

ARDUINO

CONTROL DE MOTORES  
Y BLUETOOTH

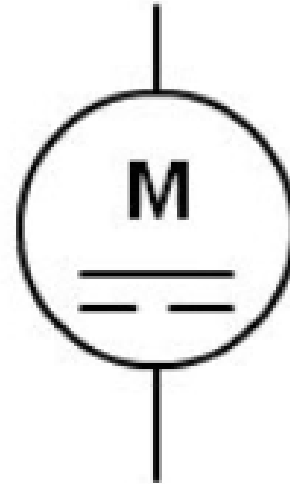
Alvaro Acosta Agón

*Magister en Educación*  
*Magister en Educación*

# CONTROL DE MOTORES

El motor DC es una máquina capaz de convertir la energía eléctrica en mecánica provocando un movimiento rotatorio.

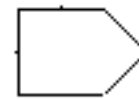
Si se alimenta el motor DC con la tensión nominal (corriente continua) el motor empieza a girar. Si se baja la tensión de alimentación el motor girará más despacio y si se intercambia la polaridad girará en sentido contrario.



# CONTROL DE UN MOTOR DC

Se necesita una fuente de alimentación externa (9V). El diodo sirve de protección para el Arduino que se alimenta con 5 Voltios, dejando que los 9 Voltios solo circule para alimentar al motor.

