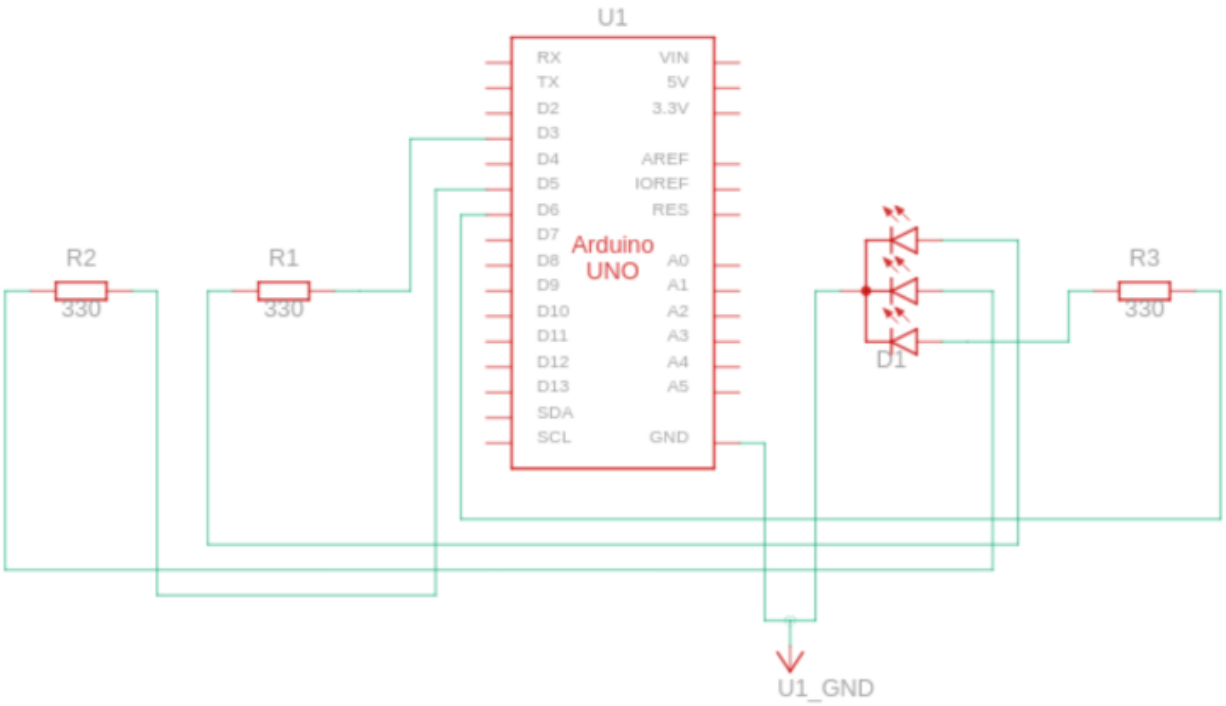
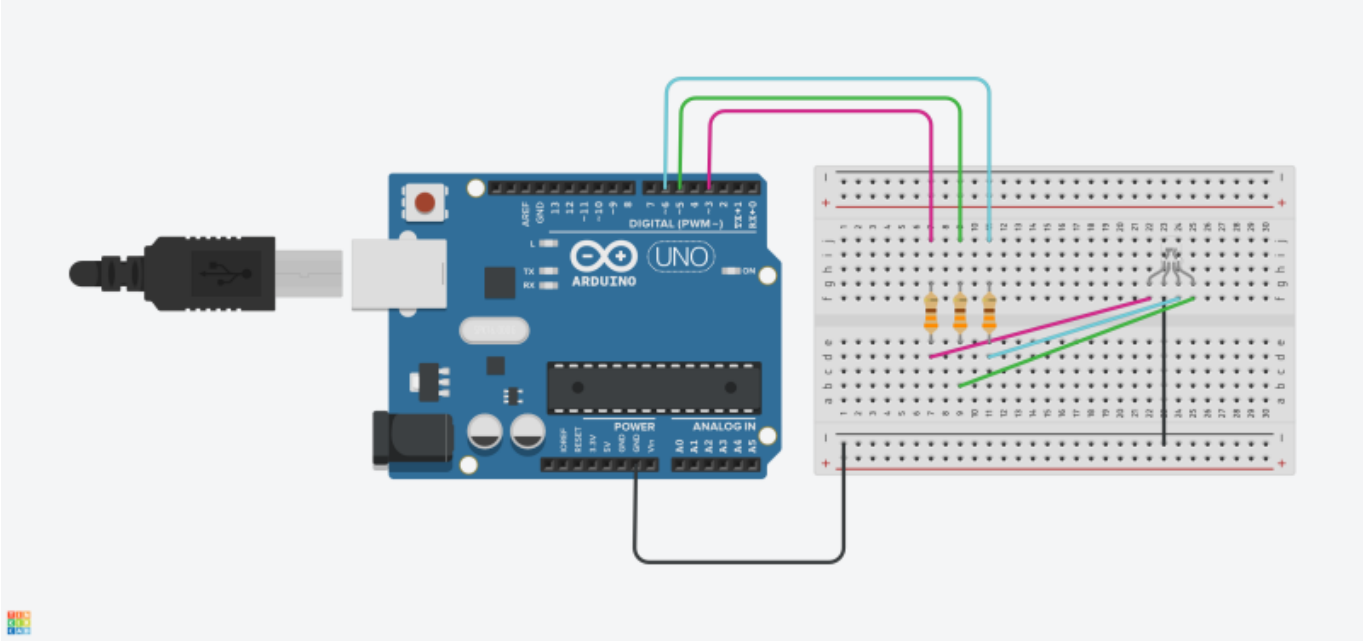
Exercices Arduino avec la maquette

