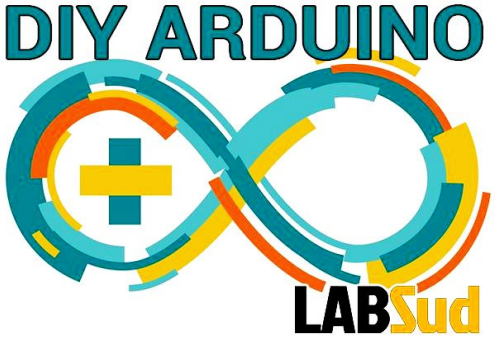
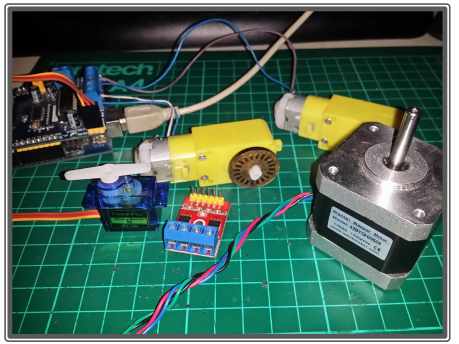
3ème atelier Arduino

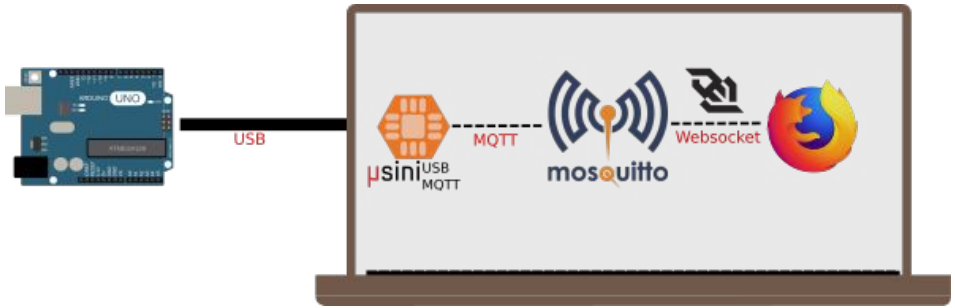


On passe à la pratique !

Au Programme :



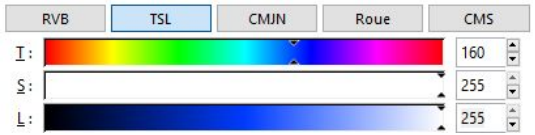
- Strips de leds neopixels (ws2812b)
- Contrôle de moteur
- Mise en oeuvre simple d’un serveur MQTT
- Capteurs

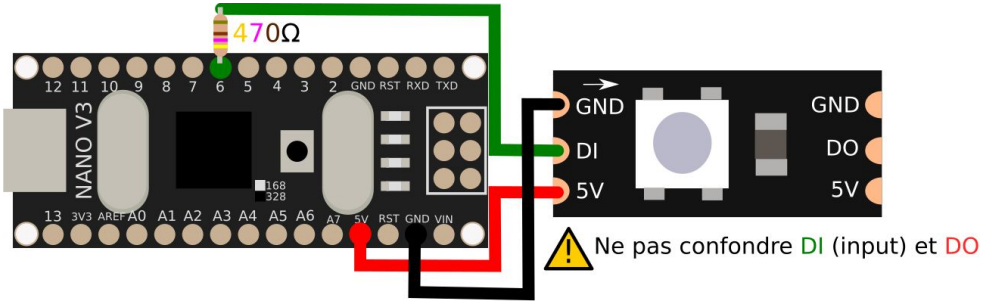


Strip de led neopixel (ws2812b)

Bibliothèque : FastLED

HSV (ou TSL) permet de régler la couleur des leds comme on le ferait dans un logiciel de dessin





leds[numéro_led] = CHSV(Hue, Saturation, Value);

Hue	Teinte (couleur de la led)
Saturation	Intensité (0 Blanc ←→ 255 Couleur)
Value	Luminosité

Diminuer/Augmenter Value fait pulser la led, ce qui donne une impression de “respiration”
Diminuer/Augmenter Hue permet d’avoir un effet Arc-en-Ciel
La Saturation permet de “blanchir” la teinte de la led met n’a que peu d’effet