Pin a Arduino



1 K $\Omega$

Base

Colector

Emisor

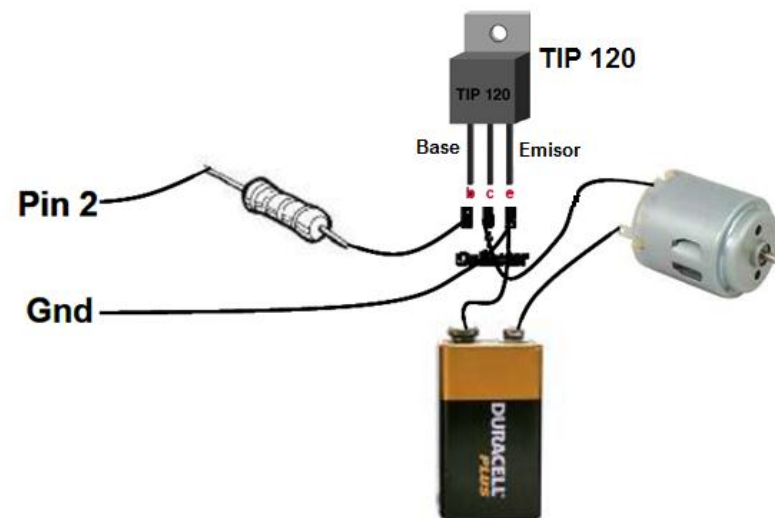
Diodo

Motor

9 V

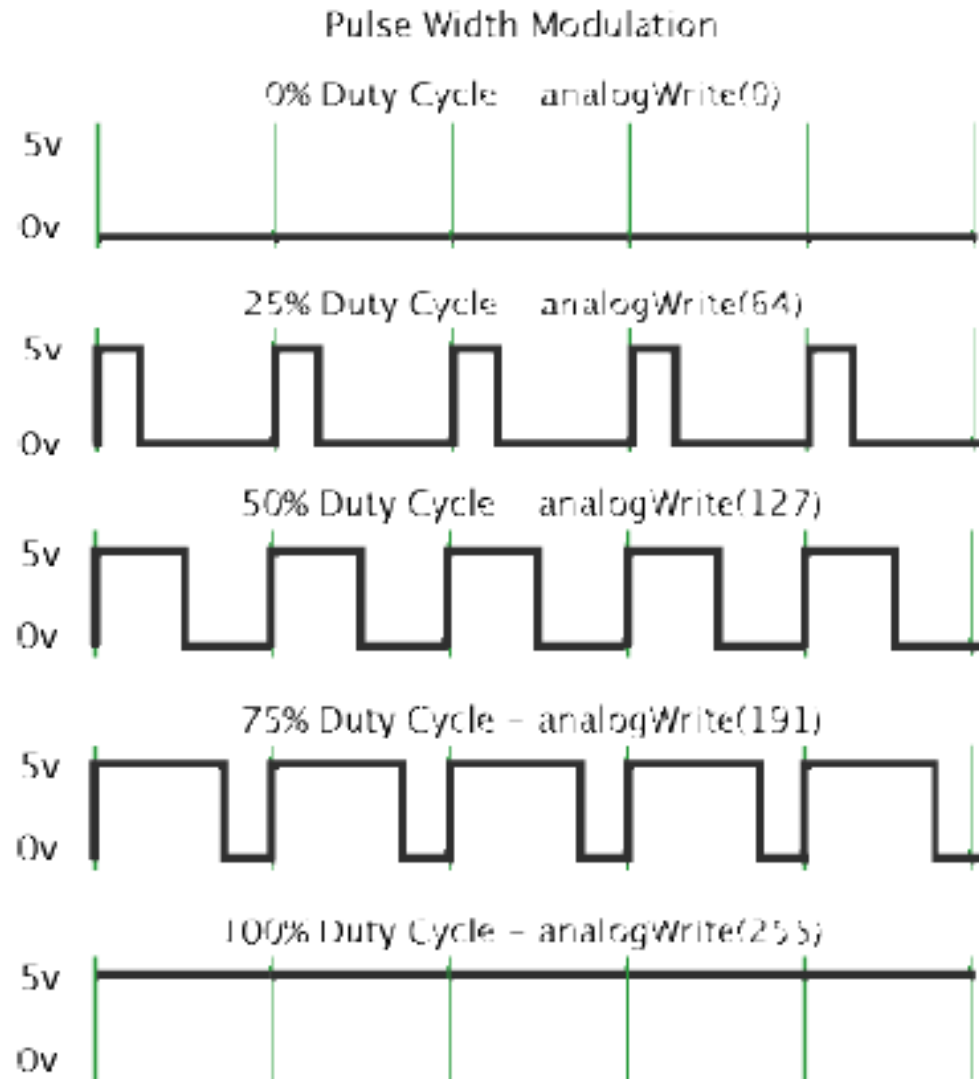
La tierra debe estar conectada a la tierra del Arduino, pues si no el circuito se desestabiliza. Esto es una regla de montajes: las tierras o masas deben estar siempre conectadas.

El transistor funciona como interruptor.  
Tiene una resistencia para protegerlo.



# P W M

El **ancho de pulso de modulación** (PWM) es una técnica utilizada para simular una salida analógica con una digital, la creación de una onda cuadrada que cambia constantemente entre encendido y apagado. Cuando el tiempo de la onda es de 5V (ON) se denomina ancho de pulso, que se modifica para cambiar el análogo valor.



