Arthur | ***Rapport de séance n°5*** | Projet Vitis

25/11/22

Objectifs de la séance :

- Faire en sorte que le moteur avance et recule avec un capteur ultrason

Réalisations :

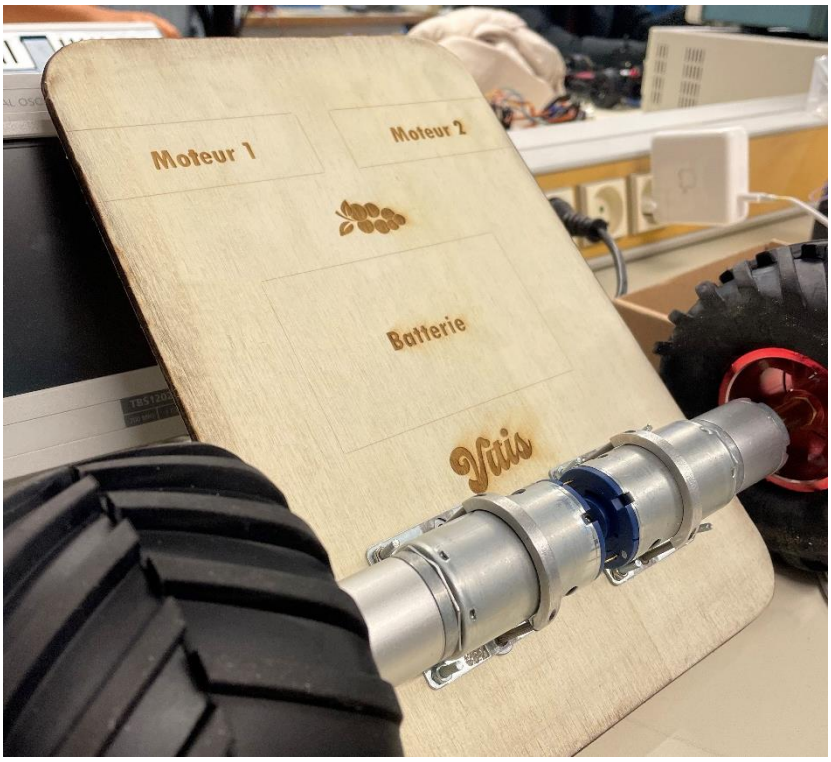
- L'objectif d'aujourd'hui était clairement de faire rouler notre Robot. Nous avions un châssis avec 2 moteurs (voir 1) et un bout de code pour le capteur ultrason.