Arduino RVB Tinkercad

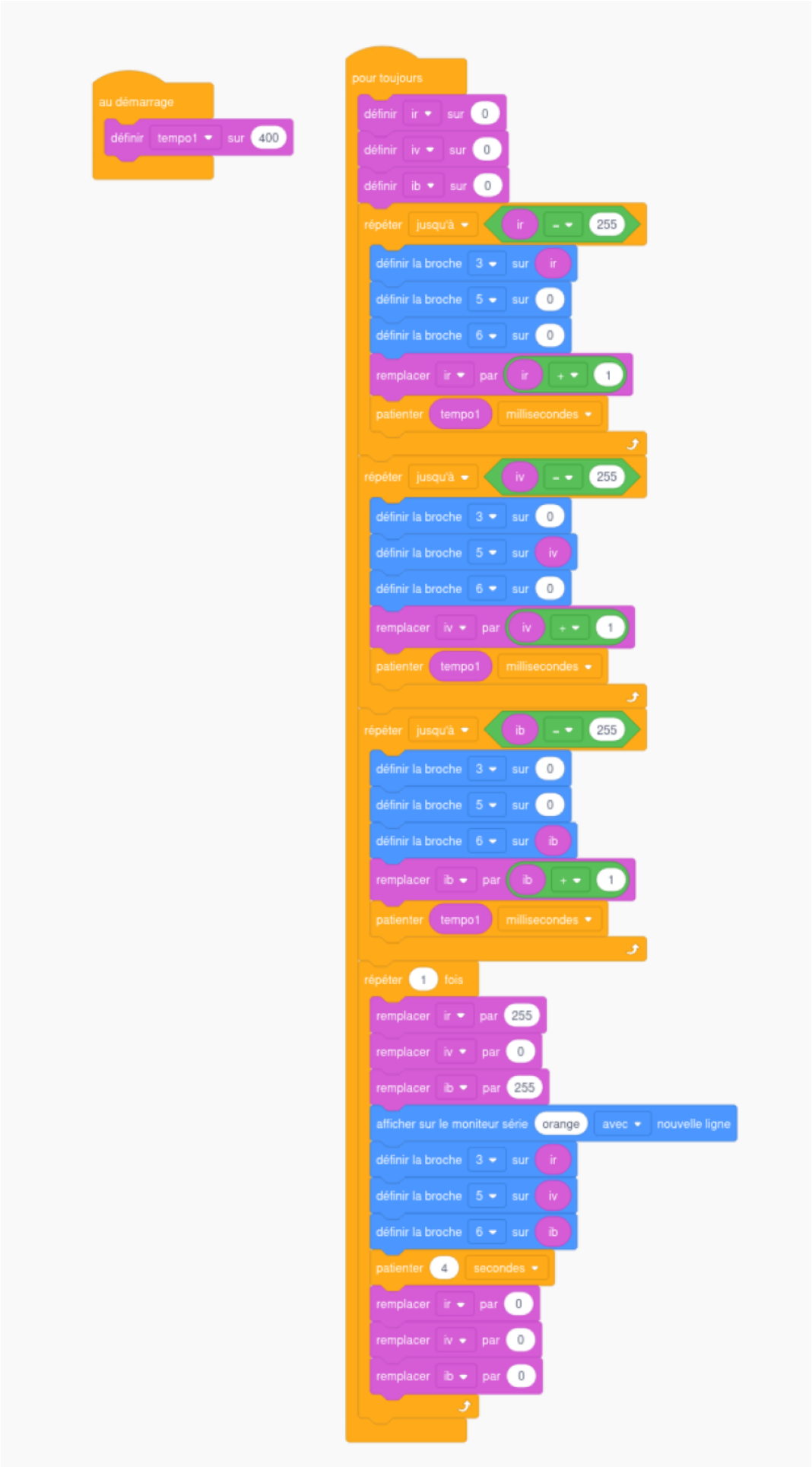
Code couleurs RVB

Plans





Code en mode Block


au démarrage
définir tempo1 sur 400
pour toujours
définir ir sur 0
définir iv sur 0
définir ib sur 0
répéter jusqu'à ir = 255
définir la broche 3 sur ir
définir la broche 5 sur 0
définir la broche 6 sur 0
remplacer ir par ir + 1
patienter tempo1 millisecondes
répéter jusqu'à iv = 255
définir la broche 3 sur 0
définir la broche 5 sur iv
définir la broche 6 sur 0
remplacer iv par iv + 1
patienter tempo1 millisecondes
répéter jusqu'à ib = 255
définir la broche 3 sur 0
définir la broche 5 sur 0
définir la broche 6 sur ib
remplacer ib par ib + 1
patienter tempo1 millisecondes
répéter 1 fois
remplacer ir par 255
remplacer iv par 0
remplacer ib par 255
afficher sur le moniteur série orange avec nouvelle ligne
définir la broche 3 sur ir
définir la broche 5 sur iv
définir la broche 6 sur ib
patienter 4 secondes
remplacer ir par 0
remplacer iv par 0
remplacer ib par 0

Code

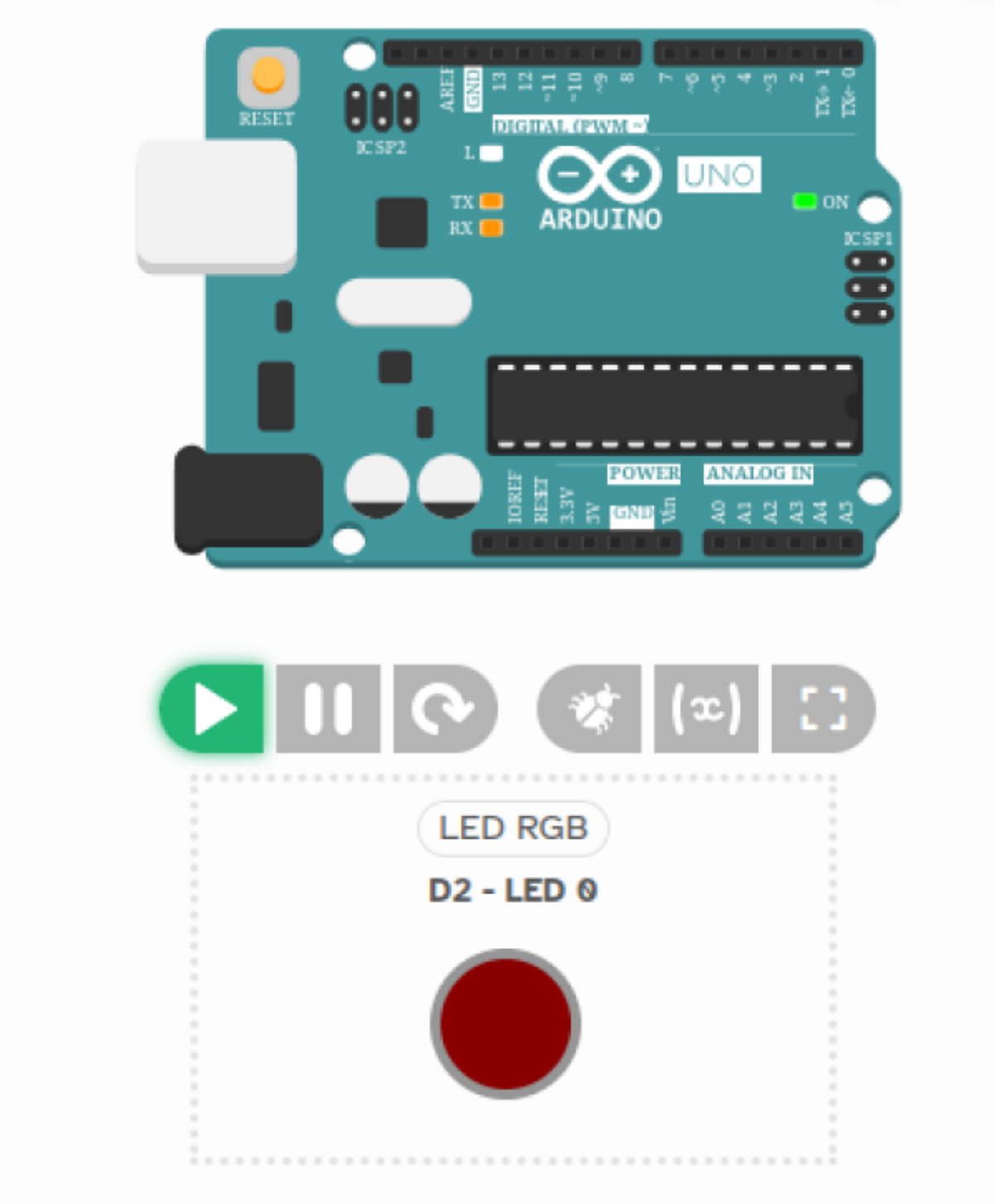
maquette_arduino001.ino

```
// C++ code
// Faire varié les couleurs de la LED RVB
//
int ir = 0;// rouge
int iv = 0;//vert
int ib = 0;//bleu
int tempo1 = 0;
int counter;

void setup()
{
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  Serial.begin(9600);

  tempo1 = 400;
}

void loop()
{
  ir = 0;
  iv = 0;
  ib = 0;
  while (!(ir == 255)) {
    analogWrite(3, ir);
    analogWrite(5, 0);
    analogWrite(6, 0);
    ir += (ir + 1);
    delay(tempo1); // Wait for tempo1 millisecond(s)
  }
  while (!(iv == 255)) {
    analogWrite(3, 0);
    analogWrite(5, iv);
    analogWrite(6, 0);
    iv += (iv + 1);
    delay(tempo1); // Wait for tempo1 millisecond(s)
  }
  while (!(ib == 255)) {
    analogWrite(3, 0);
    analogWrite(5, 0);
    analogWrite(6, ib);
    ib += (ib + 1);
    delay(tempo1); // Wait for tempo1 millisecond(s)
```

