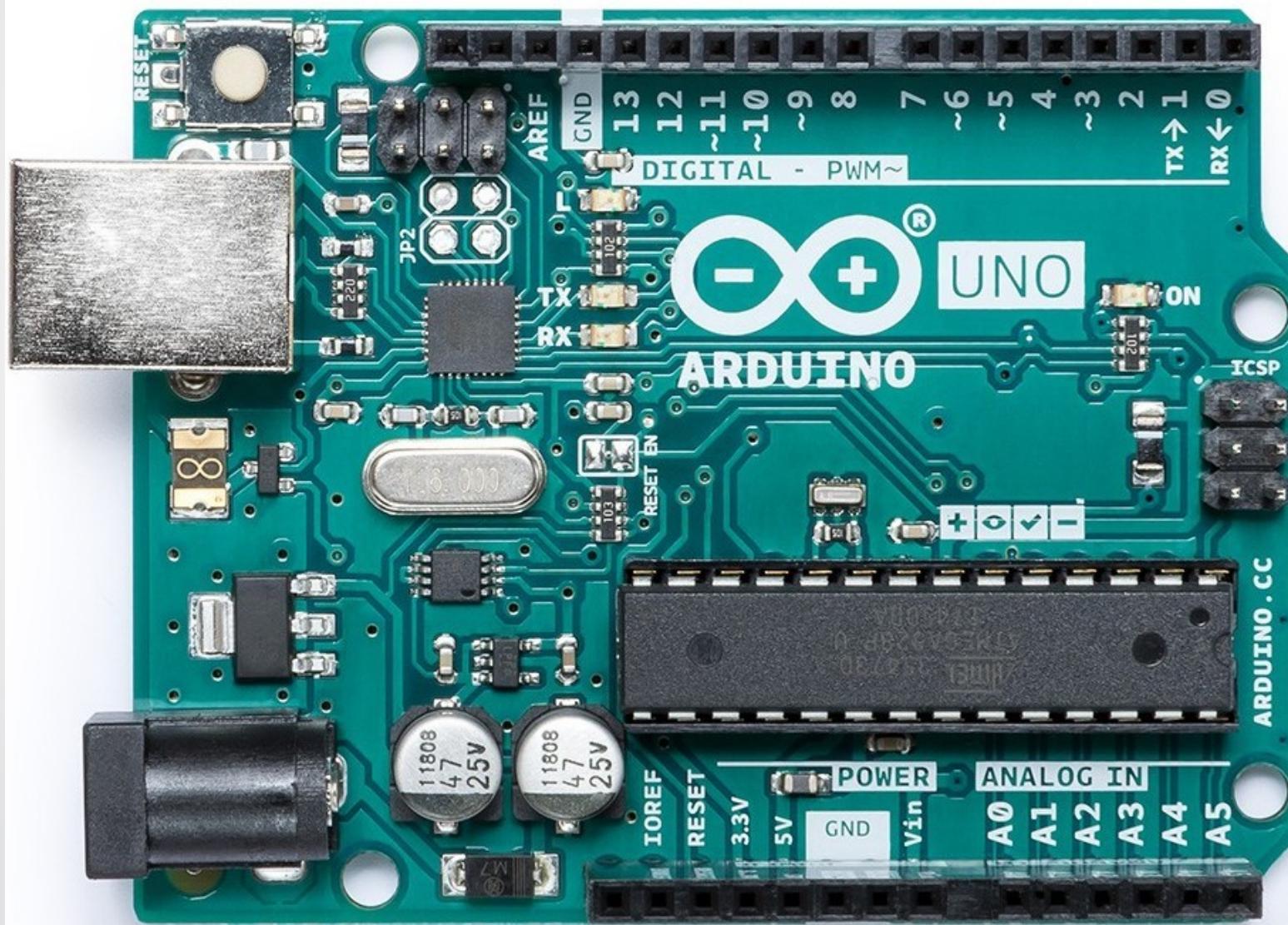
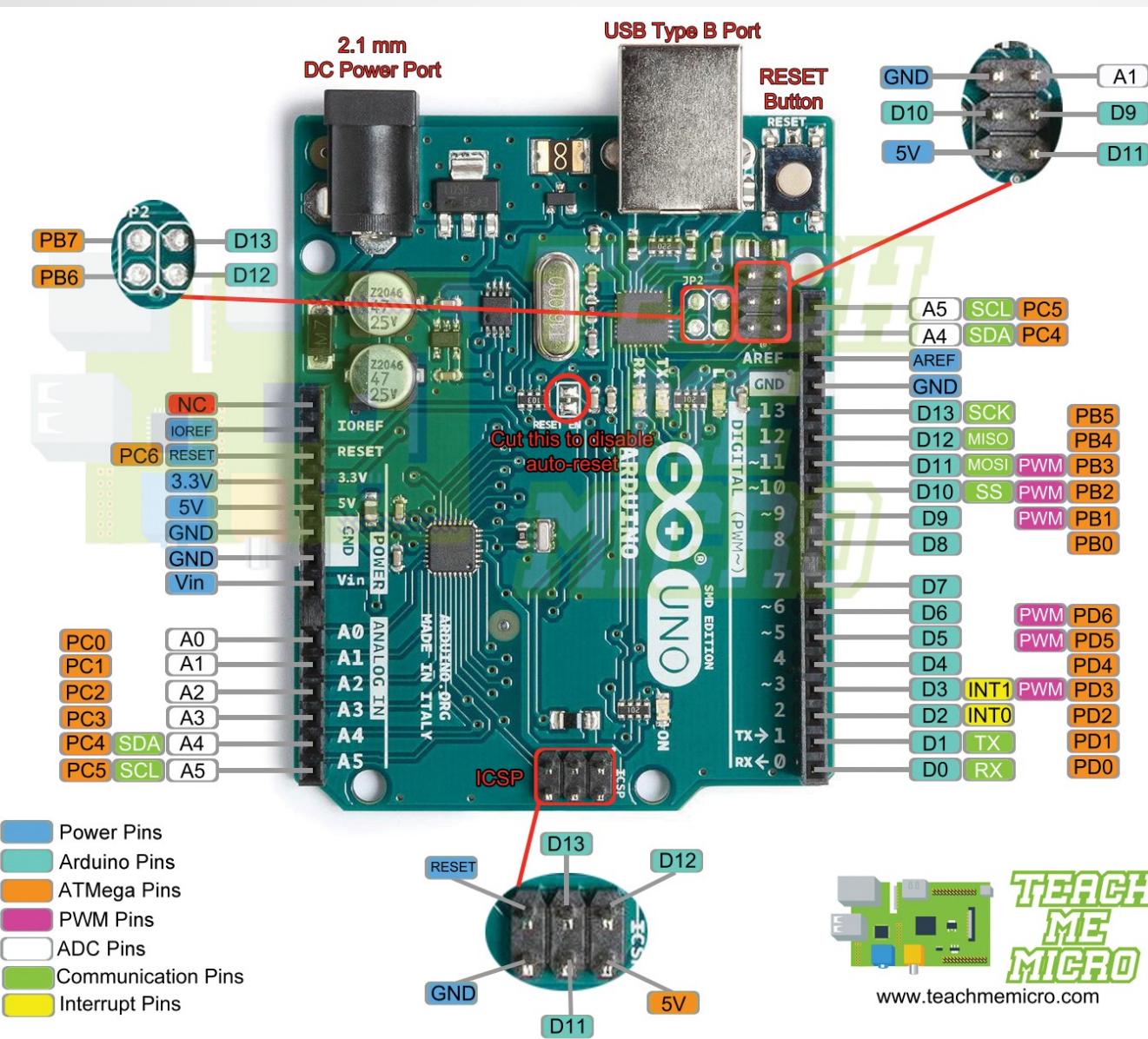


Motores DC



Motores DC – pinout Arduino



E/S

Dixitais:

