Projet semaine 1

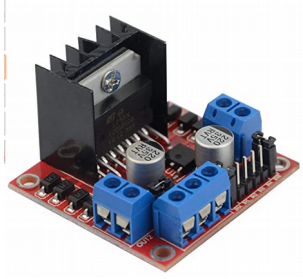
Projet de la semaine voiture autonome anti-collision.

Cette semaine j'ai découvert la ***conception*** le ***montage*** et la ***programmation*** d'une voiture autonome

J'ai donc suivi certaines étapes seul et avec mon équipe pour arriver à finaliser le projet.

J'ai tout d'abord commencé par suivre la doc sur la carte programmable ***Arduino*** que je ne connaissais pas, j'avais par le passé déjà utilisé un Raspberry mais l'Arduino m'était inconnu.

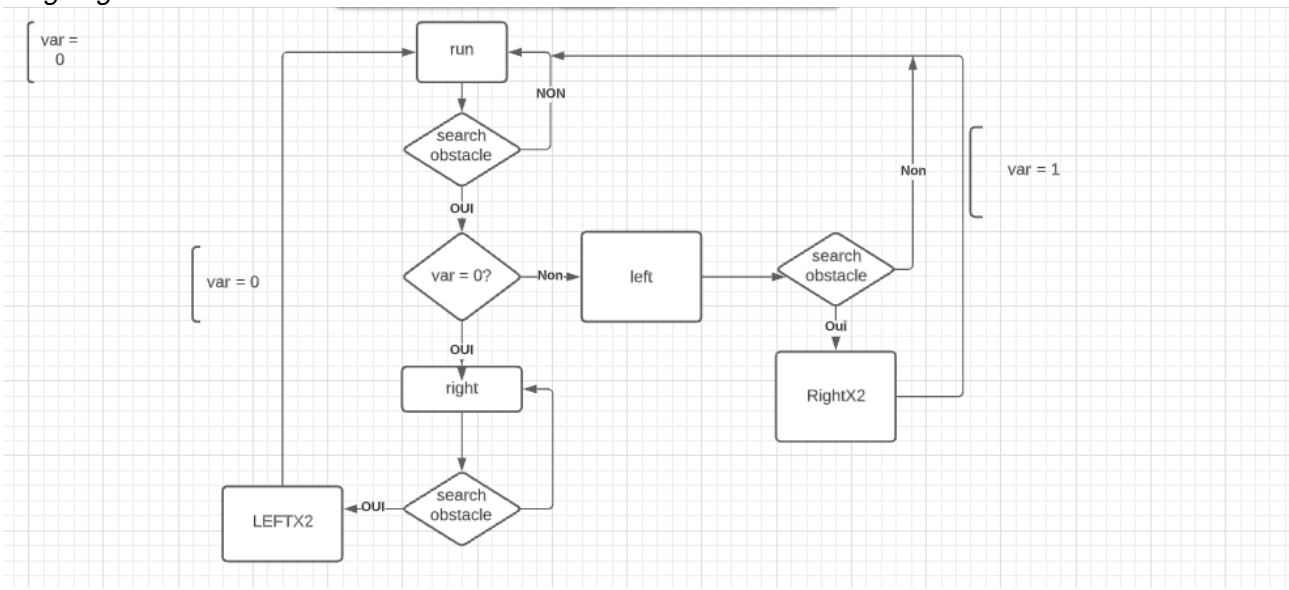
Tout cela suivie de documentation sur le ***radar de recul a ultrason,*** avec pour commencer un code qui envoyer Hello World avec des LED.