Strip de led (exemple)

```
#include <FastLED.h>

const int NUM_LEDS = 1; //Nombre de leds
const int LEDS_PIN = 6; //Broche où sont branchés les leds

CRGB leds[NUM_LEDS]; //Tableau des leds

void setup() {
  //Initalisation des leds
  FastLED.addLeds<NEOPIXEL, LEDS_PIN>(leds, NUM_LEDS);
}

void loop() {
  leds[0] = CHSV(192, 255, 255);
  FastLED.show();
  delay(1000);

  leds[0] = CHSV(0, 0, 0);
  FastLED.show();
  delay(1000);
}
```

Limiter la consommation électrique (utile si l'on ne veut pas utiliser d'alimentation externe)

```
FastLED.setMaxPowerInVoltsAndMilliamps(5,10);
                                        V, mA
```

Régler la luminosité (0 à 255)

```
FastLED.setBrightness(255);
```

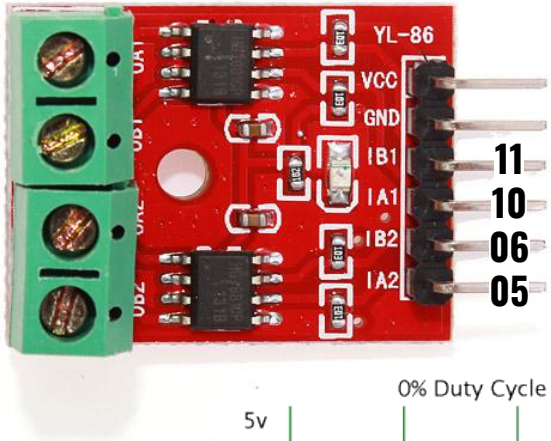
Après avoir changer les leds, on applique les changement avec cette commande:

```
FastLED.show();
```

