

Moteur CC - Commande de moteur à courant continu

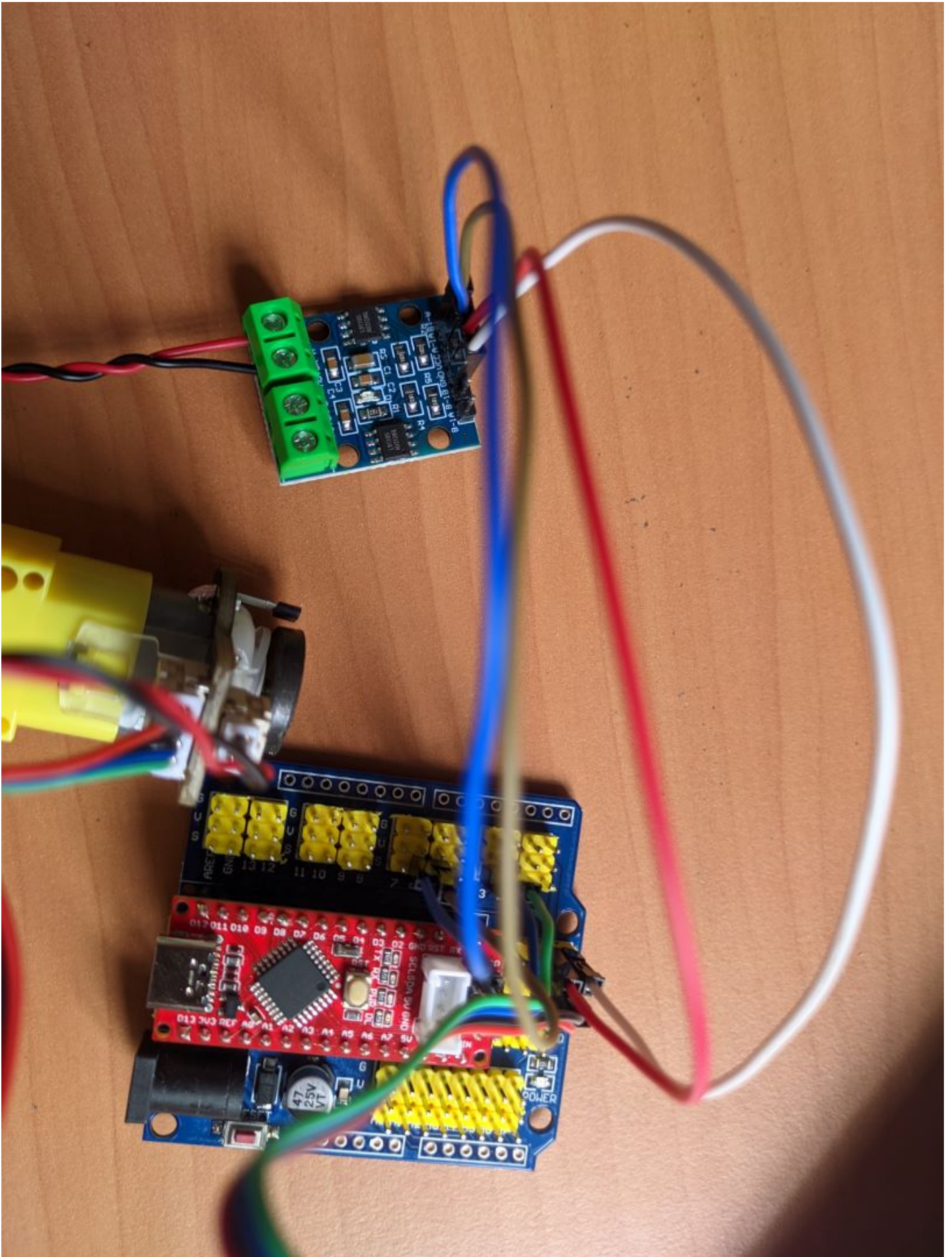
Moteur à courant continu "moteur jaune"

<https://seafire.unistra.fr/d/16f415226d224af5bd8f/>

[\Seafire\Reseau_FabLab_Alsace_Nord\01_Robotique_Educative](#)

Module de contrôle L9110S

Montage :



<https://arduino.blaise-pascal.fr/contrôleur-I9110s/>

```

#define moteurA_1 5
#define moteurA_2 6
#define moteurB_1 10
#define moteurB_2 11
int vitesseA = 255; // 0 à 255
int vitesseB = 255; // 0 à 255
void setup() {
    // Configuration des ports en mode "sortie"
    pinMode(moteurA_1, OUTPUT);
    pinMode(moteurA_2, OUTPUT);
    pinMode(moteurB_1, OUTPUT);
    pinMode(moteurB_2, OUTPUT);
}
void loop() {
    digitalWrite(moteurA_1, LOW);
    analogWrite(moteurA_2, vitesseA);
    digitalWrite(moteurB_1, LOW);
    analogWrite(moteurB_2, vitesseB);
    delay(2000);

    analogWrite(moteurA_1, vitesseA);
    digitalWrite(moteurA_2, LOW);
    analogWrite(moteurB_1, vitesseB);
    digitalWrite(moteurB_2, LOW);

    delay(2000);
    digitalWrite(moteurA_1, LOW);
    digitalWrite(moteurA_2, LOW);
    digitalWrite(moteurB_1, LOW);
    digitalWrite(moteurB_2, LOW);

    delay(3000);
}

```

Module de contrôle L298

Attention le module entraîne une chute de tension d'environ 3V. Donc si vous voulez délivrer 5V à vos moteurs à courant continu il faudra une alimentation de 8V.

```
// Pont en H L298N

//Ports de commande du moteur B
int motorPin1 = 10;
int motorPin2 = 11;
int enablePin = 5;

// Vitesse du moteur
int state = 55;

void setup() {
    // Configuration des ports en mode "sortie"
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(enablePin, OUTPUT);

    // Initialisation du port série
    Serial.begin(9600);
}

void loop() {
    if (Serial.available() > 0)
    {
        // Lecture de l'entier passé au port série
        state = Serial.parseInt();

        //
        // Sens du mouvement
        //
        if (state > 0) // avant
        {
            digitalWrite(motorPin1, HIGH);
            digitalWrite(motorPin2, LOW);
            Serial.print("Avant ");
            Serial.println(state);
        }
        else if (state < 0) // arrière
```

```
{
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    Serial.print(" Arriere ");
    Serial.println(state);
}
else // Stop (freinage)
{
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, HIGH);
    Serial.println("Stop");
}

//
// Vitesse du mouvement
//
analogWrite(enablePin, abs(state));
}
delay(100);
}
```

PWM

<https://arduino.blaisepascal.fr/conversion-numeriqueanalogique-pwm/>

Revision #8

Created 13 March 2024 14:24:41 by admin_idf

Updated 12 June 2024 13:52:52 by admin_idf