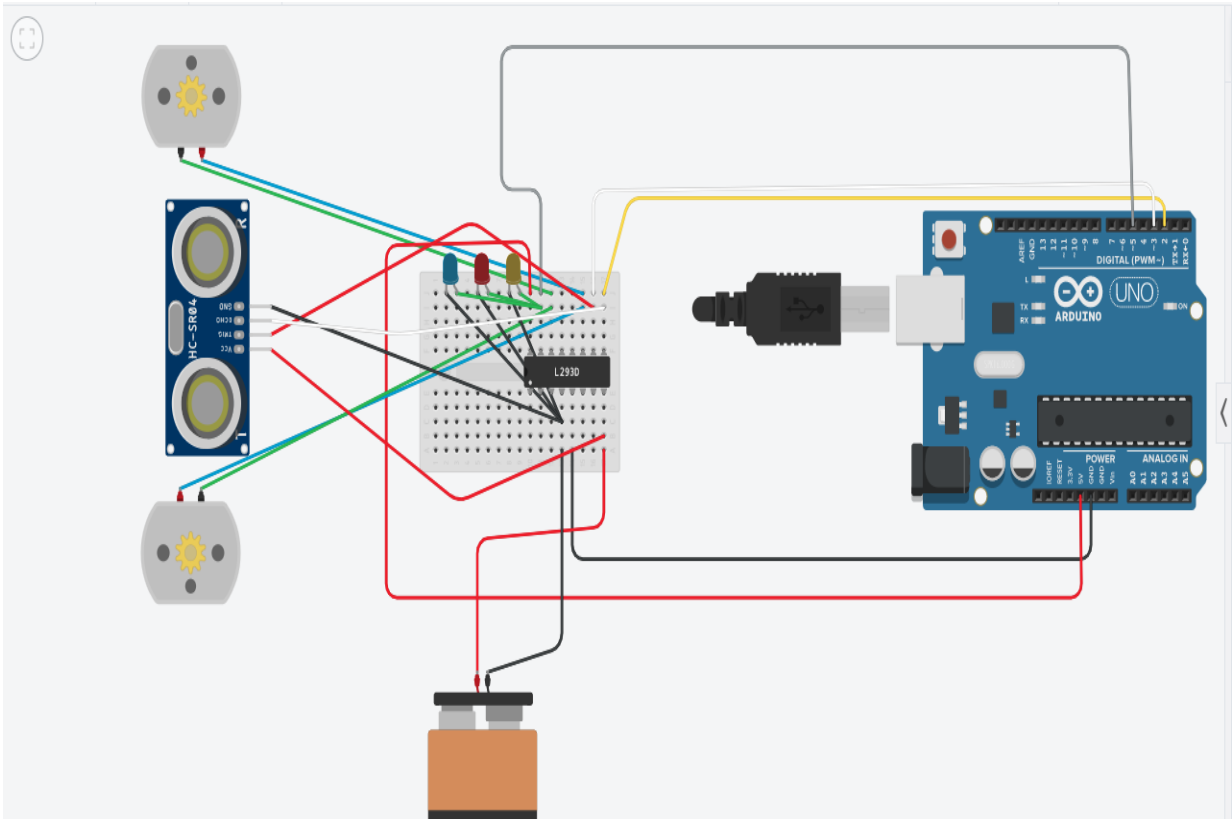
J'ai ensuite appris l'utilité d'un ***Shield LN298N*** et les liens qu'il aurait avec notre voiture
comme faire changer le sens de rotation des moteurs et dissiper la chaleur.

Nous avons ensuite en équipes réfléchies à la façon comment nous allions ***programmer*** notre voiture pour surmonter les obstacles, ensuite nous avons pu réaliser un ***organigramme*** du processus avec ***Luicidchart.***