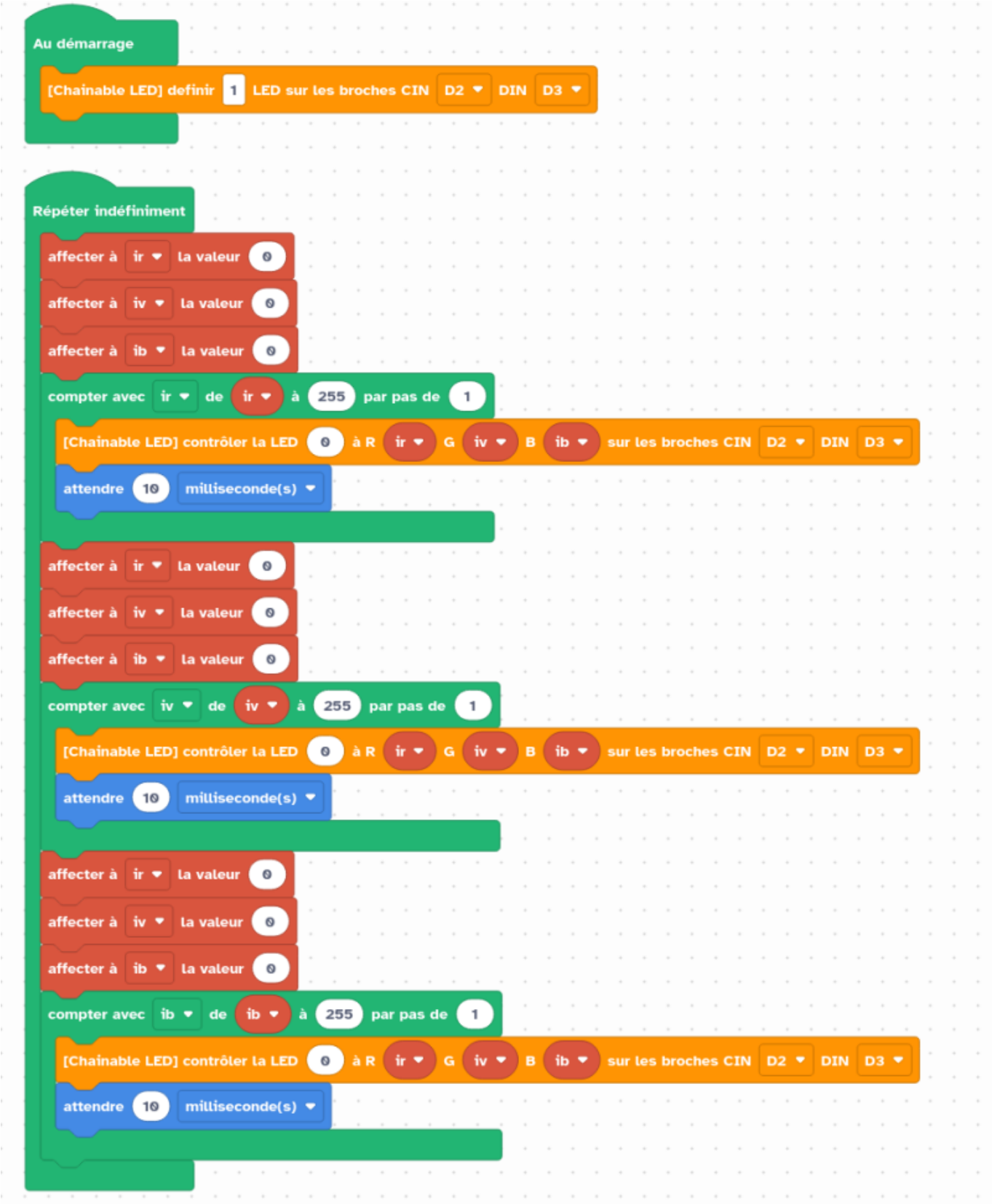
```
  }
  for (counter = 0; counter < 1; ++counter) {
    Serial.println("Rose");
    ir += 255;
    iv += 0;
    ib += 255;
    analogWrite(3, ir);
    analogWrite(5, iv);
    analogWrite(6, ib);
    delay(4000); // Wait for 4000 millisecond(s)
    ir += 0;
    iv += 0;
    ib += 0;
  }
}
```

Arduino RVB-Grove Vittascience

Plan



Code Mode Block



Code C++

arduinoRVB_VTS_001.ino

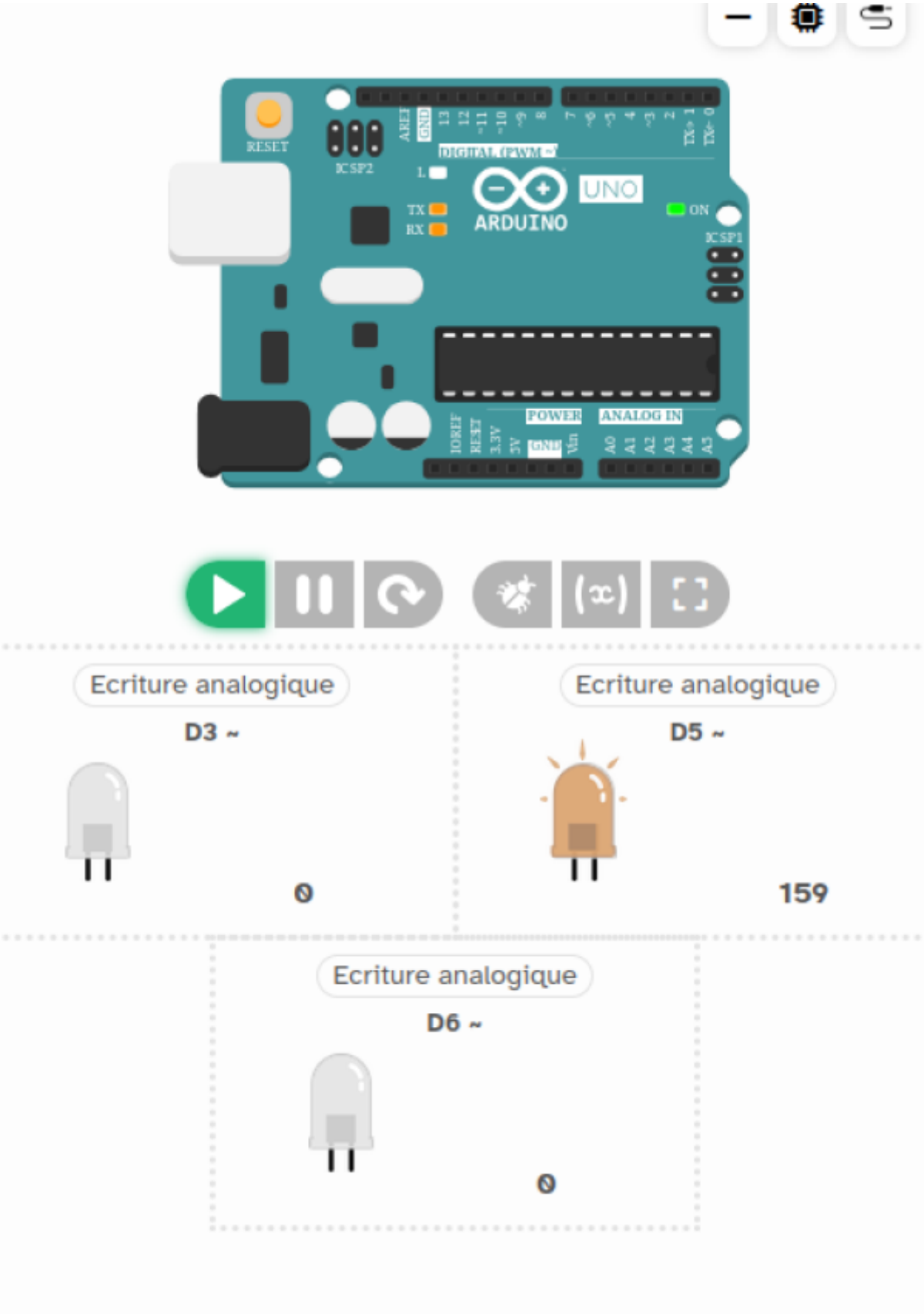
```
#include <ChainableLED.h>

#define PIN_CHAINABLE_LED_DIN_3  3
```

```
#define PIN_CHAINABLE_LED_CIN_2  2
#define CHAINABLE_LED_COUNT_2 1

ChainableLED RGBLed_2(PIN_CHAINABLE_LED_CIN_2, PIN_CHAINABLE_LED_DIN_3,
CHAINABLE_LED_COUNT_2);

int ir;
int iv;
int ib;

void setup() {
  pinMode(PIN_CHAINABLE_LED_DIN_3, OUTPUT);
  pinMode(PIN_CHAINABLE_LED_CIN_2, OUTPUT);
  RGBLed_2.init();
}

void loop() {
  ir = 0;
  iv = 0;
  ib = 0;
  int ir_inc = 1;
  if (ir > 255) {
    ir_inc = -ir_inc;
  }
  for (ir = ir;
       ir_inc >= 0 ? ir <= 255 : ir >= 255;
       ir += ir_inc) {
    RGBLed_2.setColorRGB(0, ir, iv, ib);
    delay(10);
  }
  ir = 0;
  iv = 0;
  ib = 0;
  int iv_inc = 1;
  if (iv > 255) {
    iv_inc = -iv_inc;
  }
  for (iv = iv;
       iv_inc >= 0 ? iv <= 255 : iv >= 255;
       iv += iv_inc) {
    RGBLed_2.setColorRGB(0, ir, iv, ib);
    delay(10);
  }
  ir = 0;
  iv = 0;
  ib = 0;
  int ib_inc = 1;
  if (ib > 255) {
    ib_inc = -ib_inc;
```