- Pins 0 a 13
0 a 5V, 20 mA
Low: 0 a 2V
High: 3 a 5V

Entradas Analóxicas

- Pins A0 – A5
0 a 5V, 20 mA prec 1024
0 a 3V, 50 mA prec 1024

Sáidas PWM

- ~ Pins 3, 5, 6, 9, 10, 11
0 a 5 V, 20 mA prec 256

Comunic. Serie TX/RX

- Pins 0 e 1

ICSP

- 6 pins para comunicarse directamente co proc. Atmega328

- 6 pins para programar o USB

TEACH
ME
MICRO

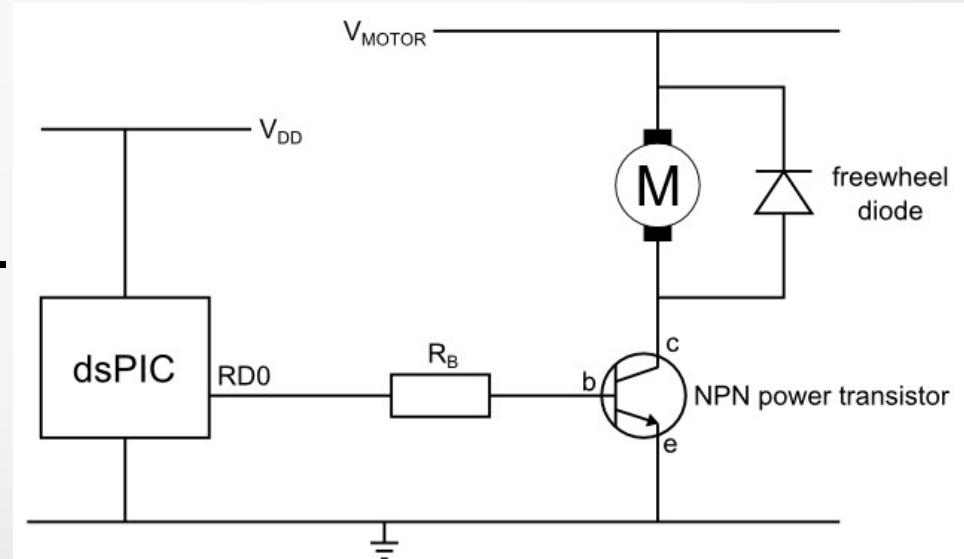
www.teachmemicro.com

Motores DC

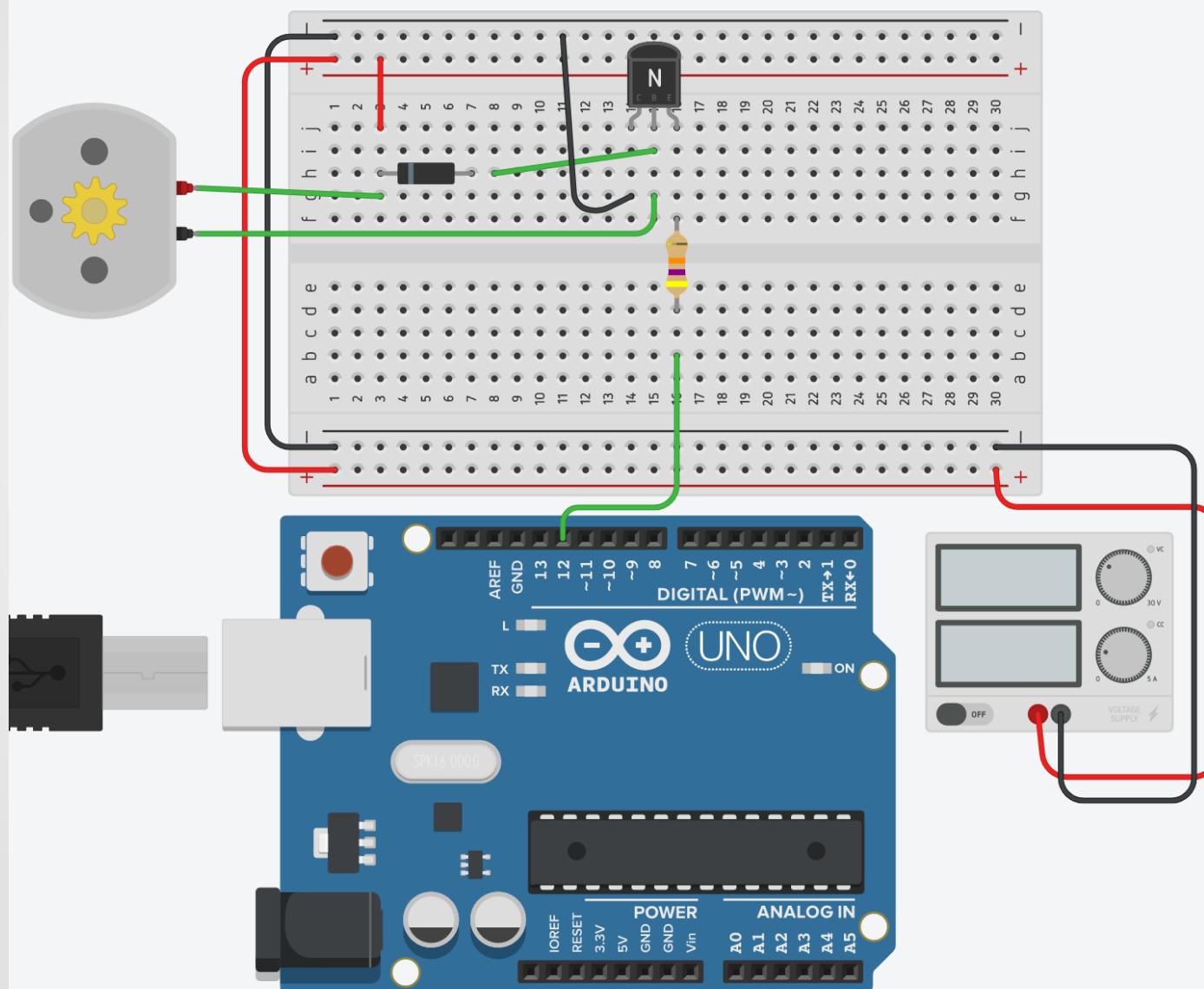
- Un motor de corrente contínua (CC en galego ou DC en inglés) é un actuador que consome unha intensidade relativamente elevada para a placa Arduino.
- A intensidade que pode suministrar un Arduino por un pin de saída está entre os 20 e os 40 mA. Polo tanto debemos buscar algunha outra alternativa para controlar un motor CC con Arduino.
- Por outra parte, os motores introducen interferencias no circuíto, de maneira que outros elementos conectados poden non funcionar correctamente. Por iso é importante telos illados do circuíto principal e con alimentación separada.
- Como controladores dun motor DC podemos empregar:
 - transistores,
 - relés,
 - ou pontes H de diodos, en particular o integrado L293D

Motores DC - Transistor

- Como exemplo de proba, imos controlar un motor DC empregando un transistor BC547, de maneira que podamos arrancalo.
- Imos realizar a montaxe da figura, empregando un transistor NPN BC547, unha resistencia de $47\text{ k}\Omega$ e unha batería externa de 9 V.
- O diodo está para asegurar que no arranque e paro o motor non introduza correntes debido ás bobinas.
- É importante GND sexa común.