Organigramme

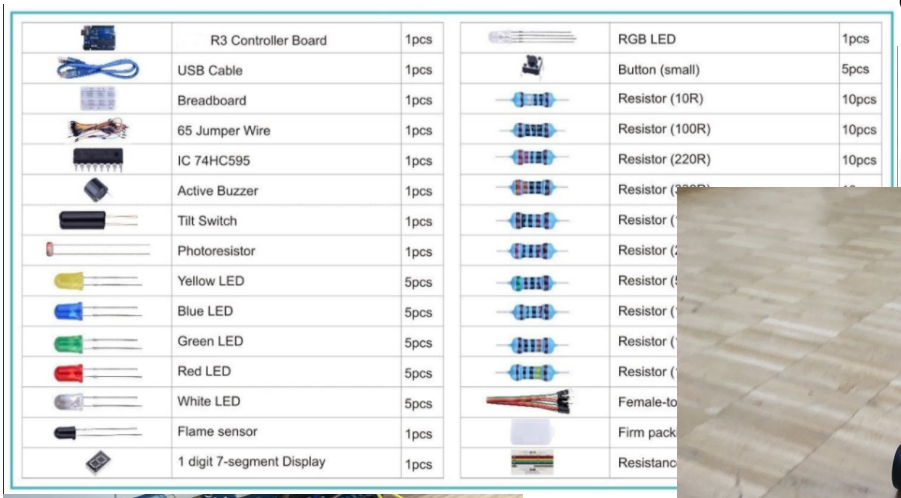


Nous avons ensuite mis en applications ce que nous avions appris dans ***Tinkercad*** afin de vérifier que les branchements et les bouts de codes soient cohérents.



Puis nous avons enfin après la découverte de tous les composants de la voiture pu la monter et ensuite commencer à tester la voiture avec des bouts de code.

Composants et chassie



	R3 Controller Board	1pcs		RGB LED	1pcs
	USB Cable	1pcs		Button (small)	5pcs
	Breadboard	1pcs		Resistor (10R)	10pcs
	65 Jumper Wire	1pcs		Resistor (100R)	10pcs
	IC 74HC595	1pcs		Resistor (220R)	10pcs
	Active Buzzer	1pcs		Resistor (330R)	
	Tilt Switch	1pcs		Resistor (1K)	
	Photoresistor	1pcs		Resistor (2K)	
	Yellow LED	5pcs		Resistor (5K)	
	Blue LED	5pcs		Resistor (10K)	
	Green LED	5pcs		Resistor (1M)	
	Red LED	5pcs		Resistor (1)	
	White LED	5pcs		Female-to	
	Flame sensor	1pcs		Firm pack	
	1 digit 7-segment Display	1pcs		Resistanc	

Transfert du code

phase de test

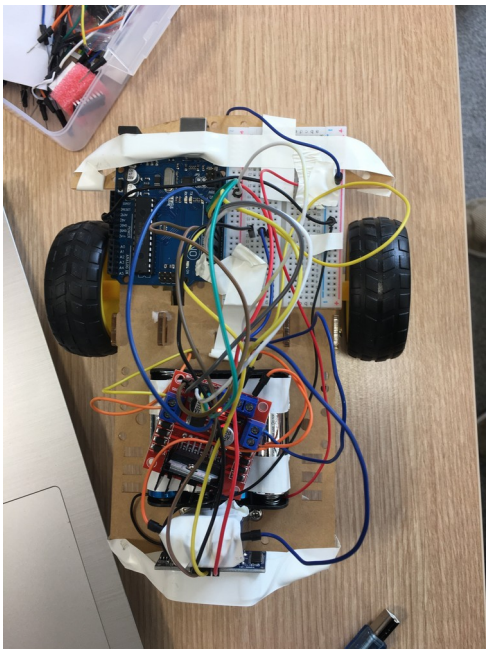


Photo du dessus

Phase de test:

-Faire tourner un moteur

-Faire fonctionner les deux moteurs

- Avancer et reculer

Démontage et remontage de la partie pilotage de la voiture après un bug qui s'avérera par la suite être un problème de piles.

Changement de Pièces:

-Câbles
-Shield
-Carte Arduino
-Piles

Reprise des phases de Test:

-Faire tourner les moteurs dans le sens opposé afin de faire tourner la voiture

-Tester le capteur de distance et le faire fonctionner

-Programmer la voiture pour quel effectue un carré et qu'elle se stoppe lors d'un obstacle rencontré et reparte une fois l'obstacle retiré en continuant la suite du carré.

Dernier jour :

Le dernier jour au matin nous avons pu faire un récapitulatif des étapes de la semaines rédiger le retour d'expérience et revérifier les fonction nécessaire aux épreuves prévu

-Mise en place de la fonction permettant à la voiture d'effectuer l'épreuve du tir au but
-Recalibrage des moteurs
-Vérification des visseries et branchements
-Re lecture du code

Conclusion :

N'ayant pas fait d'électronique ni de langage C avant j'ai pu m'appuyer sur les nombreuses ressources mises à notre disposition et sur les connaissances des membres de mon équipe qui ont su prendre le temps de m'expliquer différentes parties de notre conception .