# APLICACIÓN DEL PWM (1)

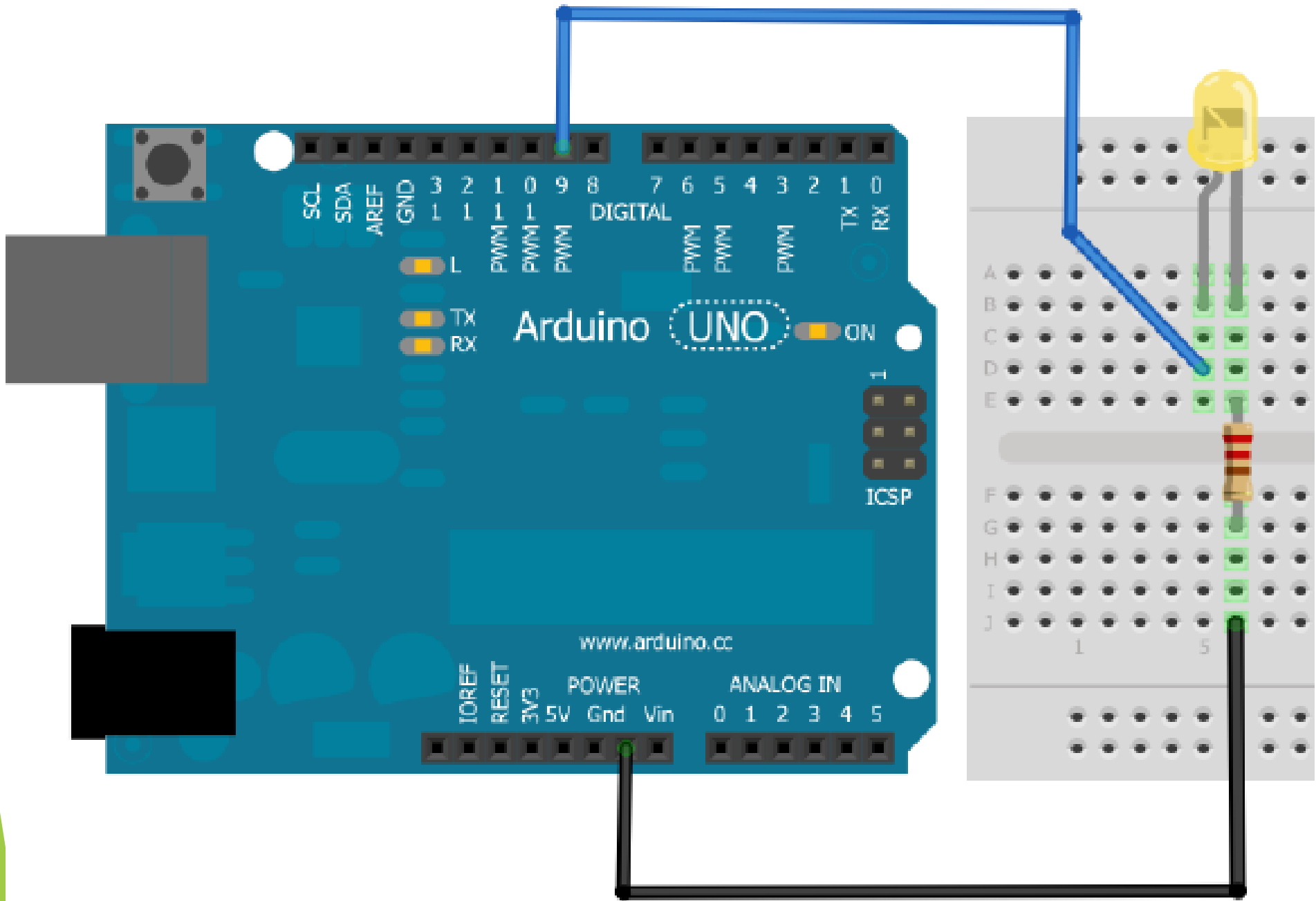
La Modulación por ancho de pulso (PWM) es una forma de cambiar una salida analógica por la salida pulsante. Esto se puede aplicar (entre otros) en:

- ATENUAR LA ILUMINACIÓN EN UN LED
- CONTROLAR UN MOTOR

```
int ledPin = 9;           //pin PWM para el LED

void setup(){}

void loop()
{
  for(int i=0; i<=255; i++) //incrementa el valor para i
  {
    analogWrite(ledPin, i); //asigna el nivel de brillo a i
    delay(100);             //pausa 100 ms
  }
  for(int i=255; i>=0; i--) //decrementa el valor para i
  {
    analogWrite(ledPin, i); //asigna el nivel de brillo a i
    delay(100);             //pausa 100 ms
  }
}
```



