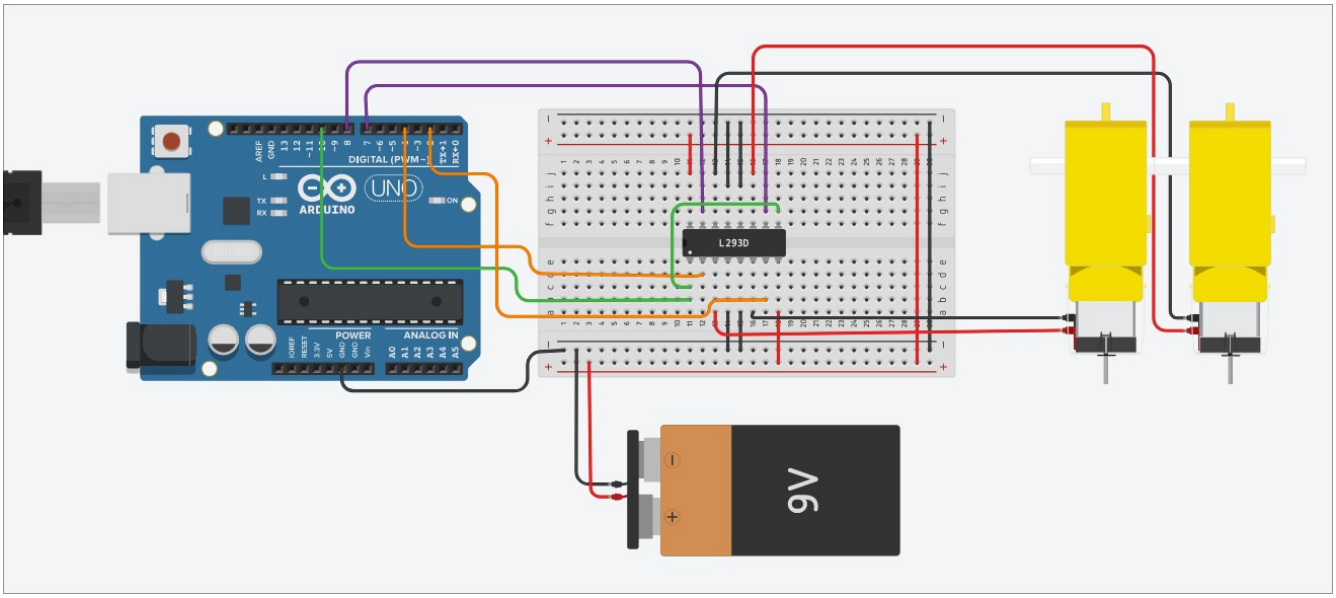
**Código**

```
const byte MOTOR1_PINS[2] = {4, 2};
const byte MOTOR2_PINS[2] = {7, 8};
const byte ENABLE_PIN = 10;
/* Con esta constante se controla la velocidad del motor (0%-100%) */
const float speed = 0.11*255;

void setup(){
  pinMode(MOTOR1_PINS[0], OUTPUT);
  pinMode(MOTOR1_PINS[1], OUTPUT);
  pinMode(MOTOR2_PINS[0], OUTPUT);
  pinMode(MOTOR2_PINS[1], OUTPUT);

  turnOffPins();
  analogWrite(ENABLE_PIN, speed);
  Serial.begin(9600);
}

void loop(){
  goForward();
  delay(1500);
  goBackward();
  delay(1500);
  stop();
  delay(850);
  turnRight();
  delay(1500);
  turnLeft();
  delay(1000);
  turnRight180();
  delay(1000);
  turnLeft180();
  delay(1000);
}

void stop() {
  digitalWrite(MOTOR1_PINS[0], LOW);
  digitalWrite(MOTOR1_PINS[1], LOW);
  digitalWrite(MOTOR2_PINS[0], LOW);
  digitalWrite(MOTOR2_PINS[1], LOW);
}

void turnOffPins() {
  stop();
  analogWrite(ENABLE_PIN, 0);
}

void turnLeft() {
```

```
  stop();
  digitalWrite(MOTOR2_PINS[0], HIGH);
}

void turnRight() {
  stop();
  digitalWrite(MOTOR1_PINS[0], HIGH);
}

void goForward() {
  stop();
  digitalWrite(MOTOR1_PINS[0], HIGH);
  digitalWrite(MOTOR2_PINS[0], HIGH);
}

void goBackward() {
  stop();
  digitalWrite(MOTOR1_PINS[1], HIGH);
  digitalWrite(MOTOR2_PINS[1], HIGH);
}

/* La lógica detrás de estos dos métodos es convertir las RPM en velocidad angular
    para después despejar el tiempo necesario para girar 180° (pi radianes) */
void turnRight180() {
  int rpm = map(speed, 0, 255, 0, 231);
  turnRight();
  delay(int(30.0/rpm*1000));
}

void turnLeft180() {
  int rpm = map(speed, 0, 255, 0, 231);
  turnLeft();
  delay(int(30.0/rpm*1000));
}
```

## Circuito




## Enlaces

https://www.tinkercad.com/things/2LS4DuymjHn-tarea-1-iot