Cependant j'ai pu constater que j'ai manqué de temps pour assimiler la partie électronique mais je compte revenir sereinement dessus par la suite pour combler mon manque de connaissances.

En conclusion ce fut une très bonne semaine enrichissante où j'ai appris de nouvelles choses je suis content d'avoir pu travailler en équipe avec des coéquipiers compétents dans le domaine imposé et à l'écoute.
Le projet de la semaine est très enrichissant, amusant et la compétition apporter le petit challenge en plus pour la motivation.

Nicolas Turck

Code Tir au But;

```
// connect motor controller pins to Arduino digital pins
// motor one
```

```
int enA = 10;
int in1 = 9;
int in2 = 8;
// motor two
int enB = 5;
int in3 = 2;
int in4 = 3;

const int Echo_in_pin = 6;
const int Trig_out_pin = 7;

const double Speed_sound = 340.0e-3; // Vitesse du son (mm/us)

float Dist_mm=0.0;
int cmpt = 0;

void setup()
{
  // set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  pinMode(Trig_out_pin, OUTPUT);
  pinMode(Echo_in_pin, INPUT);
  digitalWrite(Trig_out_pin, LOW);

  // Init interface série
  Serial.begin(115200);

  pinMode(13, OUTPUT);
}

void avancer()
{
    // this function will run the motors in both directions at a fixed speed
    // turn on motor A
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    // set speed to 200 out of possible range 0~255
    analogWrite(enA, 200);
    // turn on motor B
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    // set speed to 200 out of possible range 0~255
    analogWrite(enB, 180);
    digitalWrite(13, HIGH);
}

void arriere()
{
    // this function will run the motors in both directions at a fixed speed
    // turn on motor A
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    // set speed to 200 out of possible range 0~255
    analogWrite(enA, 200);
    // turn on motor B
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    // set speed to 200 out of possible range 0~255
    analogWrite(enB, 200);
}
```

```
void tournerG()
{
     // this function will run the motors in both directions at a fixed speed
     // turn on motor A
     digitalWrite(in1, HIGH);
     digitalWrite(in2, LOW);
     // set speed to 200 out of possible range 0~255
     analogWrite(enA, 120);
     // turn on motor B
     digitalWrite(in3, HIGH);
     digitalWrite(in4, LOW);
     // set speed to 200 out of possible range 0~255
     analogWrite(enB, 122);
     digitalWrite(13, LOW);
}

void tournerD()
{
     // this function will run the motors in both directions at a fixed speed
     // turn on motor A
     digitalWrite(in1, LOW);
     digitalWrite(in2, HIGH);
     // set speed to 200 out of possible range 0~255
     analogWrite(enA, 180);
     // turn on motor B
     digitalWrite(in3, LOW);
     digitalWrite(in4, HIGH);
     // set speed to 200 out of possible range 0~255
     analogWrite(enB, 198);
     digitalWrite(13, LOW);
}

void arret()
{
  // turn off motor A and B
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  digitalWrite(13, LOW);
}

float GetDisCM(int Echo_in_pinn, int Trig_out_pinn, double Speed_soundd)
  {
    float Dist_milm=0.0;
    long HC_val=0;

    // Déclanchement mesure avec l'envoie d'une impulsion de 10µs
    digitalWrite(Trig_out_pin, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig_out_pin, LOW);

    // Lecture du temps aller/retour de l'onde
    HC_val = pulseIn(Echo_in_pin, HIGH, 30000);

    // Calcul de distance d(mm), V=d/t==> v*t
    // Le coefficnet 2 pour la distance aller
    Dist_milm = (HC_val/2.0) * Speed_sound;

    return Dist_milm;
  }

void loop()
{
```

```
  avancer();
}
```

Fin code tir au but.