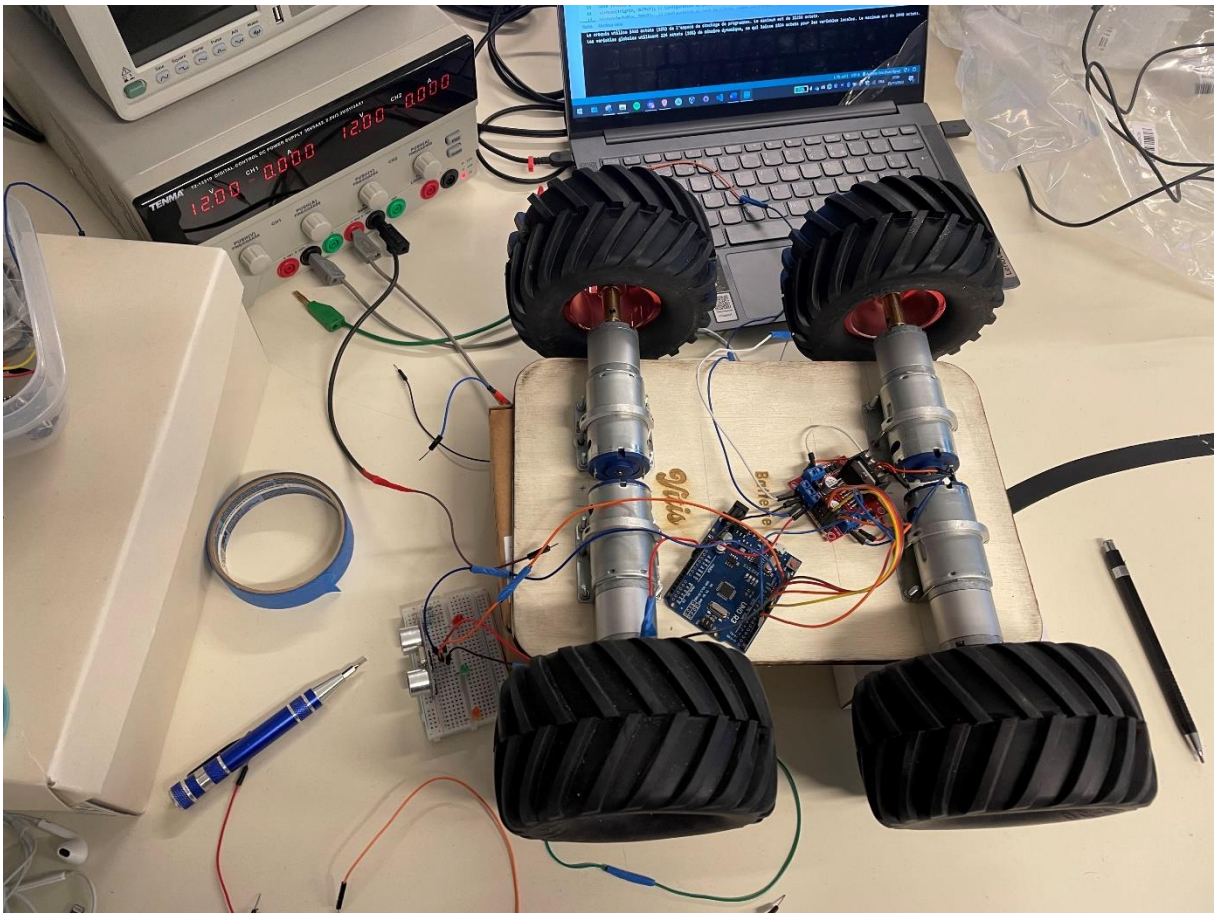
  Dans un premier temps j'ai fait un code arduino pour faire tourner une roue en avant et arrière. Je branche l'Arduino au moteur et tout fonctionne, le moteur tourne dans le sens horaire 2 sec, s'arrête puis tourne dans le sens anti-horaire 1 sec.

  Une fois que le moteur de la partie A fonctionne, je fais le même code pour le moteur de la partie B de notre carte pont en H (Module L298N). Cependant cette fois ci il y un problème. Lorsque le programme est en route, le moteur A tourne pendant 2 sec dans le sens horaire puis le moteur B tourne pendant 1 seconde dans l'autre sens.
  Or ce n'est pas ce qui est voulu, le problème ne venait pas du code ni de la source d'alimentation mais bien du module L298N. En effet sa plage de tension effective est entre 0 et 5V, au-delà ou en deçà la carte ne comprend plus.
  Et avec l'utilisation d'un multimètre on s'est rendu compte que le fait de connecter toutes les alimentations et masses sur la même breadboard faisait que le signal délivrer par l'arduino sur certain PIN dépasse les 5V.
  Nous avons résolu le problème en séparant les 2 circuits, désormais les moteurs peuvent rouler ensemble.

  Dernière étape on connecte les sources A et B sur la breadboard pour connecter les 4 roues ensemble. Les 2 roues de bâbord représentent la partie A du module et les 2 autres la partie B.