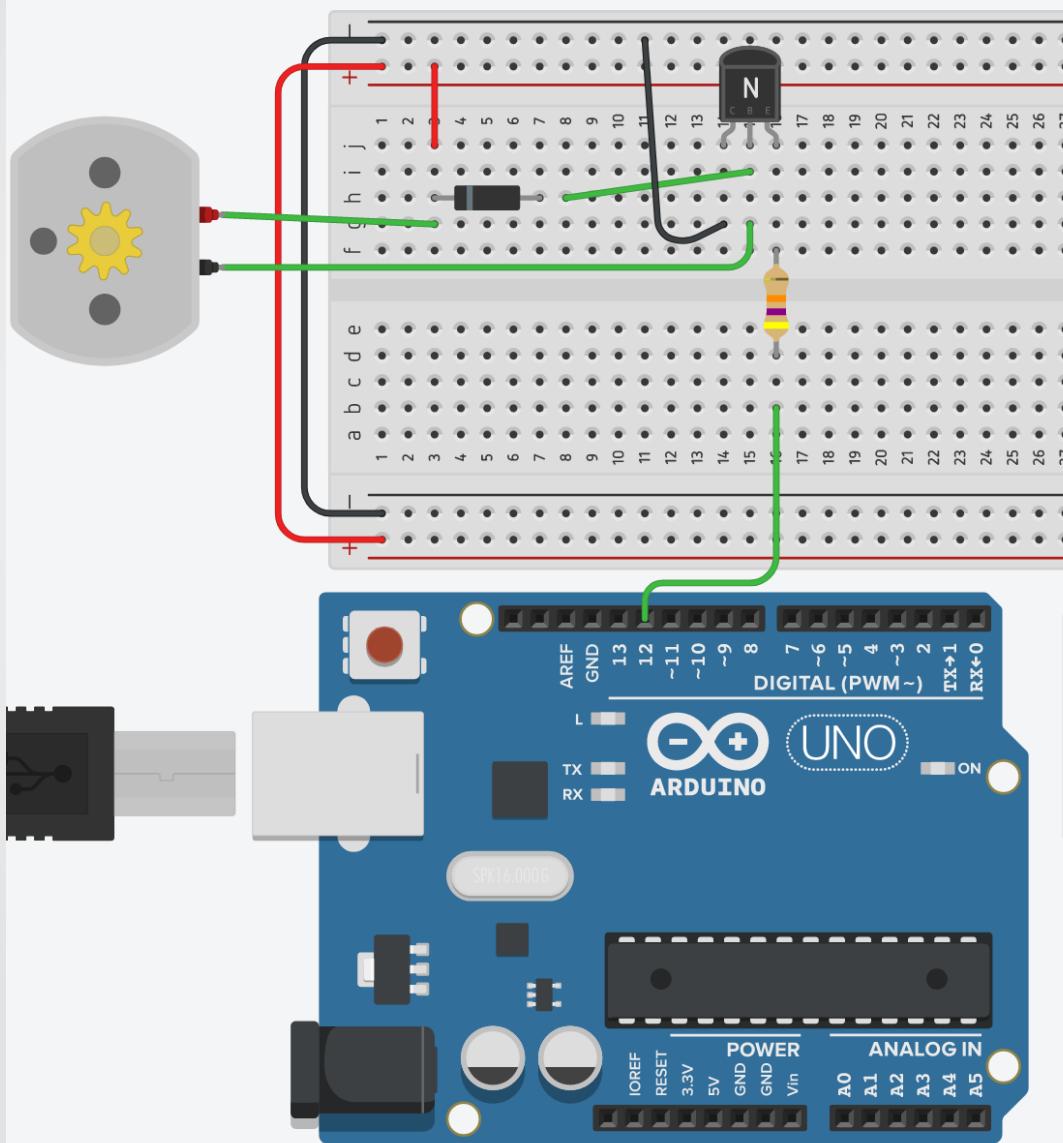


Motores DC - Transistor



Datasheet BC547

Motores DC - Transistor



```
/*
 * Control dun motor de CC
 * usando un transistor NPN BC547
 */

//Imos empregar o pin 12 tanto para
//saída dixital como para PWM
#define CONTROL 12

int espera = 1000;

void setup() {
  pinMode(CONTROL, OUTPUT);
  Serial.begin(9600);
}

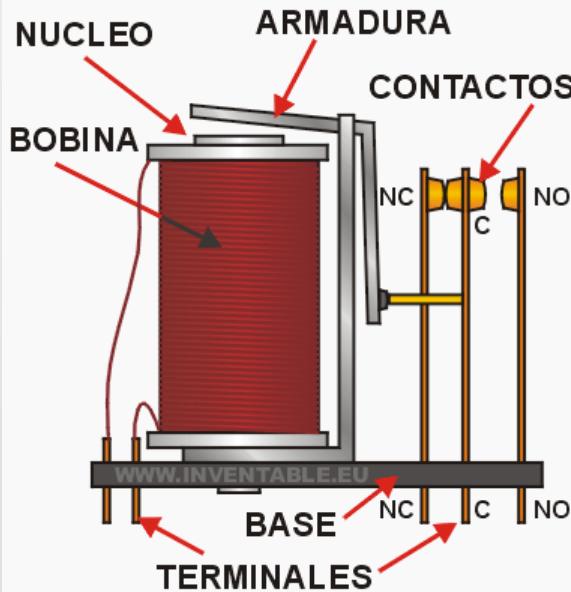
void loop() {
  //Marcha/parao do motor cao voltaxe HIGH
  digitalWrite(CONTROL, HIGH);
  delay(espera);
  digitalWrite(CONTROL, LOW);
  delay(espera);

  //Reducción de velocidade do motor dende
  //max (255) a min (0) en incrementos de 5
  for(int veloc = 255; veloc >= 0; veloc -= 5) {
    analogWrite(CONTROL, veloc);
    Serial.println("Veloc. en pasos PWM: " + String(veloc));
    delay(espera/2);
  }

  //Incremento de velocidade do motor dende
  //min (0) a max (255) en incrementos de 5
  for(int veloc = 0; veloc <= 255; veloc += 5) {
    analogWrite(CONTROL, veloc);
    Serial.println("Veloc. en pasos PWM: " + String(veloc));
    delay(espera/2);
  }
}
```

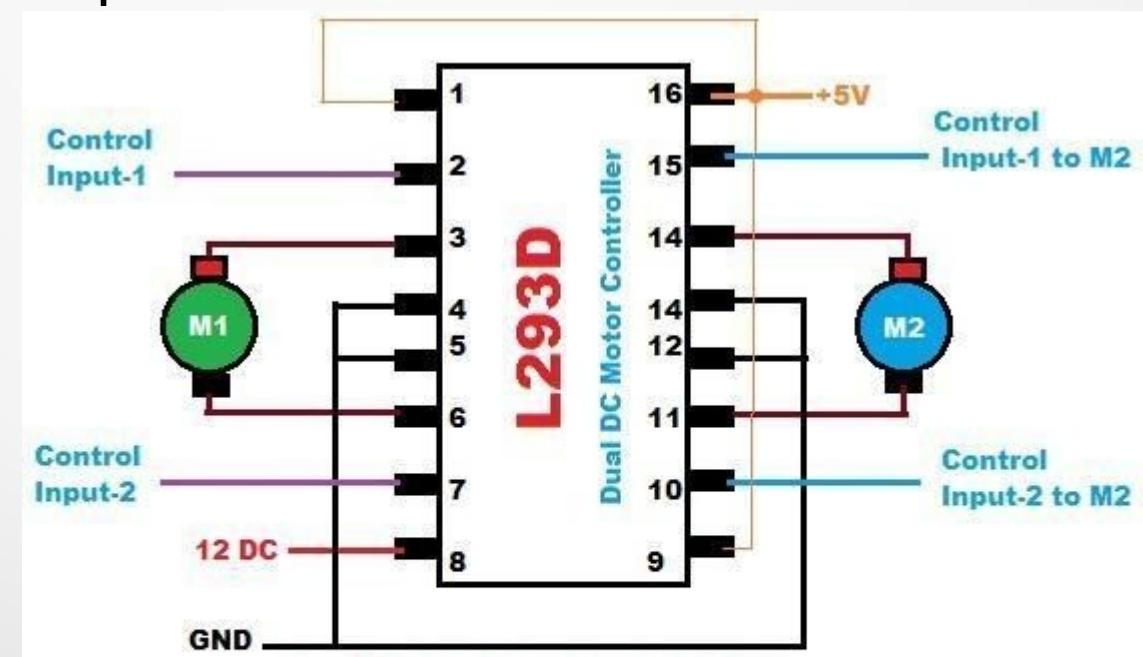
Motores DC - Relé

- Outra forma de activar/desactivar un motor DC é cun relé. Este elemento non é máis que un conmutador activado electricamente mediante alimentación dunha bobina.
- Para activar un relé temos que alimentar os contactos da bobina a 5V. Cando esta está activada pecha os contactos NO (Normalmente abertos) e abre os NC (normalmente pechados). Nas placas de relés que se usan con Arduino, hai ademais dous LEDs: un indica que a placa ten alimentación e o outro que a bobina está activada.