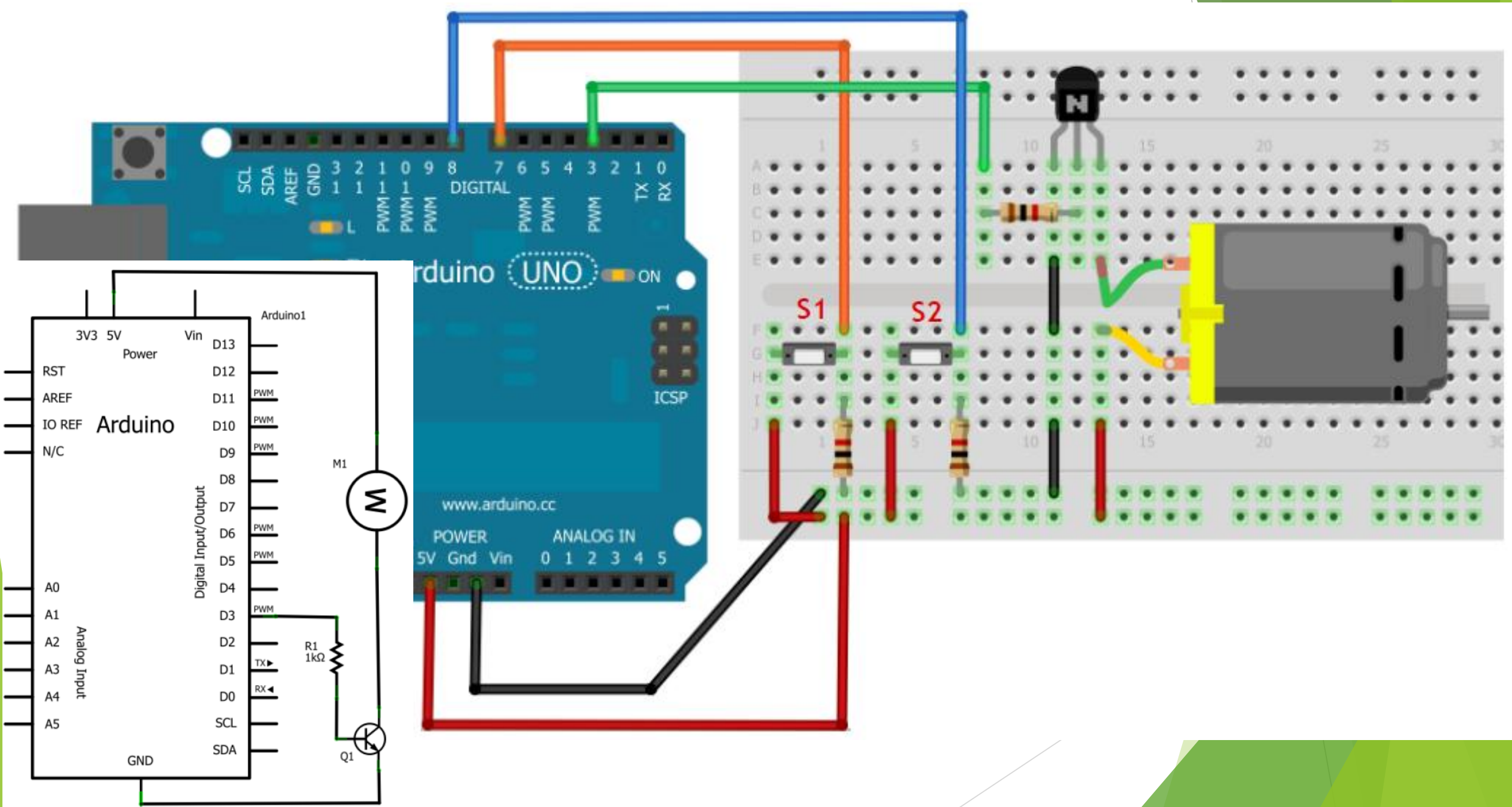
# APLICACIÓN DEL PWM (2)

Para el envío de un pulso al Motor se utiliza la instrucción `analogWrite(pin, pulso)`.

Los valores serán de 0 a 255, ya que **PWM** es una “simulación” de analógico, en la que 0 es 0 Voltios y 255 es 5 Voltios. Va de velocidad 0 a las máximas revoluciones:

```
int pulso=0;           //variable donde se almacena el valor de pulso
int pinMotor=3;       //Pin 3 que puede generar PWM

void setup()
{
}
void loop()
{
  for(pulso=0; pulso<=255; pulso ++ ) //ciclo para ir subiendo el voltaje de 0 a 5 Voltios
  {analogWrite(pinMotor, pulso);      //enviar el pulso al motor vía PWM
  delay(15);
  }
  delay(600);
}
```





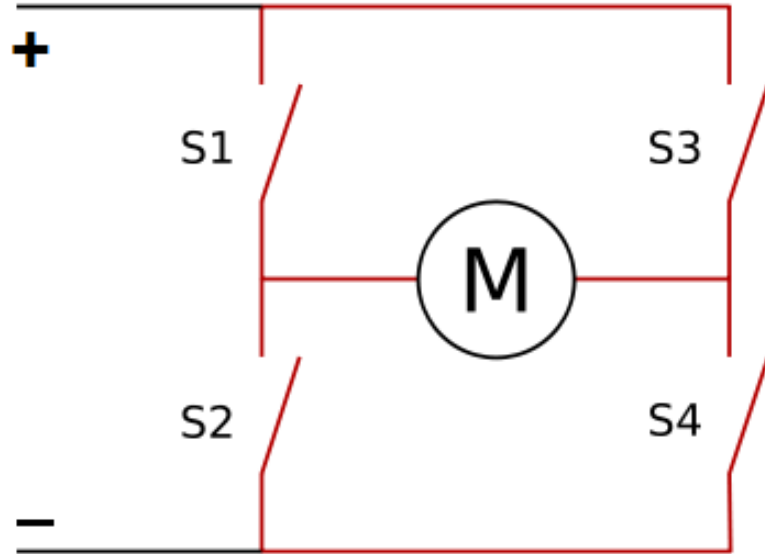
# APLICACIÓN DEL PWM (3)

