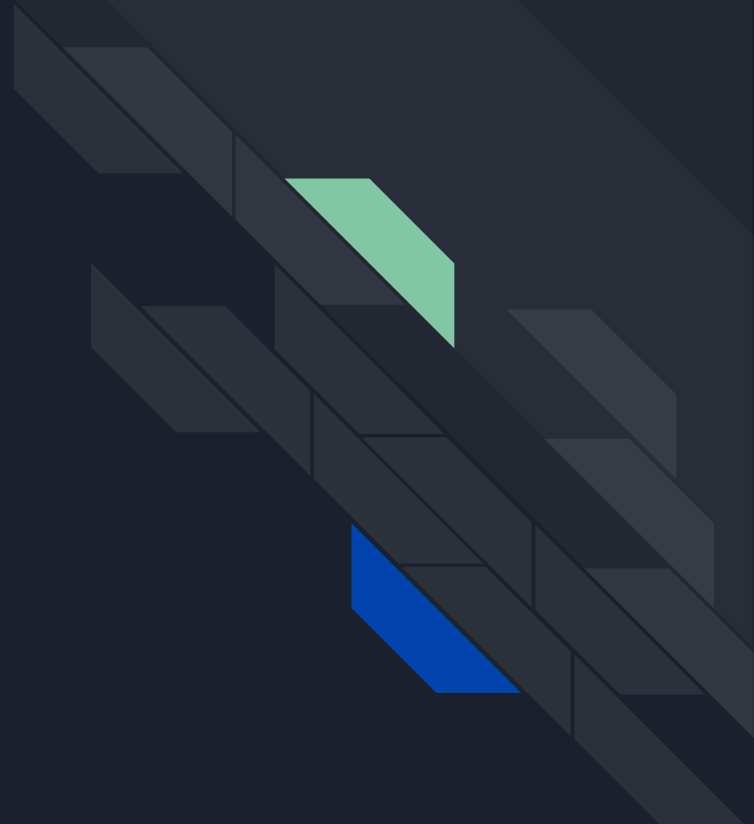


Avant



Après

ANALYSE DU CODE





Ajout des bibliothèques

```
#include <LiquidCrystal.h>
#include <Servo.h>
LiquidCrystal lcd(1, 2, 4, 5, 6, 7);
Servo myservo;
Servo myservo1;
Servo myservo2;
int x; //la valeur de la distance avant
int y; //la valeur de la distance arrière

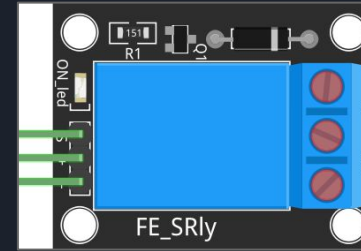
int hcsr04_getUltrasonicData(uint8_t trigPin, uint8_t echoPin, uint8_t data) {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    float duration = pulseIn(echoPin, HIGH);
    if (data == 0) {
        return duration*0.034/2.;
    } else if (data == 1) {
        return duration;
    } else {
        Serial.println("hcsr04_getDistance: invalid option '" + String(data) + "'");
    }
}
```

Le capteur à Ultrasons et le relai



La broche **Trig** est utilisée pour déclencher les impulsions sonores ultrasoniques

La broche **Echo** produit une impulsion lorsque le signal réfléchi est reçu.
(proportionnel à la distance à laquelle se trouve l'objet)

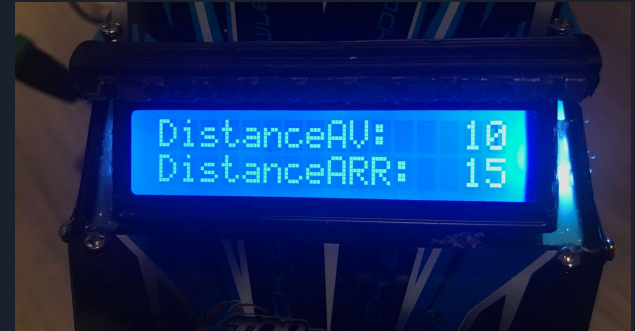


Il y a généralement 3 sorties:

