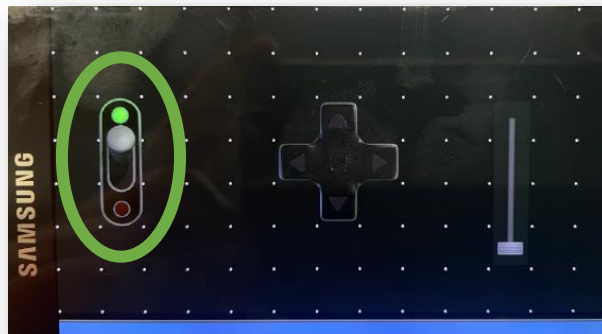


Objectifs de la séance :

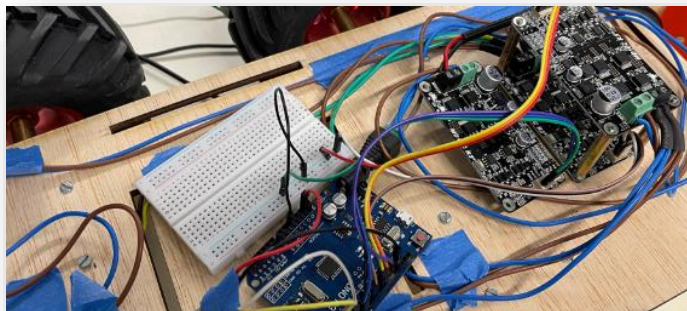
- Changer le pad pour une télécommande à 4 directions
- Contrôler le bras (vérin) avec la télécommande

Réalisations :

- La première partie de la séance est destinée au contrôle du bras. Le bras est un vérin constitué d'un moteur à courant continu que l'on pilote en fonction du sens du courant.  
Pour le contrôler à l'aide du smartphone, on utilise un interrupteur virtuel en vert sur le schéma :



Puis j'ai branché ce moteur à un pont en H, directement connecté à la carte Arduino.



- Ensuite pour diriger le robot, j'ai épuré l'ancien code qui contrôlait le robot avec une télécommande 4 directions. Aussi dans ce code le robot change sa façon de tourner, désormais le robot tourne sur lui-même pour s'orienter en faisant tourner les roues opposées dans des sens contraires.

Or pour palier au problème des moteurs qui passent de 0 à 100% de leurs capacités, j'ai ajouté un slider à la télécommande qui contrôle la vitesse en direct :



Nous avons désormais un robot qui se contrôle entièrement avec le smartphone, bras compris.

## telecommande\_et\_bras.ino

```
#include <SoftwareSerial.h>

SoftwareSerial BT(4, 2); // RX, TX

#define dir_g 8
#define pwm_g 6
#define dir_d 10
#define pwm_d 11
#define dir_b 12
#define pwm_b 5

int speed = 0;

void setup() {
  pinMode(dir_g, OUTPUT);
  pinMode(pwm_g, OUTPUT);
  pinMode(dir_d, OUTPUT);
  pinMode(pwm_d, OUTPUT);
  pinMode(dir_b, OUTPUT);
  pinMode(pwm_b, OUTPUT);

  // Configuration de la liaison Bluetooth
  BT.begin(9600);

  // Initialise les pins à l'état bas
  digitalWrite(dir_g, LOW);
  analogWrite(pwm_g, 0);
  digitalWrite(dir_d, LOW);
  analogWrite(pwm_d, 0);
  digitalWrite(dir_b, LOW);
  analogWrite(pwm_b, 0);
}

void loop() {
  if (BT.available()) {
    char c = BT.read();

    // Si la commande est une commande de direction
    if (c == 'U') {
      digitalWrite(dir_g, HIGH);
      digitalWrite(dir_d, HIGH);
      analogWrite(pwm_g, speed);
      analogWrite(pwm_d, speed);
    } else if (c == 'D') {
      digitalWrite(dir_g, LOW);
      digitalWrite(dir_d, LOW);
      analogWrite(pwm_g, speed);
      analogWrite(pwm_d, speed);
    } else if (c == 'L') {
      digitalWrite(dir_g, LOW);
      digitalWrite(dir_d, HIGH);
      analogWrite(pwm_g, speed);
    }
  }
}
```

```
    analogWrite(pwm_d, speed);

} else if (c == 'R') {
    digitalWrite(dir_g, HIGH);
    digitalWrite(dir_d, LOW);
    analogWrite(pwm_g, speed);
    analogWrite(pwm_d, speed);
}
// Si la commande est une commande de vitesse
else if (c == 'V') {
    speed = BT.parseInt();
    analogWrite(pwm_g, speed);
    analogWrite(pwm_d, speed);

}

else if (c == 'B') {
    digitalWrite(dir_b, LOW);
    analogWrite(pwm_b, 255);
}
else if (c == 'H') {
    digitalWrite(dir_b, HIGH);
    analogWrite(pwm_b, 255);
}

}

}
```