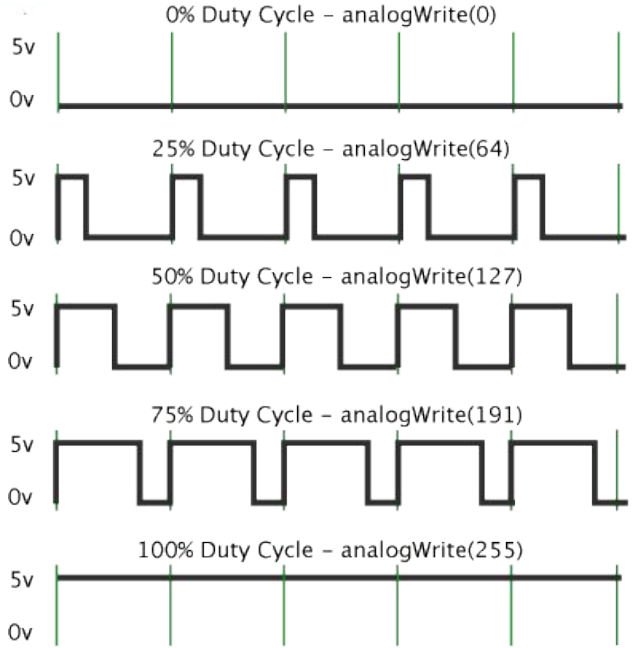
Contrôle de moteur avec un L9110

Moteur 1

Moteur 2



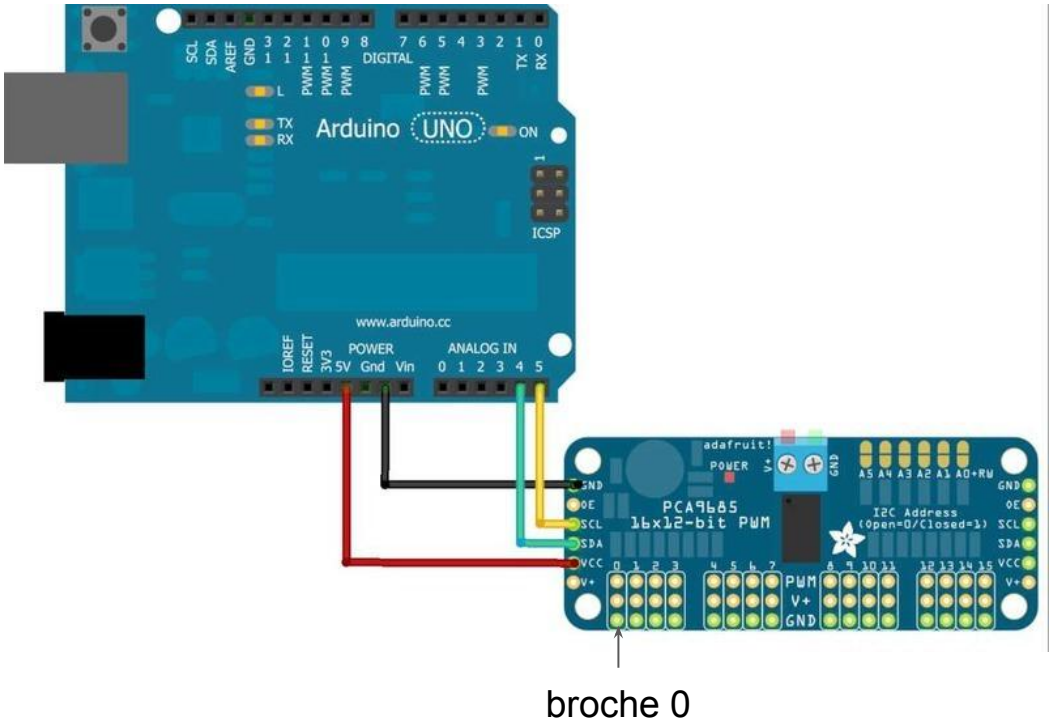

```
void setup() {
  pinMode(5, OUTPUT);      // IA2 moteur 2 en avant
  pinMode(6, OUTPUT);      // IB2 moteur 2 en arrière
  pinMode(10, OUTPUT);     // IA1 moteur 1 en avant
  pinMode(11, OUTPUT);     // IB1 moteur 1 en arrière
 }

void loop() {
  digitalWrite(5, LOW);          // avant d'agir sur un moteur
  digitalWrite(6, LOW);          // il vaut mieux couper les 2 commandes
  delay(100);
 analogWrite(5, 255);    // marche avant moteur 2 vitesse max
  delay(1000);
 }
```



Contrôler des servo moteur avec un PCA9685

Récupérer la bibliothèque :
adafruit PWM Servo Driver

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

// Par défaut utilise l'adresse 0x40
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

void setup() {
  pwm.begin();
  pwm.setPWMFreq(60);  // Les servos analos marche à 60hz
  delay(10);
}

void loop() {
                  // (broche, ? ,position du moteur)
    pwm.setPWM(0, 0, 675); // sg90 max
  delay(750);
    pwm.setPWM(0, 0, 175);          // sg90 min
  delay(750);
}
```



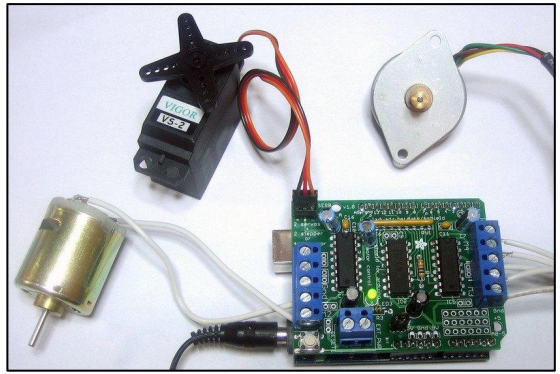
Motor shield

Moteur pas à pas

```
#include <AFMotor.h>

// ( 360/angle par step , M1 et M2 )
AF_Stepper stepper(200, 1);

void setup() {
  stepper.setSpeed(100);  // 100 rpm
  stepper.release();
}

void loop() {
  stepper.step(200, FORWARD, SINGLE);
  delay(1000);
  stepper.step(50, BACKWARD, SINGLE);
  delay(1000);
}
```

Moteur DC

```
#include <AFMotor.h>
AF_DCMotor motor(4); // DC motor on M4
void setup() {
  motor.setSpeed(200); // turn on motor #4
  motor.run(RELEASE);
}
void loop() {
  motor.run(FORWARD);
  motor.setSpeed(255);
  delay(500);
  motor.setSpeed(0);
  delay(500);
  motor.run(BACKWARD);
  motor.setSpeed(255);
  delay(500);
  motor.setSpeed(0);
  delay(500);
  }
```



Servo Moteur

```
#include <Servo.h>

// DC hobby servo
Servo servo1;

void setup() {
  servo1.attach(10); // turn on servo
}

void loop() {
  servo1.write(17); // servo min
  delay(800);
  servo1.write(180); // servo max
  delay(800);
}
```