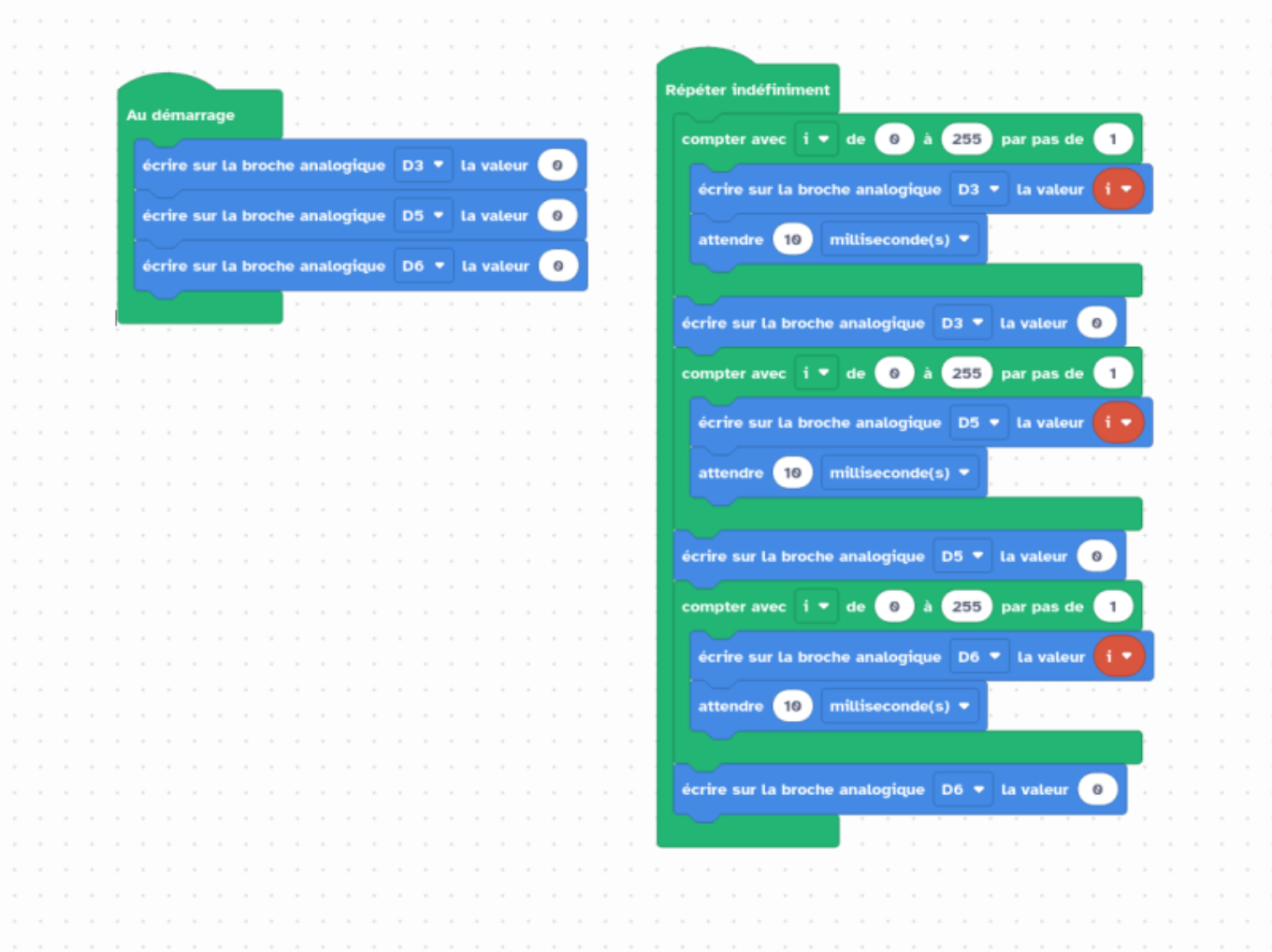
```
  }
  for (ib = ib;
       ib_inc >= 0 ? ib <= 255 : ib >= 255;
       ib += ib_inc) {
    RGBLed_2.setColorRGB(0, ir, iv, ib);
    delay(10);
  }
}
```

Arduino RVB Cathode commune

Plan


RESET
ICSP2
DIGITAL (PWM ~)
UNO
ARDUINO
TX
RX
ON
ICSP1
POWER
ANALOG IN
Ecriture analogique
D3 ~
0
Ecriture analogique
D5 ~
159
Ecriture analogique
D6 ~
0

Code block



Code C++

arduinoRVBCatC.ino

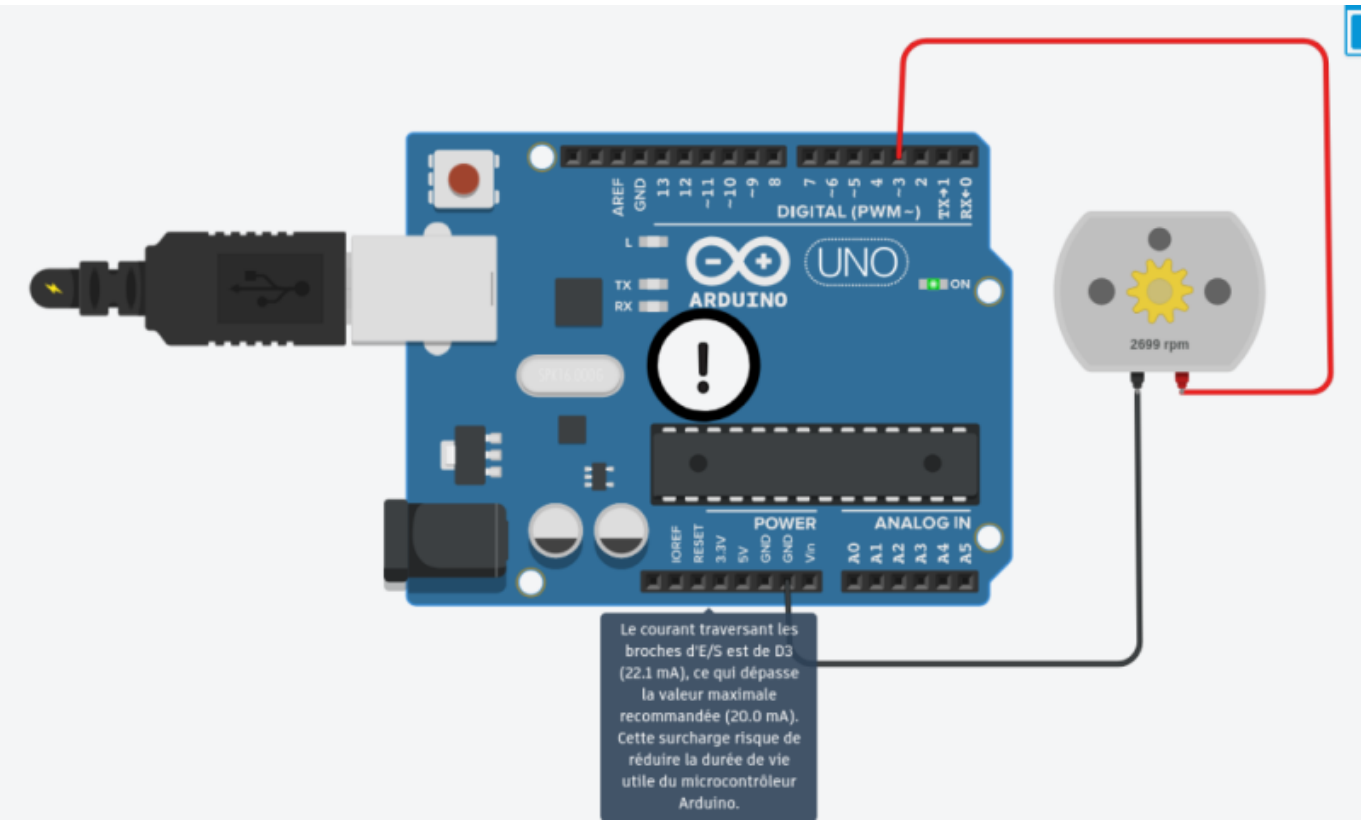
```
int i;

void setup() {
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  analogWrite(3, 0);
  analogWrite(5, 0);
  analogWrite(6, 0);
}

void loop() {
  for (i = 0; i <= 255; i++) {
    analogWrite(3, i);
    delay(10);
  }
  analogWrite(3, 0);
  for (i = 0; i <= 255; i++) {
    analogWrite(5, i);
```