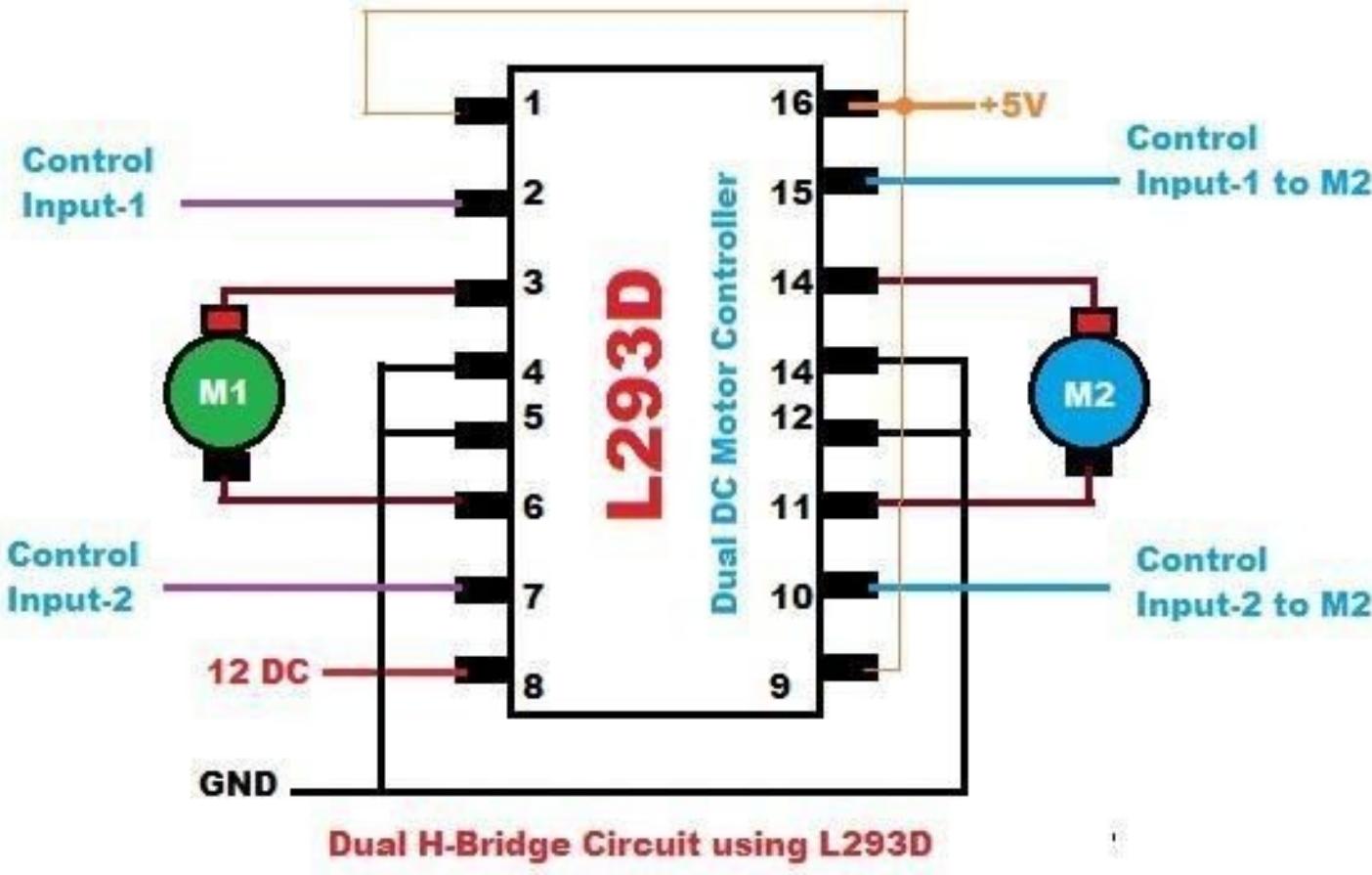
Motores DC – Integrado L293D

- Unha opción mellor para controlar un motor DC é co integrado L293D. É unha versión da ponte H, que mediante diodos permite alimentar dous motores distintos para que avancen ou retrocedan.
- Permite alimentar motores DC entre 4,5 e 36V, cunha corrente máxima de 600 mA por canal.



Datasheet L293D

Motores DC – Integrado L293D

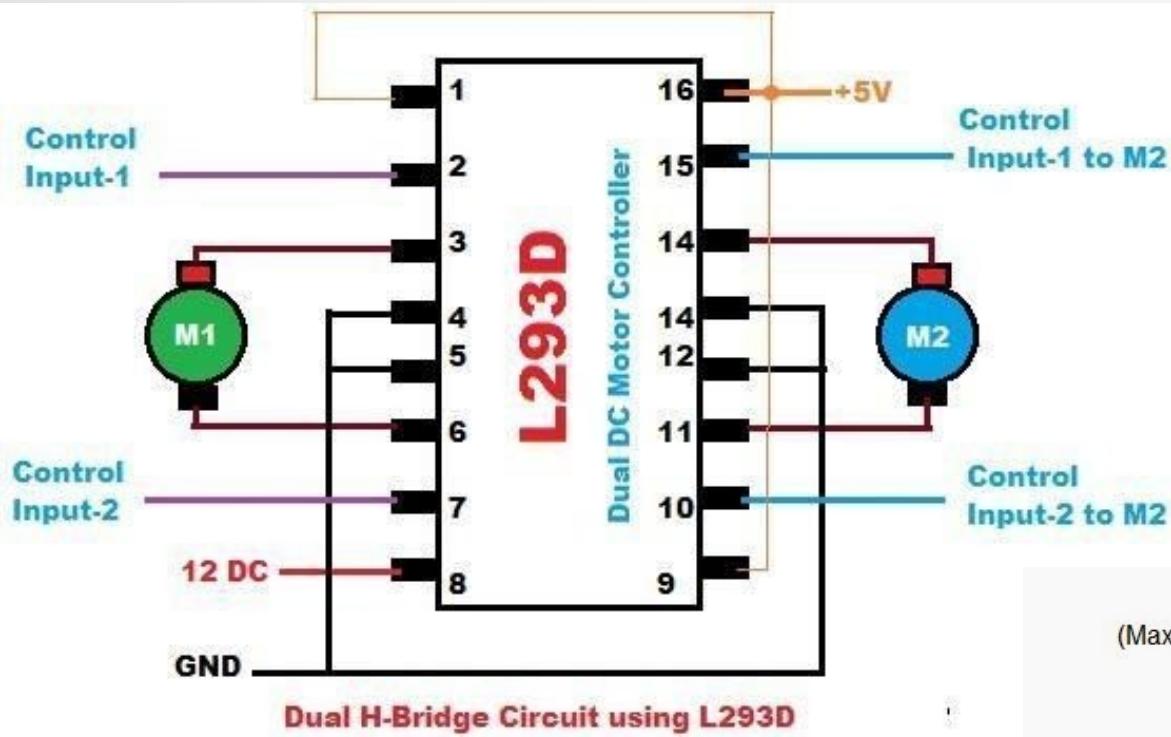


All inputs Low: Motor M1 & M2 = OFF.
Input-1 is High and Input-2 is Low: Motor M1 = Forward Direction.
Input-1 is Low and Input-2 is High: Motor M1 = Backward Direction.
Same Condition to M2.

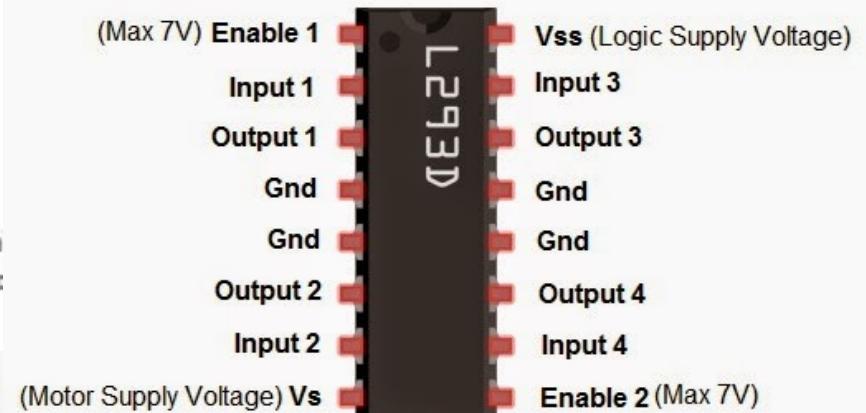
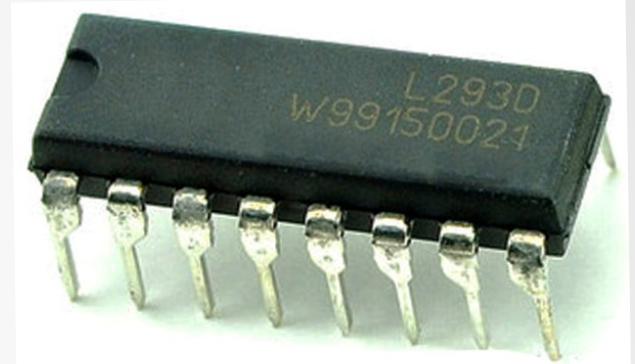


