

# elec code

🕒 Date de création	@21 février 2024 14:31
☑ Relue ?	<input type="checkbox"/>

## ▼ allumer Led

```
int LED = 13;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(2000);
  digitalWrite(LED, LOW);
  delay(2000);
}
```

## ▼ clignoter Led

```
int LED = 3;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
}
```

#### ▼ Allumer RGB

```
int RED = A1;
int GREEN = A2;
int BLUE = A3;

void setup() {
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
}

void loop() {
  analogWrite(RED, 255);
  analogWrite(GREEN, 0);
  analogWrite(BLUE, 255);
}
```

#### ▼ Pull up puis pull down

```
int BTN = 3;
int BTN2 = 4;
int LED = 13;
int LED2 = 12;

int value;
int value2;

void setup() {
  pinMode(BTN, INPUT);
  pinMode(BTN2, INPUT);
  pinMode(LED, OUTPUT);
  pinMode(LED2, OUTPUT);
}
```

```

void loop() {

    value = digitalRead(BTN);
    if (value == HIGH) {
        digitalWrite(LED, LOW);
    } else {
        digitalWrite(LED, HIGH);
    }

    value2 = digitalRead(BTN2);
    if (value2 == HIGH) {
        digitalWrite(LED2, HIGH);
    } else {
        digitalWrite(LED2, LOW);
    }
}

```

▼ joystick et potentiomètre et pull up et pull down

```

int BTN = 3;
int BTN2 = 4;
int POT = A5;
int JoystickX = A1;
int JoystickY = A2;
int LED = 13;
int LED2 = 12;

int valueBTN;
int valueBTN2;
int valuePOT;
int valueJoystickX;
int valueJoystickY;

void setup() {
    Serial.begin(9600);
    pinMode(BTN, INPUT);
    pinMode(BTN2, INPUT);
}

```

```

pinMode(POT, INPUT);
pinMode(JoystickX, INPUT);
pinMode(JoystickY, INPUT);
pinMode(LED, OUTPUT);
pinMode(LED2, OUTPUT);
}

void loop() {
  valueBTN = digitalRead(BTN);
  if (valueBTN == HIGH) {
    digitalWrite(LED, LOW);
  } else {
    digitalWrite(LED, HIGH);
  }

  valueBTN2 = digitalRead(BTN2);
  if (valueBTN2 == HIGH) {
    digitalWrite(LED2, HIGH);
  } else {
    digitalWrite(LED2, LOW);
  }

  valuePOT = analogRead(POT);
  Serial.print("POT value: ");
  Serial.print(valuePOT);

  Serial.print(", ");

  valueJoystickX = analogRead(JoystickX);
  Serial.print("X: ");
  Serial.print(valueJoystickX);

  Serial.print(", ");

  valueJoystickY = analogRead(JoystickY);
  Serial.print("Y: ");
  Serial.print(valueJoystickY);

```

```
Serial.println(".");  
}
```

▼ Faire tourner servo moteur dans un certains angle

```
#include<Servo.h>  
  
Servo servoMotor;  
int pos;  
int startpos = 30;  
int endpos = 90;  
  
void setup() {  
    servoMotor.attach(13);  
    for(pos=0;pos<=30;pos++){  
        servoMotor.write(pos);  
        delay(15);  
    }  
}  
  
void loop(){  
    for(pos=startpos;pos<=endpos;pos++){  
        servoMotor.write(endpos);  
        delay(5);  
    }  
  
    delay(500); // delay between each turns  
  
    for(pos=endpos;pos>=startpos;pos--){  
        servoMotor.write(startpos);  
        delay(5);  
    }  
  
    delay(500); // delay between each turns  
}
```

#### ▼ Tourner moteur step by step

```
#include <Stepper.h>

int in1Pin = 11;
int in2Pin = 10;
int in3Pin = 9;
int in4Pin = 8;
int revolution = 4000;

Stepper myStepper = Stepper(revolution, in4Pin, in3Pin, in2Pin, in1Pin);

void setup() {
    // Set the speed to 5 rpm:
    myStepper.setSpeed(5);
}

void loop() {

    // Step one revolution in one direction:
    myStepper.step(revolution);
    delay(500);

    // Step one revolution in the other direction:
    myStepper.step(-revolution);
    delay(500);
}
```

#### ▼ Faire tourner moteur DC

```
int hbridgeIN1 = 2;
int hbridgeIN2 = 3;

void setup() {
    pinMode(hbridgeIN1, OUTPUT);
    pinMode(hbridgeIN2, OUTPUT);
}
```

```

    pinMode(hbridgeIN2, OUTPUT);
}

void loop() {
    digitalWrite(hbridgeIN1, HIGH);
    delay(1000);
    digitalWrite(hbridgeIN1, LOW);
    digitalWrite(hbridgeIN2, HIGH);
    delay(1000);
    digitalWrite(hbridgeIN2, LOW);
}

```

#### ▼ Capteur de contact

```

int SHOCK_PIN = 12;

void setup() {
    pinMode(LED_BUILTIN, OUTPUT);           // on-board LED, usu
    pinMode(SHOCK_PIN, INPUT);              // shock sensor pin
}

void loop() {
    if (digitalRead(SHOCK_PIN)) {           // shock detected?
        // shock detected with pull-down resistor
        digitalWrite(LED_BUILTIN, HIGH);    // switch LED on
        delay(2000);                        // leave LED on for
    }
    else {
        // shock not detected with pull-down resistor
        digitalWrite(LED_BUILTIN, LOW);     // switch LED off
    }
}

```

#### ▼ Contact via relais

```

int relay = 7;

void setup() {
  pinMode(relay, OUTPUT);
}

void loop() {
  digitalWrite(relay, HIGH);
  delay(1000);
  digitalWrite(relay, LOW);
  delay(1000);
}

```

#### ▼ Capteur de luminosité

```

int PHOTORENSOR = A0;

void setup() {
  Serial.begin(9600);
  pinMode(PHOTORENSOR, INPUT);
}

void loop() {
  int analogValue = analogRead(PHOTORENSOR);

  Serial.print("Analog reading: ");
  Serial.println(analogValue);

  delay(500);
}

```

#### ▼ Capteur de distance

```

int trigPin = 11;
int echoPin = 10;

```



```

float duration_us, distance_cm;

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration_us = pulseIn(echoPin, HIGH);

  distance_cm = 0.017 * duration_us;

  Serial.print("distance: ");
  Serial.print(distance_cm);
  Serial.println(" cm");

  delay(500);
}

```

#### ▼ Capteur infrarouge

```

#include <IRremote.h>
#include <IRremoteInt.h>

int RECV_PIN = 7;          // The digital pin that the si
IRrecv receiver(RECV_PIN); // Create a new receiver obje
decode_results results;    // A varuable that would be u

void setup() {
  Serial.begin(9600);      // Setup serial port to send
  receiver.enableIRIn();   // Enable receiver so that it

```

```
}

void loop() {
    if(receiver.decode(&results)) {           // Decode t
        Serial.println(results.value, HEX);    // Print th
        receiver.resume();                     // Continue
    }
}
```