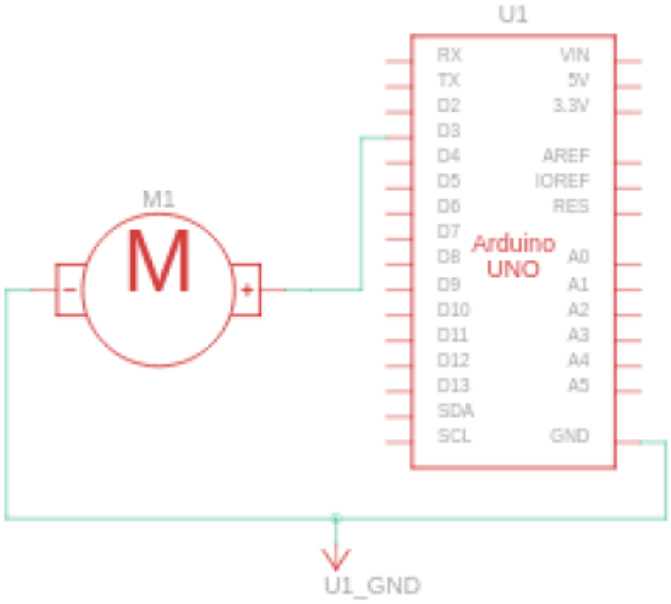
```
    delay(10);
  }
  analogWrite(5, 0);
  for (i = 0; i <= 255; i++) {
    analogWrite(6, i);
    delay(10);
  }
  analogWrite(6, 0);
}
```

Arduino Moteur CC 1 sens

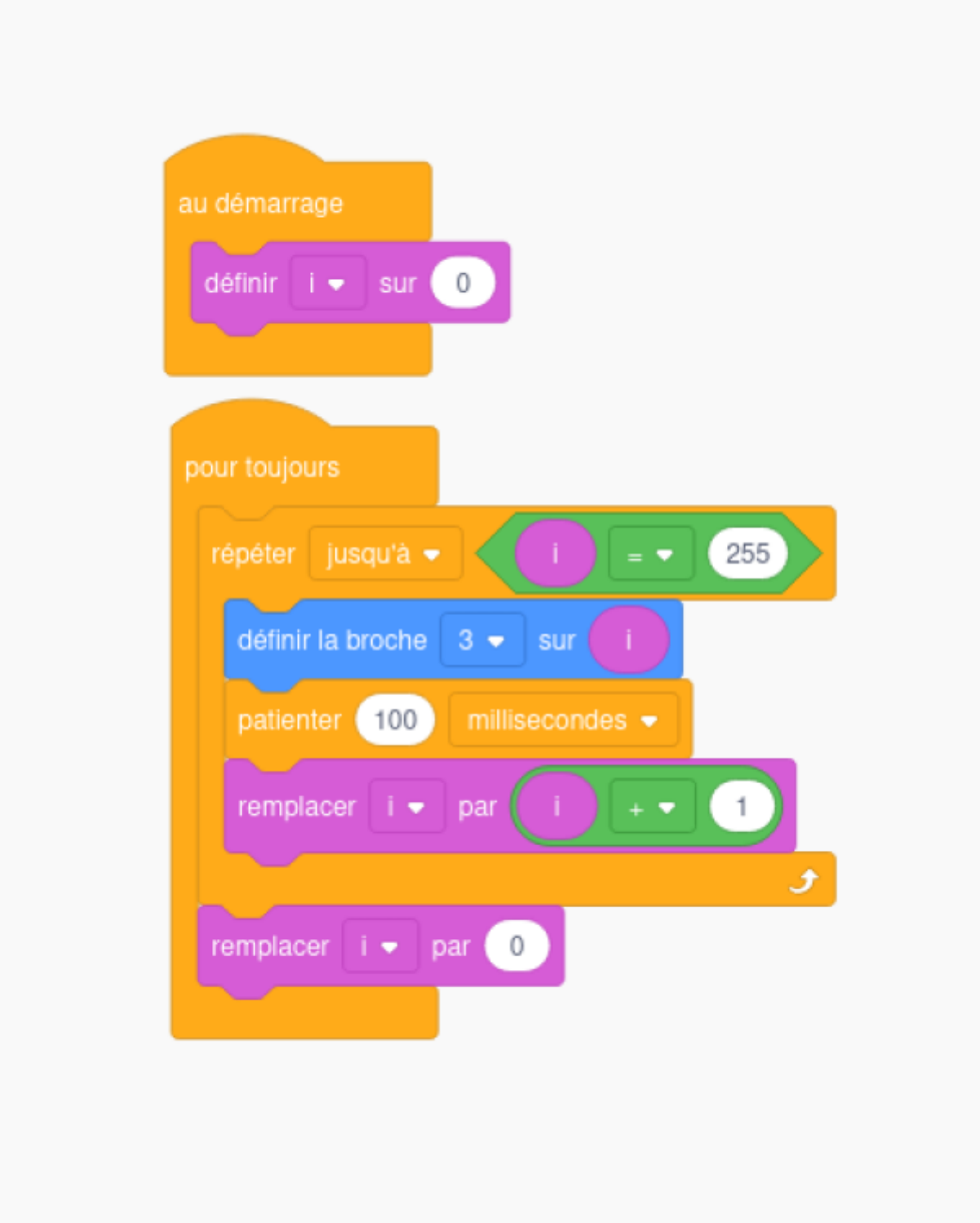
Plans

Moteur en direct sur sortie Arduino




U1
RX
TX
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
SDA
SCL
VIN
5V
3.3V
AREF
IOREF
RES
A0
A1
A2
A3
A4
A5
GND
Arduino
UNO
M1
M
U1_GND