Datasheet L293D

Motores DC – Integrado L293D

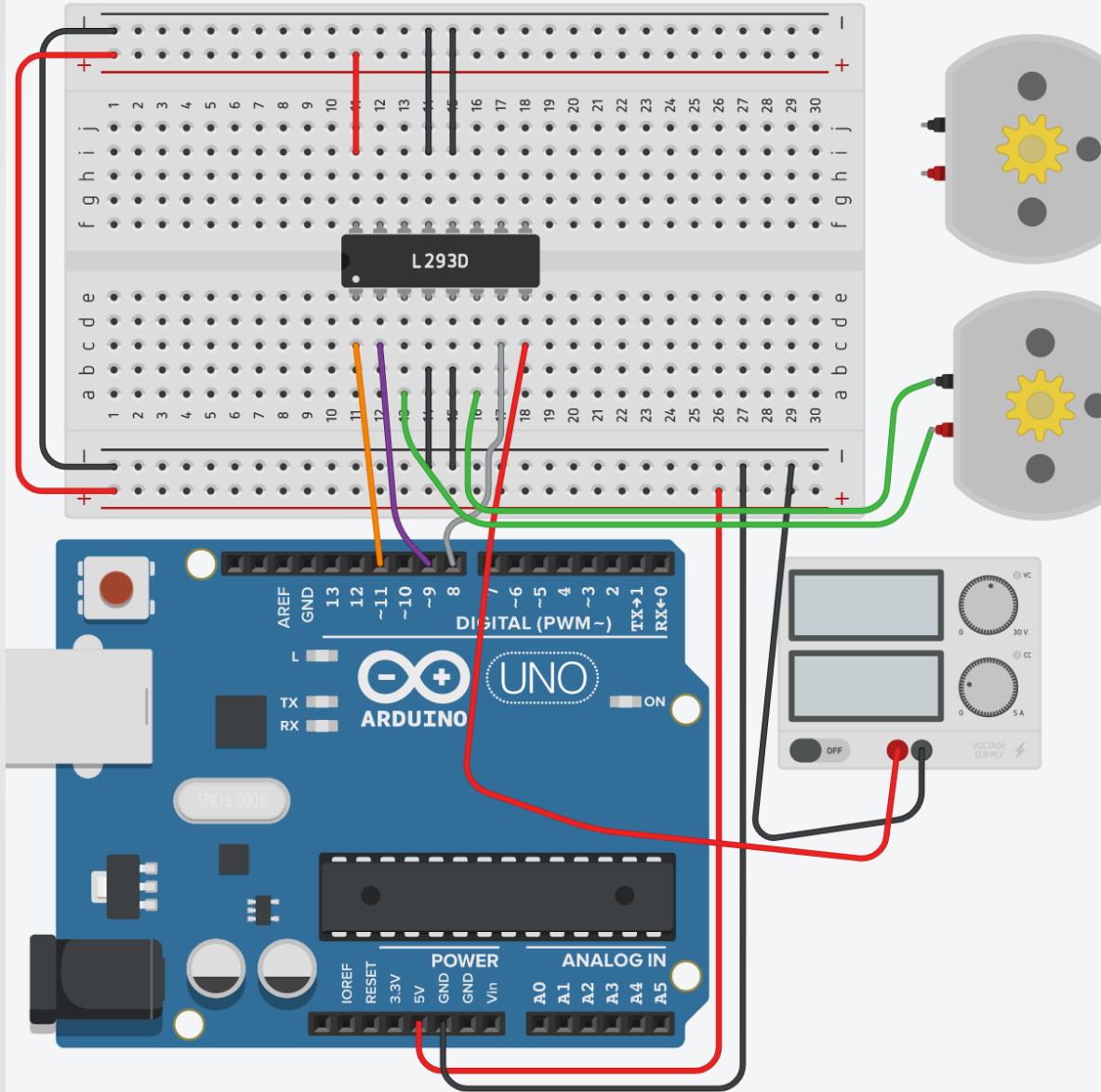


All inputs Low: Motor M1 & M2 = OFF.
Input-1 is High and Input-2 is Low: Motor M1 = Forward Direction
Input-1 is Low and Input-2 is High: Motor M1 = Backward Direction
Same Condition to M2.



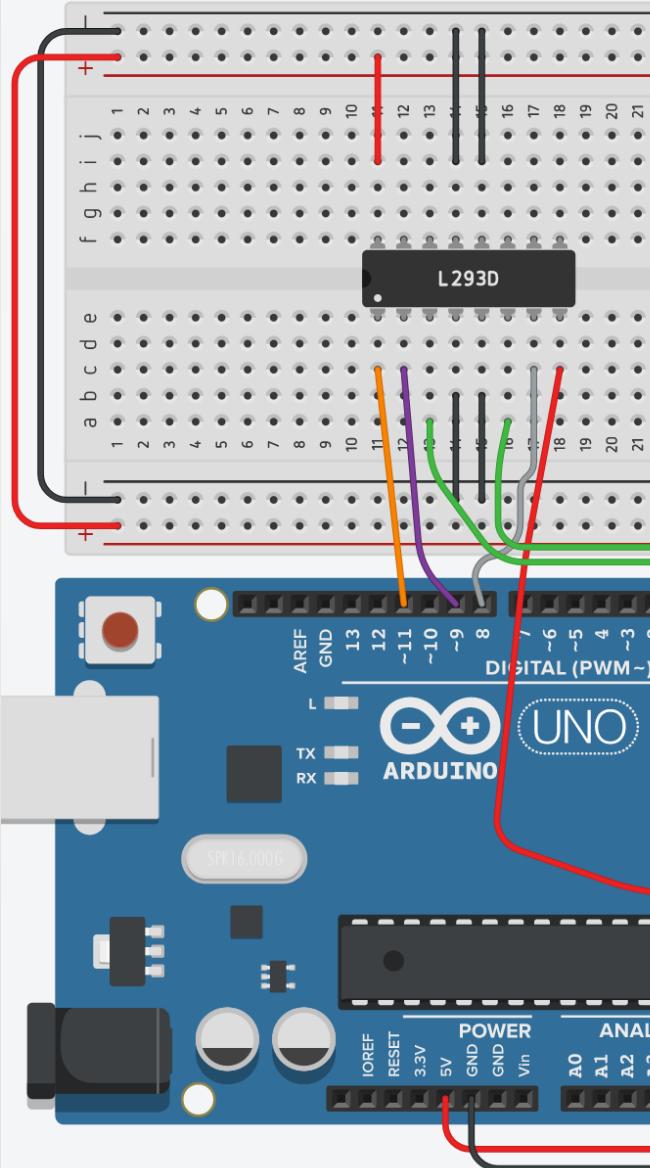
Datasheet L293D

Motores DC – Integrado L293D



- Na figura do lado está feito o conexionado dun motor DC, tendo en conta o pinout do L293D das transparencias anteriores.
- O motor aliméntase aparte a 9V, e controláse mediante os pins 11 (enable), 9 (avance) e 8 (retroceso).
- O pin 11 permite regular a velocidade do motor apagando e encendendo sucesivamente co sinal PWM.

Motores DC – Integrado L293D



motor.dc.L293D

```
/*
 * Control dun motor DC usando
 * o integrado L293D e a entrada
 * serie para introducir ordes.
 */

//Imos empregar o pin 12 tanto para
//saída dixital como para PWM
#define CONTROL 11
#define AVANCE 9
#define RETROCESO 8

int espera = 1000;

void setup() {
    pinMode(AVANCE, OUTPUT);
    pinMode(RETROCESO, OUTPUT);
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, LOW);
    Serial.begin(9600);
}

void loop() {
    //Marcha/paro do motor co voltaxe HIGH
    analogWrite(CONTROL, 255);
    digitalWrite(AVANCE, HIGH);
    digitalWrite(RETROCESO, LOW);
    Serial.println("Motor avanzando...");
    delay(5*espera);
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, HIGH);
    Serial.println("Motor retrocedendo...");
    delay(5*espera);
    digitalWrite(CONTROL, 0);
    Serial.println("Motor parado...");
    delay(5*espera);

    //Redución de velocidade do motor dende
    //max (255) a min (0) en incrementos de 25
    //... co motor avanzando
    digitalWrite(AVANCE, HIGH);
    digitalWrite(RETROCESO, LOW);
    for(int veloc = 255; veloc >= 0; veloc -= 25) {
        analogWrite(CONTROL, veloc);
        Serial.println("Veloc. en pasos PWM: " + String(veloc));
        delay(espera);
    }

    //Incremento de velocidade do motor dende
    //min (0) a max (255) en incrementos de 25
    //... co motor en retroceso
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, HIGH);
    for(int veloc = 0; veloc <= 255; veloc += 25) {
        analogWrite(CONTROL, veloc);
        Serial.println("Veloc. en pasos PWM: " + String(veloc));
        delay(espera);
    }
}
```