Code Labyrinthe :

```
// connect motor controller pins to Arduino digital pins
// motor one
int enA = 10;
int in1 = 9;
int in2 = 8;
// motor two
int enB = 5;
int in3 = 2;
int in4 = 3;

const int Echo_in_pin = 6;
const int Trig_out_pin = 7;

const double Speed_sound = 340.0e-3; // Vitesse du son (mm/us)

float Dist_mm=0.0;
int cmpt = 1;

void setup()
{
  // set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  pinMode(Trig_out_pin, OUTPUT);
  pinMode(Echo_in_pin, INPUT);
  digitalWrite(Trig_out_pin, LOW);

  // Init interface série
  Serial.begin(115200);

  pinMode(13, OUTPUT);
}

void avancer()
{
    // this function will run the motors in both directions at a fixed speed
    // turn on motor A
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    // set speed to 200 out of possible range 0~255
    analogWrite(enA, 150); //200
    // turn on motor B
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    // set speed to 200 out of possible range 0~255
    analogWrite(enB, 145); //190
    digitalWrite(13, HIGH);
}

void arriere()
{
    // this function will run the motors in both directions at a fixed speed
    // turn on motor A
```

```
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    // set speed to 200 out of possible range 0~255
    analogWrite(enA, 200);
    // turn on motor B
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    // set speed to 200 out of possible range 0~255
    analogWrite(enB, 200);
}

void tournerG()
{
    // this function will run the motors in both directions at a fixed speed
    // turn on motor A
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    // set speed to 200 out of possible range 0~255
    analogWrite(enA, 120);
    // turn on motor B
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    // set speed to 200 out of possible range 0~255
    analogWrite(enB, 122);
    digitalWrite(13, LOW);
}

void tournerD()
{
    // this function will run the motors in both directions at a fixed speed
    // turn on motor A
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    // set speed to 200 out of possible range 0~255
    analogWrite(enA, 180);
    // turn on motor B
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    // set speed to 200 out of possible range 0~255
    analogWrite(enB, 198);
    digitalWrite(13, LOW);
}

void arret()
{
  // turn off motor A and B
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  digitalWrite(13, LOW);
}

float GetDisCM(int Echo_in_pinn, int Trig_out_pinn, double Speed_soundd)
  {
    float Dist_milm=0.0;
    long HC_val=0;

    // Déclanchement mesure avec l'envoie d'une impulsion de 10µs
    digitalWrite(Trig_out_pin, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig_out_pin, LOW);

    // Lecture du temps aller/retour de l'onde
    HC_val = pulseIn(Echo_in_pin, HIGH, 30000);
```

```
    // Calcul de distance d(mm), V=d/t==> v*t
    // Le coefficnet 2 pour la distance aller
    Dist_milm = (HC_val/2.0) * Speed_sound;

    return Dist_milm;
  }

void loop()
{
  /* Mesure de la distance */
  Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound); // Valeur brute
  //Dist_mm=GetMean(8,  Echo_in_pin,  Trig_out_pin,  Speed_sound); // Valeur moyenne

  if(Dist_mm == 0 || Dist_mm > 350)//350
  {
    avancer();
    delay(200);
    Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound); // Valeur brute
  }
  else
  {
    arret();
    delay(400);
    if(cmpt==0)
    {
      tournerD();
      delay(300);
      arret();
      cmpt = 1;
      Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
    }
    else
    {
      tournerG();
      delay(300);
      arret();
      delay(200);
      cmpt = 0;
      Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
    }
    delay(500);
    Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound); // Valeur brute
    if(Dist_mm < 350)//350
    {
      if(cmpt == 0)
      {
        tournerD();
        delay(250);
        Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
        arret();
        delay(300);
        Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
        tournerD();
        delay(250);
        Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
        arret();
        delay(200);
        Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
        cmpt = 1;
      }
      else
      {
        tournerG();
```

```
          delay(400);
          Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
          arret();
          delay(300);
          Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
          tournerG();
          delay(300);
          Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
          arret();
          delay(200);
          Dist_mm=GetDisCM( Echo_in_pin,  Trig_out_pin,  Speed_sound);
          cmpt = 0;
        }
      }
    }
}
```