Code Block



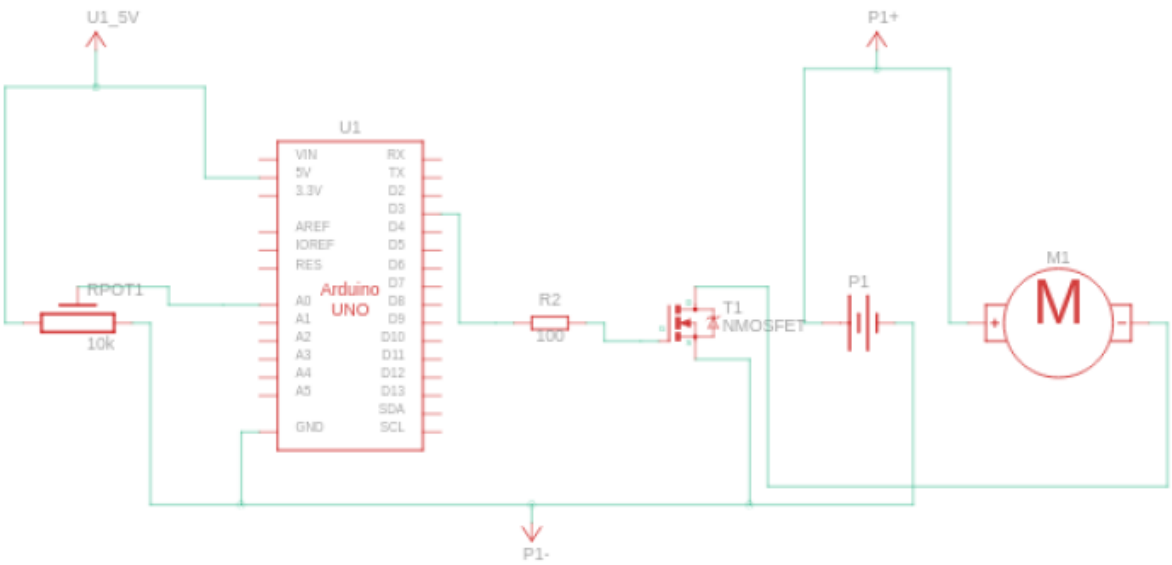
code C++

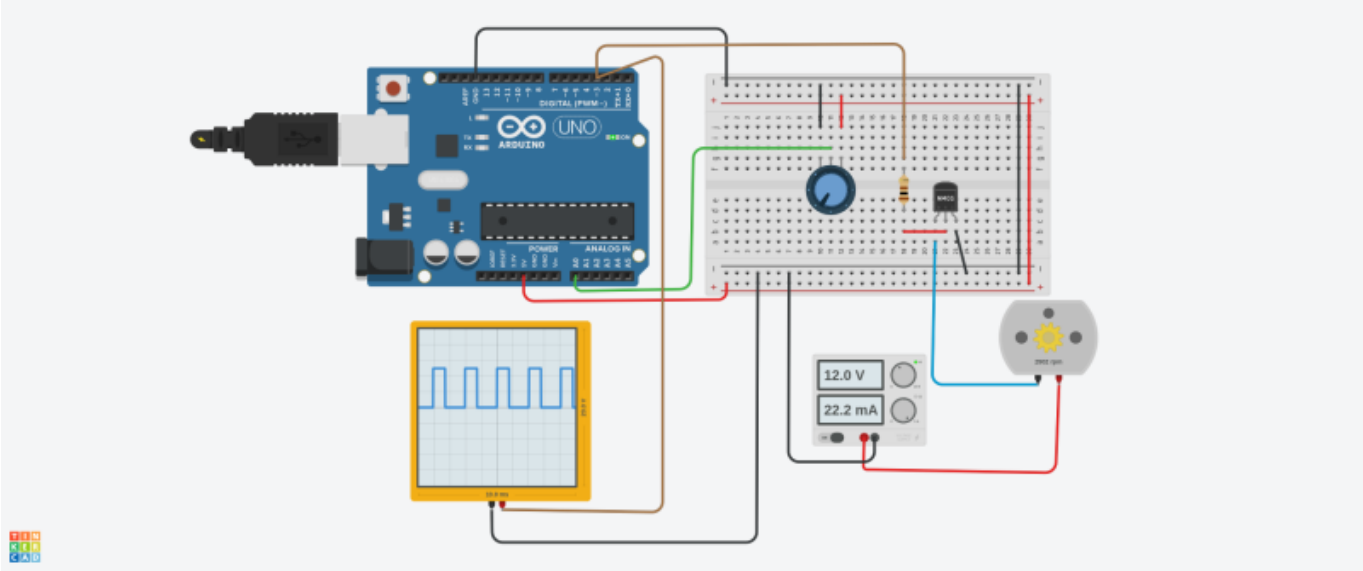
Arduinomoteurdirect.ino

```
// C++ code
//
int i = 0;

void setup()
```

```
{
  pinMode(3, OUTPUT);

  i = 0;
}

void loop()
{
  while (!(i == 255)) {
    analogWrite(3, i);
    delay(100); // Wait for 100 millisecond(s)
    i += (i + 1);
  }
  i += 0;
}
```

Avec commande moteur via Mosfet





Code C++

ArduinoMoteurViaMosfet.ino

```
/*
Faire varier la vitesse du moteur en fonction de la position et donc de la valeur  du potentiometre
*/

  int motorPin = 3;     // Controle de vitesse PWM vers la gate du Mofset
  int potarPin=0;       // on branche un potentiomètre sur la PIn A0
  int sensorValue=0;    // la valeur entre 0 et 1023 déduite de la position du potentiomètre
  int outputValue=0;    // la valeur entre 0 et 255 que l'on va utiliser pour régler la vitesse du moteur

  // Executee une seule fois

  void setup() {
     // initialiser la communication serie
     Serial.begin(19200);
     pinMode(motorPin, OUTPUT); // motorPin est utilisee en sortie
  }

  // executee en boucle
  void loop() {
     // lire la valeur en entrée de la Pin A0
     sensorValue = analogRead(potarPin);
     // afficher sur le moniteur série la valeur lue, pas obligatoire mais toujours intéressant
     Serial.println(sensorValue);
     // conversion de la valeur sensorvalue (qui va de 0 à 1023) en une
```

```
valeur entre 0 et 255, outputvalue
     outputValue = map(sensorValue, 0, 1023, 0, 255);
     // outputvalue permet de controler la vitesse du moteur
     analogWrite(motorPin,outputValue );  //  valeur de 0 à 255
     delay(100);
  }
```