motor.dc.L293D

```
Serial.begin(9600);
```

```
}
```

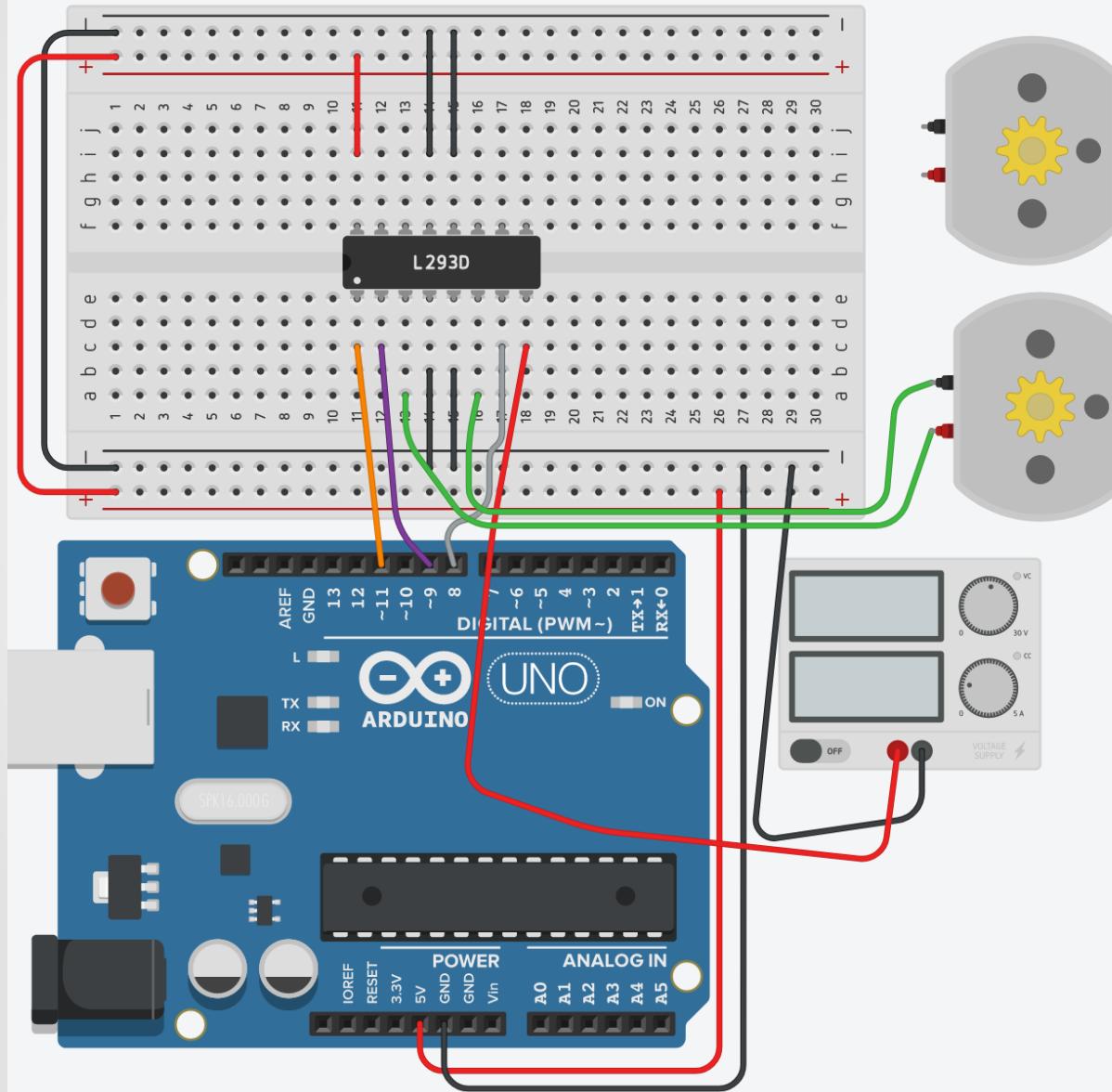
```
void loop()
```

```
    //Marcha/paro do motor co voltaxe HIGH
    analogWrite(CONTROL, 255);
    digitalWrite(AVANCE, HIGH);
    digitalWrite(RETROCESO, LOW);
    Serial.println("Motor avanzando...");
    delay(5*espera);
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, HIGH);
    Serial.println("Motor retrocedendo...");
    delay(5*espera);
    digitalWrite(CONTROL, 0);
    Serial.println("Motor parado...");
    delay(5*espera);

    //Redución de velocidade do motor dende
    //max (255) a min (0) en incrementos de 25
    //... co motor avanzando
    digitalWrite(AVANCE, HIGH);
    digitalWrite(RETROCESO, LOW);
    for(int veloc = 255; veloc >= 0; veloc -= 25) {
        analogWrite(CONTROL, veloc);
        Serial.println("Veloc. en pasos PWM: " + String(veloc));
        delay(espera);
    }

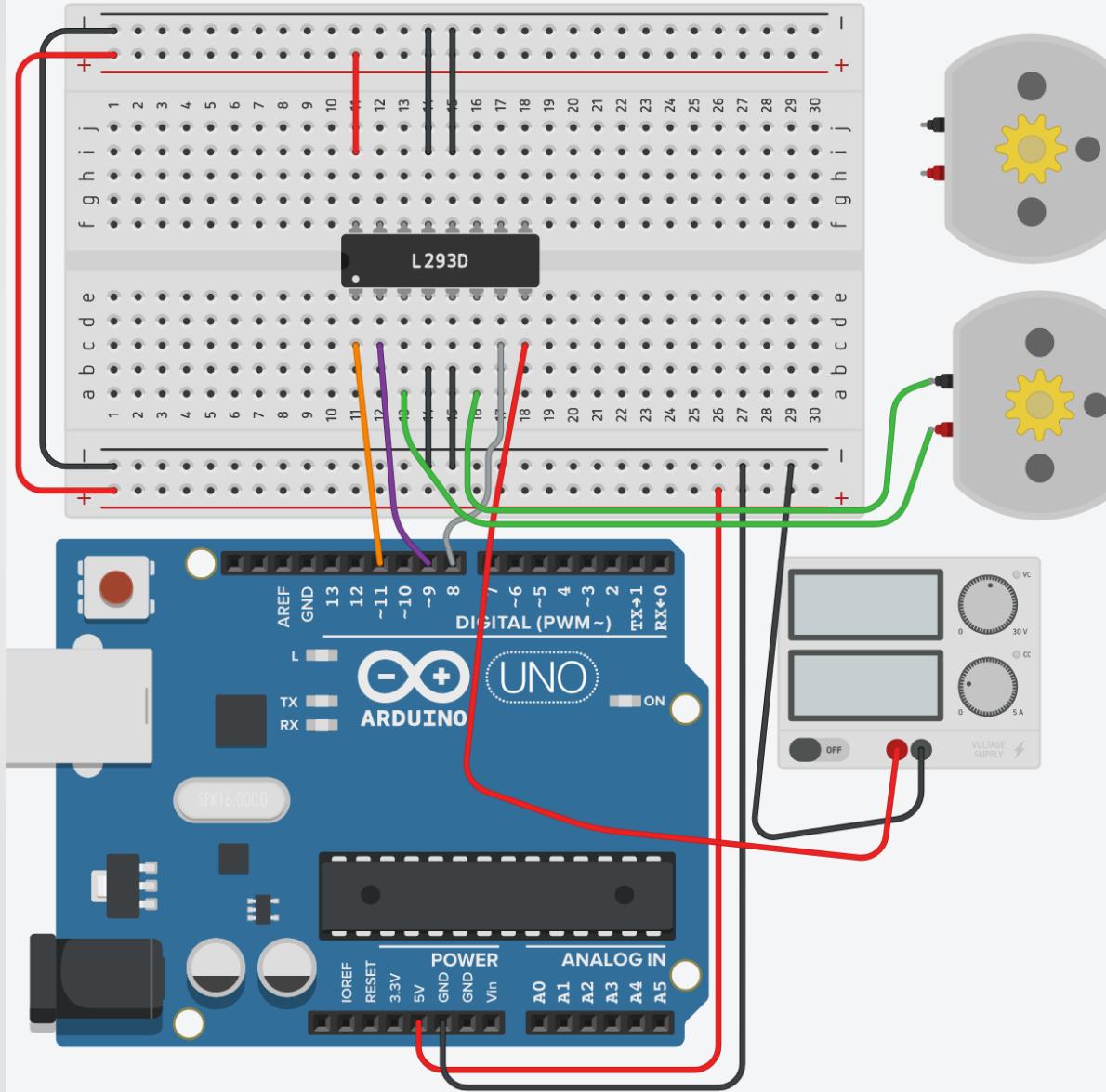
    //Incremento de velocidade do motor dende
    //min (0) a max (255) en incrementos de 25
    //... co motor en retroceso
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, HIGH);
    for(int veloc = 0; veloc <= 255; veloc += 25) {
        analogWrite(CONTROL, veloc);
        Serial.println("Veloc. en pasos PWM: " + String(veloc));
        delay(espera);
    }
}
```