  Voici la vidéo des premiers pas du robot : https://youtu.be/6eqNRxpRVlw





Et voici le code Arduino qui fait tourner les 4 roues :

```ino
// Variables utiles
long duree;         // durée de l'echo
int distance;       // distance
int V_avant = 22;   // Valeur à partir de laquelle le robot avance
int V_arriere = 18; // Valeur à partir de laquelle le robot recul

#define borneENA 10 // On associe la borne "ENA" du L298N à la pin D10 de l'arduino
#define borneIN1 9  // On associe la borne "IN1" du L298N à la pin D9 de l'arduino
#define borneIN2 8  // On associe la borne "IN2" du L298N à la pin D8 de l'arduino
#define borneIN3 7  // On associe la borne "IN3" du L298N à la pin D7 de l'arduino
#define borneIN4 6  // On associe la borne "IN4" du L298N à la pin D6 de l'arduino
#define borneENB 5  // On associe la borne "ENB" du L298N à la pin D5 de l'arduino
#define echoPin 12  // Echo    (réception)
#define trigPin 11  // Trigger (emission)

void setup()
{
    pinMode(trigPin, OUTPUT); // Configuration du port du Trigger comme une SORTIE
    pinMode(echoPin, INPUT);  // Configuration du port de l'Echo  comme une ENTREE

    // Configuration de toutes les pins de l'Arduino en "sortie" (car elles attaquent les entrées du module L298N)
    pinMode(borneENA, OUTPUT);
    pinMode(borneIN1, OUTPUT);
    pinMode(borneIN2, OUTPUT);
    pinMode(borneIN3, OUTPUT);
    pinMode(borneIN4, OUTPUT);
    pinMode(borneENB, OUTPUT);

    Serial.begin(9600); // Démarrage de la communication série
}

void loop()
{

    // Émission d'un signal de durée 10 microsecondes
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Écoute de l'écho
    duree = pulseIn(echoPin, HIGH);

    // Calcul de la distance
    distance = duree * 0.034 / 2;

    // Affichage de la distance dans le Moniteur Série
    Serial.print("Distance : ");
    Serial.print(distance);
    Serial.println("cm");

    if (distance > V_avant)
    {
        avant();
        delay(100);
        lancerRotationMoteurPont();
        Serial.println("avant");
```

```
    }
    else if (distance < V_arriere)
    {
        arriere();
        delay(100);
        lancerRotationMoteurPont();
        Serial.println("arriere");
    }
    else
    {
        stop_moteur();
        Serial.println("stop");
    }
}

void avant()
{
    // Configuration du L298N en "marche avant", pour le moteur connecté au pont B. Selon sa table de vérité, il faut que :
    digitalWrite(borneIN3, HIGH); // L'entrée IN3 doit être au niveau haut
    digitalWrite(borneIN4, LOW);  // L'entrée IN4 doit être au niveau bas

    // Configuration du L298N en "marche avant", pour le moteur connecté au pont A. Selon sa table de vérité, il faut que :
    digitalWrite(borneIN1, HIGH); // L'entrée IN1 doit être au niveau haut
    digitalWrite(borneIN2, LOW);  // L'entrée IN2 doit être au niveau bas
}

void arriere()
{
    // Puis on configure le L298N en "marche arrière", pour le moteur câblé sur le pont B. Selon sa table de vérité, il faut que :
    digitalWrite(borneIN3, LOW);  // L'entrée IN3 doit être au niveau bas
    digitalWrite(borneIN4, HIGH); // L'entrée IN4 doit être au niveau haut

    // Puis on configure le L298N en "marche arrière", pour le moteur câblé sur le pont A. Selon sa table de vérité, il faut que :
    digitalWrite(borneIN1, LOW);  // L'entrée IN1 doit être au niveau bas
    digitalWrite(borneIN2, HIGH); // L'entrée IN2 doit être au niveau haut
}

void lancerRotationMoteurPont()
{
    digitalWrite(borneENB, HIGH);
    digitalWrite(borneENA, HIGH); // Active l'alimentation du moteur 2
}

void stop_moteur()
{
    digitalWrite(borneENB, LOW);
    digitalWrite(borneENA, LOW); // Désactive l'alimentation du moteur 2
}

```