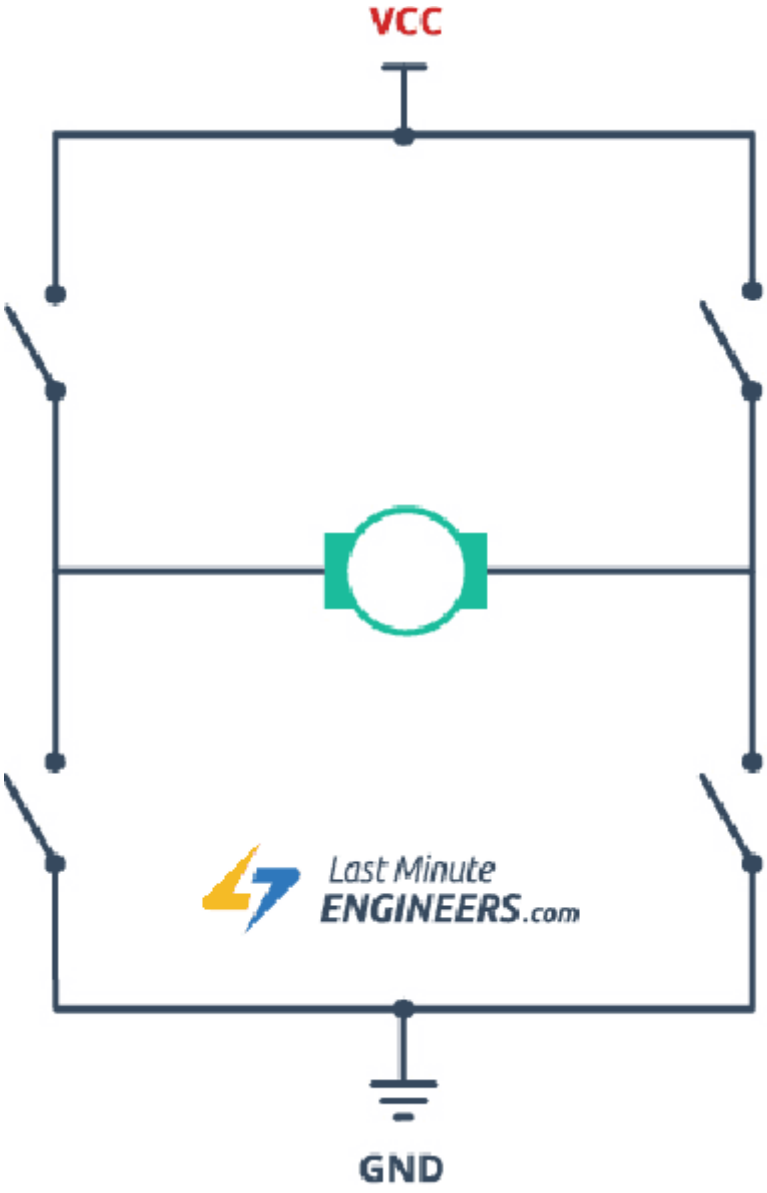
Arduino moteur 2 sens Via L293D

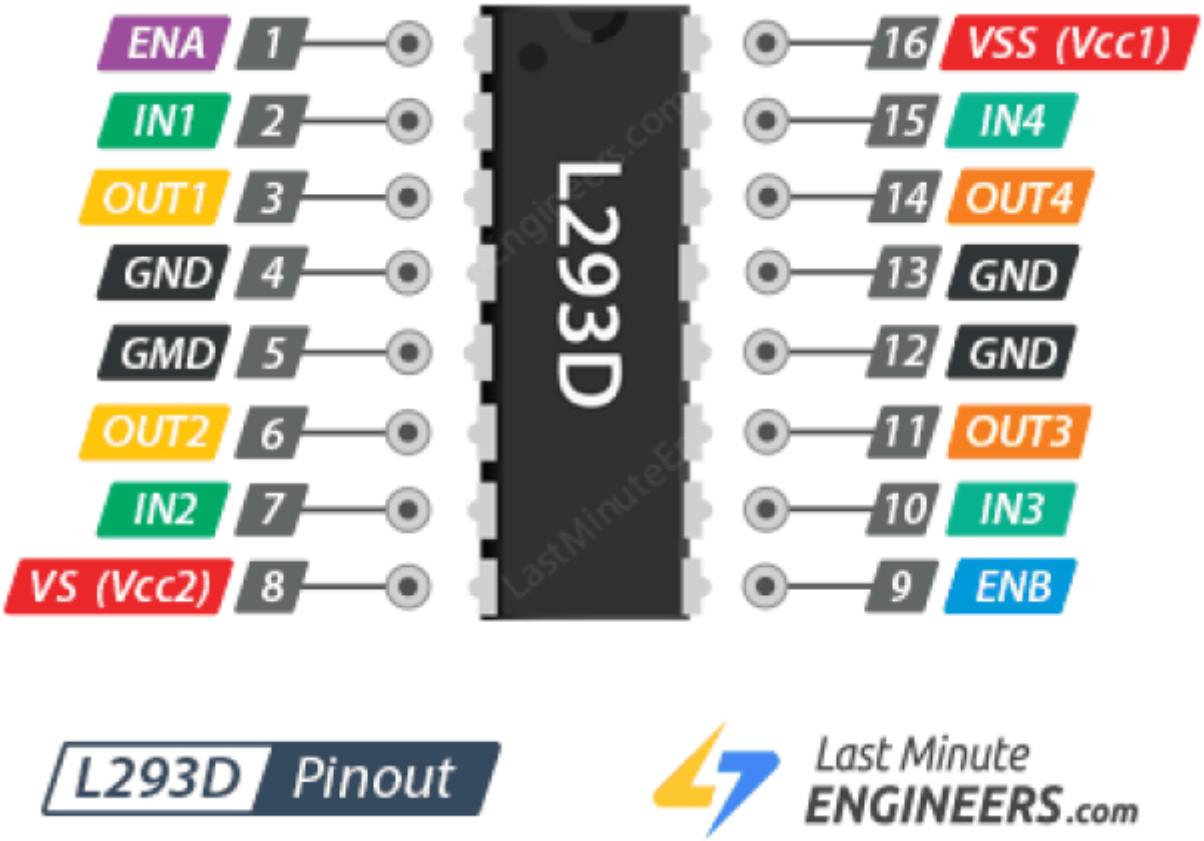
Doc L293D

Notice technique L293D EN

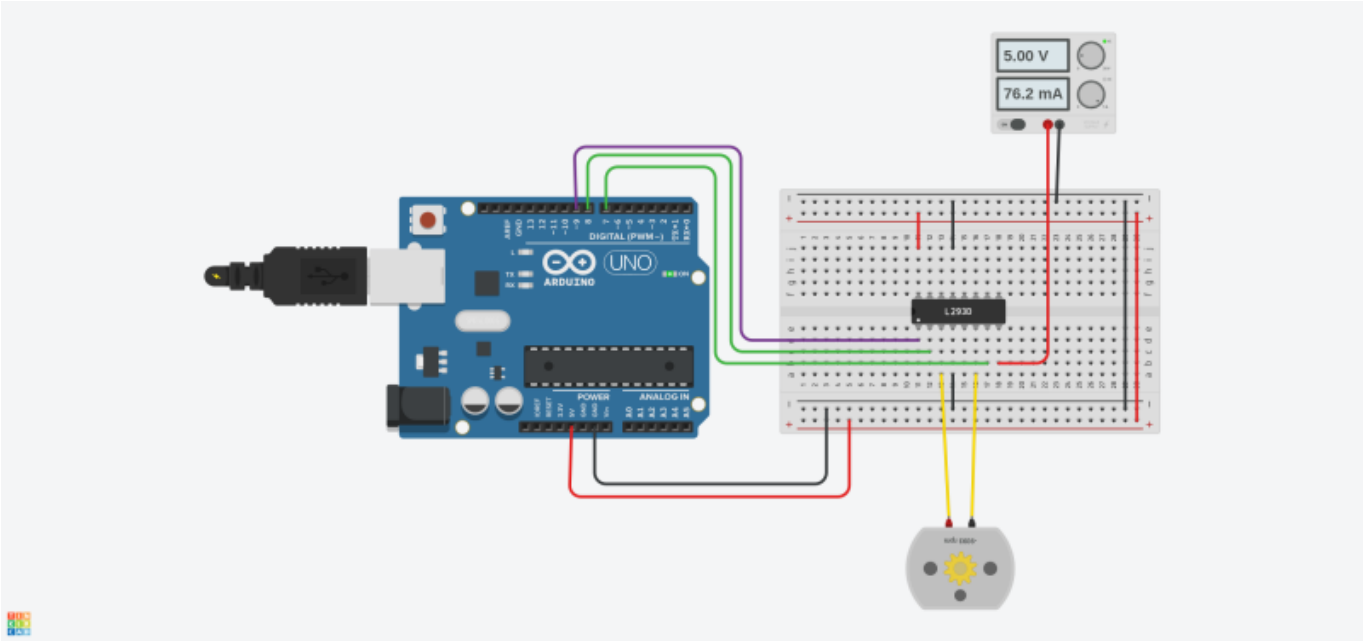
Tuto l293D et Arduino

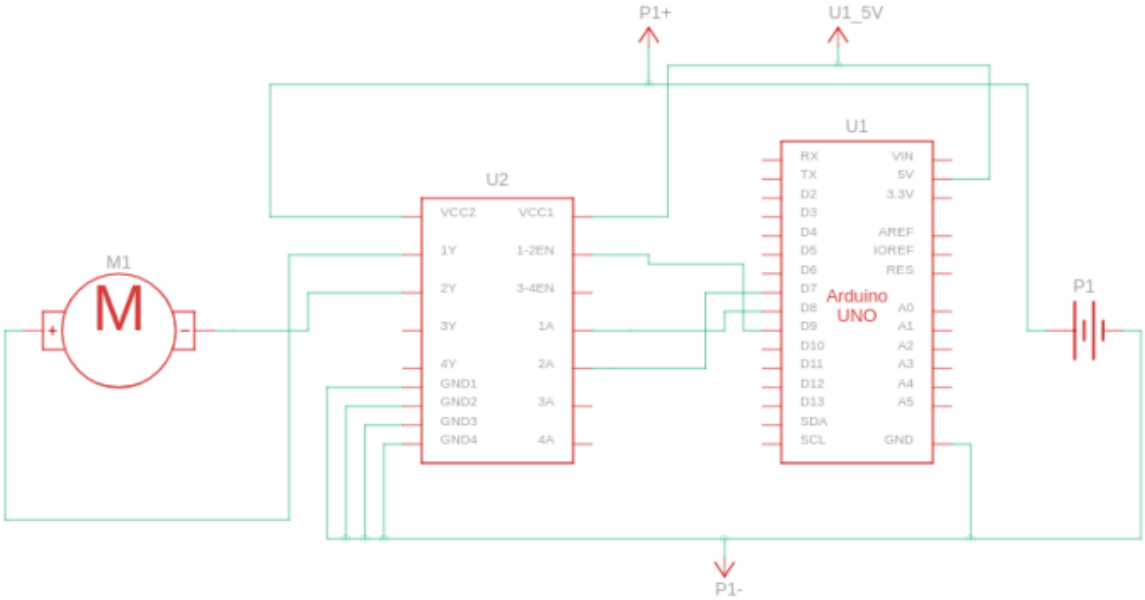
l293d


VCC
Last Minute
ENGINEERS.com
GND

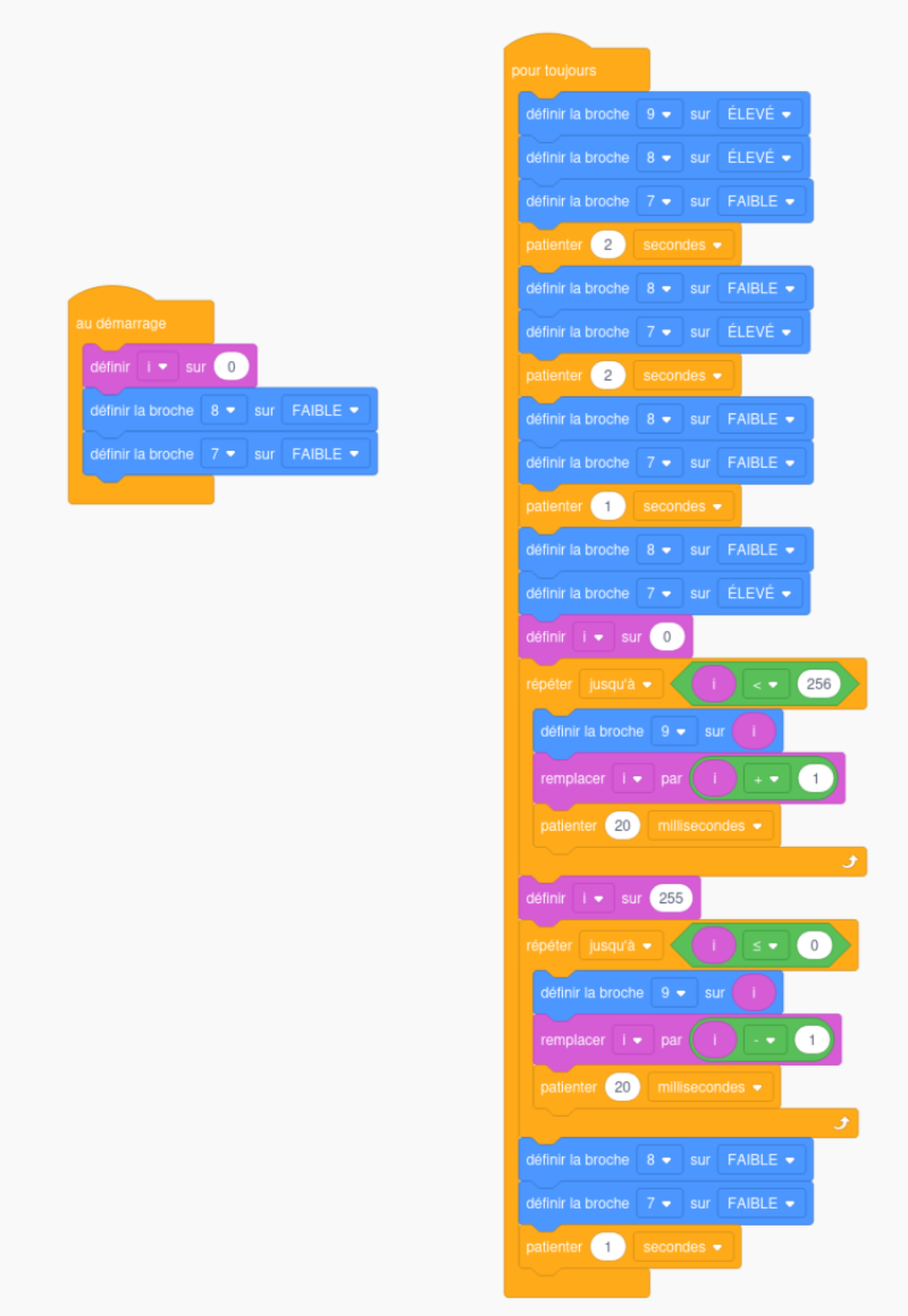

ENA 1
IN1 2
OUT1 3
GND 4
GMD 5
OUT2 6
IN2 7
VS (Vcc2) 8
L293D
16 VSS (Vcc1)
15 IN4
14 OUT4
13 GND
12 GND
11 OUT3
10 IN3
9 ENB
L293D Pinout
Last Minute
ENGINEERS.com

Plans


5.00 V
76.2 mA
UNO
ARDUINO
L293D


P1+
U1_5V
U1
U2
M1
M
+
-
VCC2
VCC1
1Y
1-2EN
2Y
3-4EN
3Y
1A
4Y
2A
GND1
GND2
3A
GND3
GND4
4A
RX
TX
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
SDA
SCL
VIN
5V
3.3V
AREF
IOREF
RES
Arduino
UNO
A0
A1
A2
A3
A4
A5
GND
P1
P1-

Code Block


au démarrage
définir i sur 0
définir la broche 8 sur FAIBLE
définir la broche 7 sur FAIBLE
pour toujours
définir la broche 9 sur ÉLEVÉ
définir la broche 8 sur ÉLEVÉ
définir la broche 7 sur FAIBLE
patienter 2 secondes
définir la broche 8 sur FAIBLE
définir la broche 7 sur ÉLEVÉ
patienter 2 secondes
définir la broche 8 sur FAIBLE
définir la broche 7 sur FAIBLE
patienter 1 secondes
définir la broche 8 sur FAIBLE
définir la broche 7 sur ÉLEVÉ
définir i sur 0
répéter jusqu'à i < 256
définir la broche 9 sur i
remplacer i par i + 1
patienter 20 millisecondes
définir i sur 255
répéter jusqu'à i ≤ 0
définir la broche 9 sur i
remplacer i par i - 1
patienter 20 millisecondes
définir la broche 8 sur FAIBLE
définir la broche 7 sur FAIBLE
patienter 1 secondes

Code C++

arduinomoteurL293D.ino

```
// Motor A connections
int enA = 9;
int in1 = 8;
int in2 = 7;

void setup() {
  // Set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);


  // Turn off motors - Initial state
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);

}

void loop() {
  directionControl();
  delay(1000);
  speedControl();
  delay(1000);
}

// This function lets you control spinning direction of motors
void directionControl() {
  // Set motors to maximum speed
  digitalWrite(enA, HIGH);


  // Turn on motor A
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  delay(2000);

  // Now change motor directions
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  delay(2000);

  // Turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
```

```
 }

// This function lets you control speed of the motors
void speedControl() {
  // Turn on motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);


  // Accelerate from zero to maximum speed
  for (int i = 0; i < 256; i++) {
    analogWrite(enA, i);
    delay(20);
  }

  // Decelerate from maximum speed to zero
  for (int i = 255; i >= 0; --i) {
    analogWrite(enA, i);
    delay(20);
  }

  // Now turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
}
```

From:
https://www.magenealogie.chanterie37.fr/www/fablab37110/ - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:
https://www.magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=exercices:arduino&rev=1791461632

Last update: **2026/10/08 14:13**