Motores DC – Integrado L293D



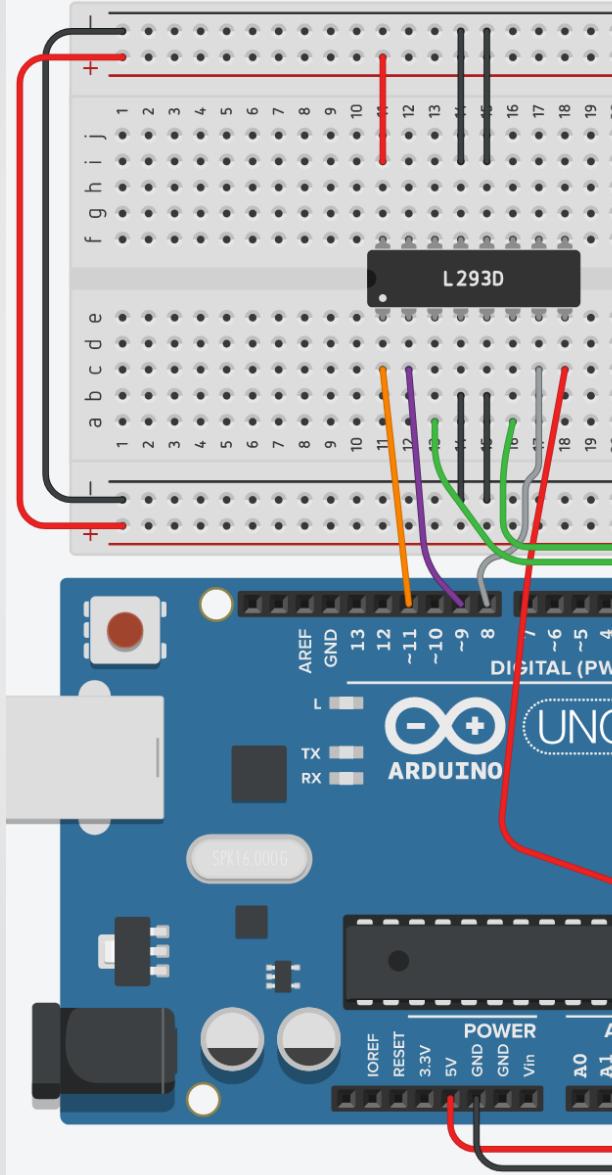
- Usando como exemplo a montaxe anterior, fai a conexión do segundo motor nos pins libres do L293D.
 - Modifica o script para controlar tamén o segundo motor, de maneira que faga o contrario do primeiro.
 - Lembra empregar un pin PWM no enable do L293D, a fin de poder mudar a velocidade do segundo motor.

Motores DC – Integrado L293D



- Na seguinte montaxe imos por un sensor de temperatura TMP36 (dos que xa usamos) para controlar un motor dun ventilador.
- Imos facer que a temperatura se divida en tramos para facer que o ventilador estea: (a) parado, (b) veloc. suave, (c) veloc. media e (d) máxima veloc.
- Para o anterior empregaremos varias veces a función map(), combinada con constrain().

Motores DC – Integrado L293D



motor.dc.L293D.con.TMP36

```
/*
 * Control dun motor DC usando
 * o integrado L293D e o sensor
 * de temperatura TMP36.
 */

//Imos empregar o pin 13 tanto para
//saída dixital como para PWM
#define CONTROL 11
#define AVANCE 9
#define RETROCESO 8

//0 pin3 da lectura de temperatura
#define TMP A0

int temp;
int veloc = 0; //Veloc. do ventilador
int espera = 2000;

void setup() {
    pinMode(AVANCE, OUTPUT);
    pinMode(RETROCESO, OUTPUT);
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, LOW);
    analogWrite(CONTROL, 0);
    Serial.begin(9600);
}

void loop() {
    //Lectura do sensor de temperatura
    temp = map(analogRead(TMP), 0, 1023, 0, 5000);
    temp = (temp - 500) / 10; //Calcula temp en °C
    Serial.print("Temperatura: " + String(temp) + " oC");
    //Restrinxe temp entre 15 e 35 °C
    temp = constrain(temp, 15, 35);
    //... para facer catro intervalos:
    temp = map(temp, 15, 35, 0, 4);
    //A partir de 15 °C aumenta a velocidade
    //do ventilador en tres intervalos:
    digitalWrite(AVANCE, HIGH);
    digitalWrite(RETROCESO, LOW);
    switch(temp) {
        //temp até 19 °C: ventilador parado
        case 0: veloc = 0; break;
        //temp entre 20 °C e 24 °C: ventilador suave
        case 1: veloc = 140; break;
        //temp entre 25 °C e 29 °C: ventilador medio
        case 2: veloc = 212; break;
        //temp más de 30 °C: ventilador a tope
        case 3:
        case 4: veloc = 255; break;
    }
    analogWrite(CONTROL, veloc);
    Serial.println("\tVeloc. ventilador: " + String(veloc));
    delay(espera);
}
```