- Naturellement Ouvert
- Commun
- Naturellement Fermé

Fonctions vides

```
void avancer() {  
    digitalWrite(11, LOW);  
    digitalWrite(0, HIGH);  
}  
  
void Stop() {  
    digitalWrite(0, LOW);  
    digitalWrite(11, LOW);  
}  
  
void reculer() {  
    digitalWrite(0, LOW);  
    digitalWrite(11, HIGH);  
    delay(700);  
}  
  
void tourner_gauche() {  
    myservo.write(140);  
}  
  
void tourner_droite() {  
    myservo.write(60);  
}  
  
void afficher_distance() {  
    lcd.setCursor(0,0);  
    lcd.print("DistanceAV:");  
    lcd.setCursor(14,0);  
    lcd.print(x);  
    lcd.setCursor(0,1);  
    lcd.print("DistanceARR:");  
    lcd.setCursor(14,1);  
    lcd.print(y);  
}
```



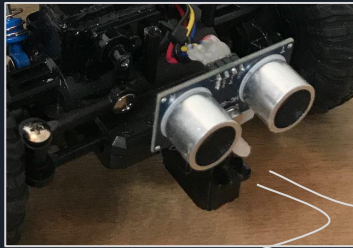
La partie centrale

setup = s'exécute au démarrage

loop = s'exécute indéfiniment

```
void setup() {  
  lcd.begin(16,2);  
  
  pinMode(9, INPUT); //echo  
  pinMode(10, OUTPUT); //trig  
  
  pinMode(12, INPUT); //echo  
  pinMode(13, OUTPUT); //trig  
  
  pinMode(0, OUTPUT); //moteur(relai)avant  
  pinMode(11, OUTPUT); //moteur(relai)arriere  
  
  myservo.attach(3);  
  myservol.attach(8);  
  myservo2.attach(30);  
  myservo.write(100);  
  myservol.write(35);  
  myservo2.write(105);  
}  
  
void loop() {  
  digitalWrite(0, HIGH);  
  myservo.write(100);  
  myservol.write(35);  
  myservo2.write(105);  
  x = hcsr04_getUltrasonicData(10, 9, 0);  
  y = hcsr04_getUltrasonicData(13, 12, 0);  
  afficher_distance();  
  
  if (x <= 30 ) {  
    Stop();  
    delay(500);  
    for (int count = 0; count < 1; count++) { //repete une fois  
      tourner_gauche();  
      regarder_gauche();  
      reculer();  
      delay(700);  
      Stop();  
      delay(500);  
    }  
  }  
  
  if(y < 10 && x <= 30){  
    Stop();  
  }  
}
```

AMÉLIORATIONS EN COURS



servomoteur avant

Les modifications

```
#include <NewPing.h>
#define TRIG_PIN 10
#define ECHO_PIN 9

NewPing sonar(TRIG_PIN, ECHO_PIN);

int lire_distance() {
    delay(100);
    int cm = sonar.ping_cm();
    if(cm==0)
    {
        cm = 250;
    }
    return cm;
}
```

```
int lookRight()
{
    myservo2.write(60);
    delay(500);
    int distance = lire_distance();
    delay(100);
    myservo2.write(105);
    return distance;
}

int lookLeft()
{
    myservo2.write(160);
    delay(500);
    int distance = lire_distance();
    delay(100);
    myservo2.write(105);
    return distance;
    delay(100);
}
```

```
void loop() {
    int distanceD = 0;
    int distanceG = 0;
    delay(40);

    if(distance<=25)
    {
        Stop();
        delay(1000);
        reculer();
        delay(700);
        avancer();
        delay(200);
        distanceD = lookRight();
        delay(200);
        distanceG = lookLeft();
        delay(200);

        if (distanceD>=distanceG)
        {
            tourner_droite();
            avancer();
            delay(500);
        }
        else
        {
            tourner_gauche();
            Stop();
        }
    }

    else
    {
        avancer();
        myservo.write(100);
    }
    distance = lire_distance();
}
```

Conclusion



Batteries

servomoteurs

Ecran LCD

capteurs de distance