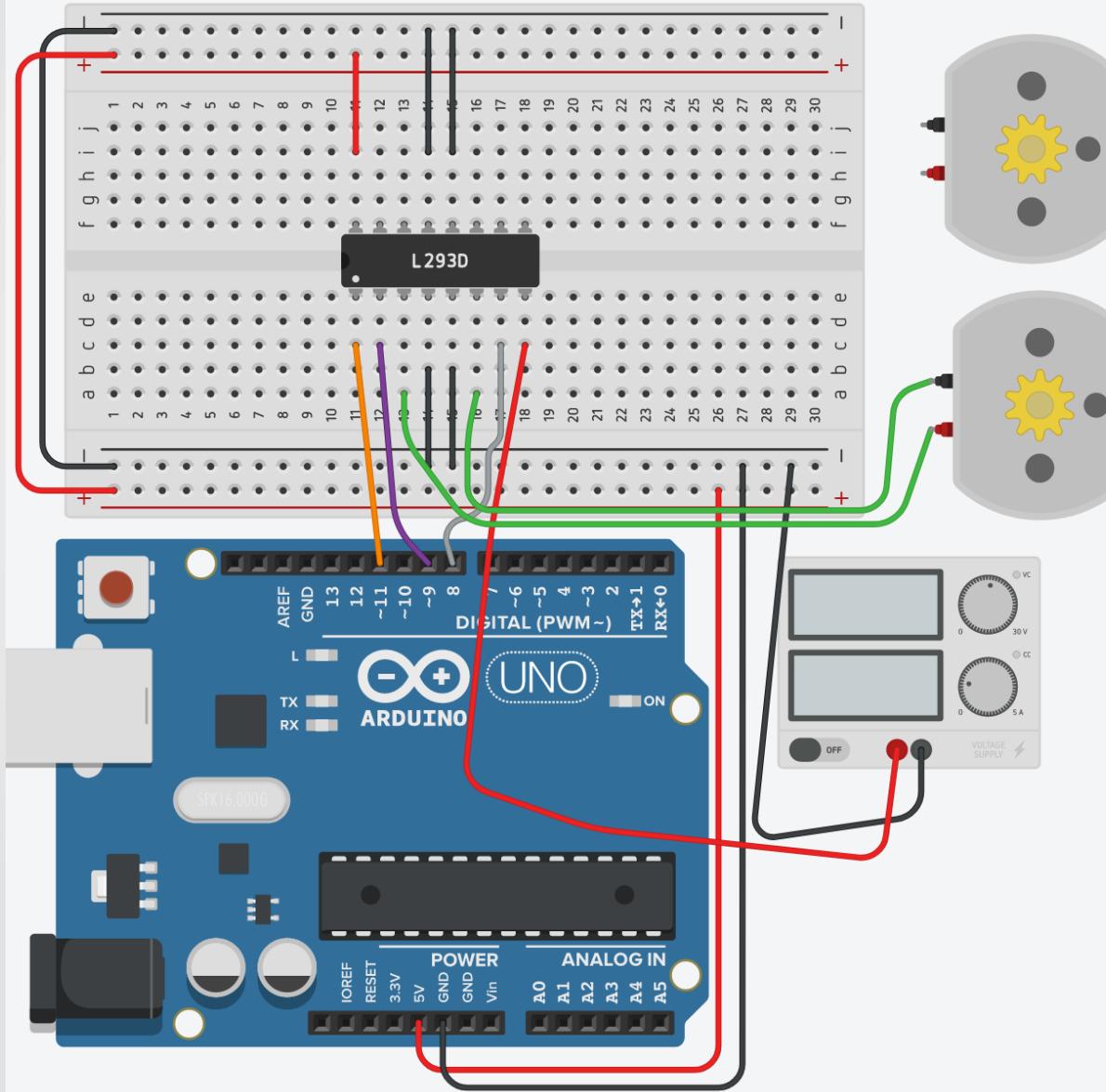
motor.dc.L293D.con.TMP36

```
int veloc = 0; //Veloc. do ventilador
int espera = 2000;

void setup() {
    pinMode(AVANCE, OUTPUT);
    pinMode(RETROCESO, OUTPUT);
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, LOW);
    analogWrite(CONTROL, 0);
    Serial.begin(9600);
}

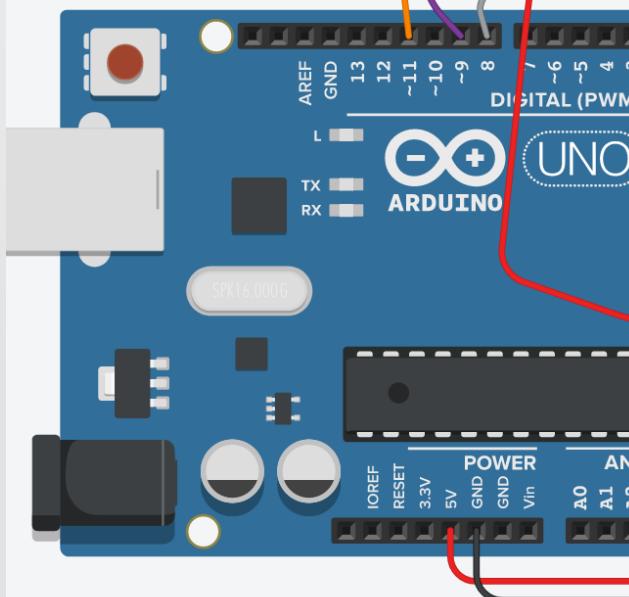
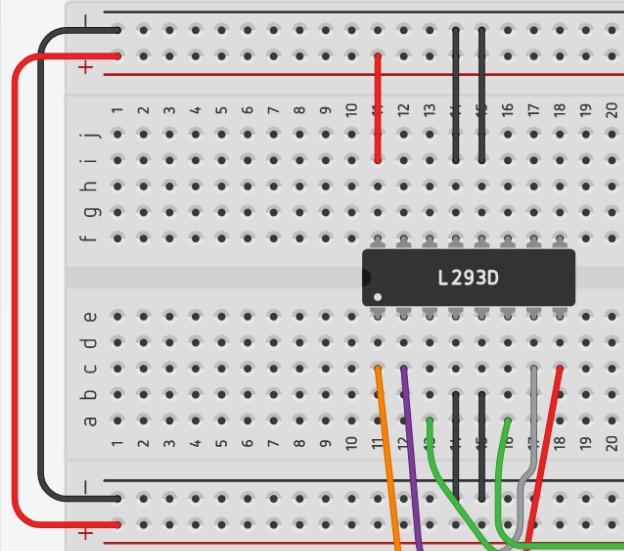
void loop() {
    //Lectura do sensor de temperatura
    temp = map(analogRead(TMP), 0, 1023, 0, 5000);
    temp = (temp - 500) / 10; //Calcula temp en °C
    Serial.print("Temperatura: " + String(temp) + " oC");
    //Restrinxe temp entre 15 e 35 °C
    temp = constrain(temp, 15, 35);
    //... para facer catro intervalos:
    temp = map(temp, 15, 35, 0, 4);
    //A partir de 15 °C aumenta a velocidade
    //do ventilador en tres intervalos:
    digitalWrite(AVANCE, HIGH);
    digitalWrite(RETROCESO, LOW);
    switch(temp) {
        //temp até 19 °C: ventilador parado
        case 0: veloc = 0; break;
        //temp entre 20 °C e 24 °C: ventilador suave
        case 1: veloc = 140; break;
        //temp entre 25 °C e 29 °C: ventilador medio
        case 2: veloc = 212; break;
        //temp más de 30 °C: ventilador a tope
        case 3:
        case 4: veloc = 255; break;
    }
    analogWrite(CONTROL, veloc);
    Serial.println("\tVeloc. ventilador: " + String(veloc));
    delay(espera);
}
```

Motores DC – Integrado L293D



- En vez de controla a velocidade do ventilador usando un sensor adicional, imos facer o mesmo enviando ordes pola entrada serie.
- De maneira que o ventilador estea: (a) parado, (b) veloc. suave, (c) veloc. media e (d) máxima veloc., (e) en avance ou retroceso e (f) con velociade seleccionada polo usuario numericamente.

Motores DC – Integrado L293D



motor.dc.L293D.e.entrada.serie

```
/*
 * Control dun motor DC usando
 * o integrado L293D e a comunicación
 * de ordes pola entrada serie.
 *
 * As ordes disponibles son:
 * (a) parado ou stop,
 * (b) suave,
 * (c) medio,
 * (d) tope ou full,
 * (e) directo,
 * (f) inverso, ou
 * (g) número entre 0 e 255
 */

//Imos empregar o pin 13 tanto para
//saída dixital como para PWM
#define CONTROL 13
#define AVANCE 11
#define RETROCESO 10
```

```
int veloc = 0; //Veloc. do ventilador
int espera = 600;
String orde = "";
//Direccion de avance do motor
bool directo = true;

void setup() {
    pinMode(AVANCE, OUTPUT);
    pinMode(RETROCESO, OUTPUT);
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, LOW);
    analogWrite(CONTROL, 0);
    Serial.begin(9600);
}

void loop() {
    //Lectura da entrada serie
    if(Serial.available()) {
        orde = Serial.readStringUntil('\n');
```

motor.dc.L293D.e.entrada.serie

```
int veloc = 0; //Veloc. do ventilador
int espera = 600;
String orde = "";
//Direccion de avance do motor
bool directo = true;

void setup() {
    pinMode(AVANCE, OUTPUT);
    pinMode(RETROCESO, OUTPUT);
    digitalWrite(AVANCE, LOW);
    digitalWrite(RETROCESO, LOW);
    analogWrite(CONTROL, 0);
    Serial.begin(9600);}
```

void loop()

```
//Lectura da entrada serie
if(Serial.available()) {
    orde = Serial.readStringUntil('\n');

    //Actualización da veloc. do ventilador
    digitalWrite(AVANCE, directo);
    digitalWrite(RETROCESO, !directo);
    analogWrite(CONTROL, veloc);
    Serial.print("Veloc. ventilador: " + String(veloc));
    Serial.println("\t\tAvance motor: " + String(directo));
    delay(espera);}
```

Motores DC

- Nesta unidade aprendemos a:
 - controlar a marcha e paro de motores DC empregando transistores e relés,
 - cablear e alimentar un integrado L293D (ponte H) para controlar motores DC, en particular:
 - marcha e paro de até dous motores DC,
 - avance ou retroceso dos mesmos,
 - velocidade de avance ou retroceso dos mesmos mediante un sinal PWM sobre os pins enable do integrado
 - combinar a velocidade do motor cun sinal analóxico a partir de sensores,
 - combinar a velocidade do motor con ordes textuais e numéricas, lidas a partir da entrada serie de Arduino.