Configura el motor 5 velocidades distintas, desde el teclado del PC

```
-----  
int motor=3;      //Declara Pin del motor  
  
//-----  
//Funcion principal  
//-----  
void setup() // Se ejecuta cada vez que el Arduino se inicia  
{  
  Serial.begin(9600); //Inicia la comunicacion serial Arduino-PC  
}  
  
//-----  
//Funcion ciclicla  
//-----  
void loop() // Esta funcion se mantiene ejecutando  
{ // cuando este energizado el Arduino  
  
  // Si hay algun valor en la Consola Serial  
  if (Serial.available()){  
  
    //Variable donde se guarda el caracter enviado desde teclado  
    char a = Serial.read();  
  
    // Si el caracter ingresado esta entre 0 y 5  
    if (a>='0' && a<='5'){  
  
      //Variable para escalar el valor ingresado a rango de PWM  
      int velocidad = map(a, '0', '5', 0, 255);  
      //Escritura de PWM al motor  
      analogWrite(motor, velocidad);  
      //Mensaje para el usuario  
      Serial.print("El motor esta girando a la velocidad ");  
      Serial.println(a);  
  
    }else{ // Si el caracter ingresado NO esta entre 0 y 5  
  
      //Mensaje para el usuario  
      Serial.print("Velocidad invalida");  
      Serial.println(a);  
  
    }  
  
  }  
  
}  
  
}
```

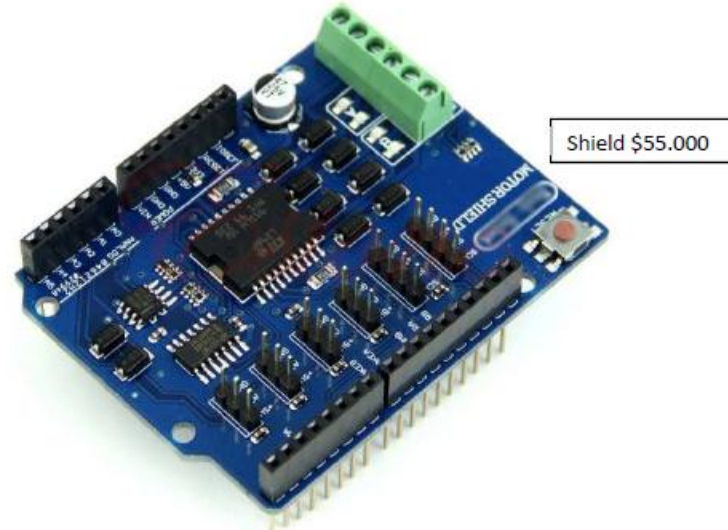
# PUENTE H

Para controlar el Motor DC se utiliza en Puente H.



Si se cierran S1 y S4 el motor gira en un sentido  
Si se cierran S2 y S3 el motor gira en sentido contrario

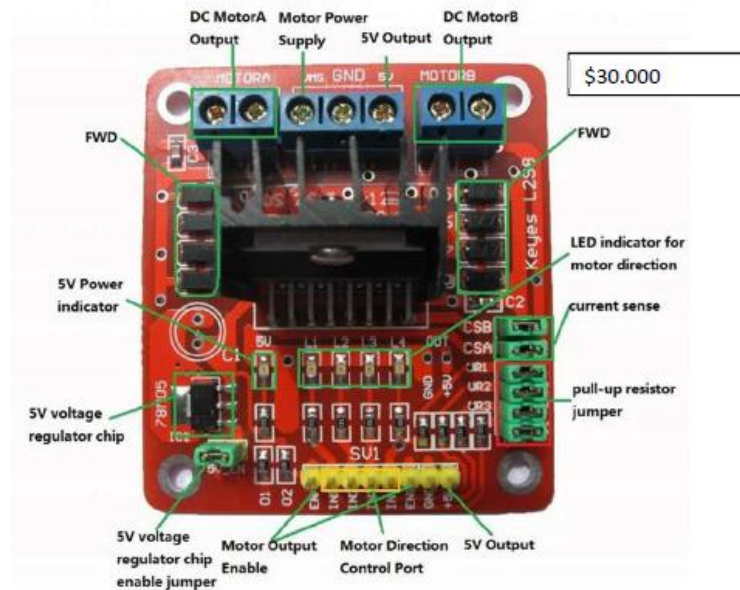
# Tarjetas Puente H



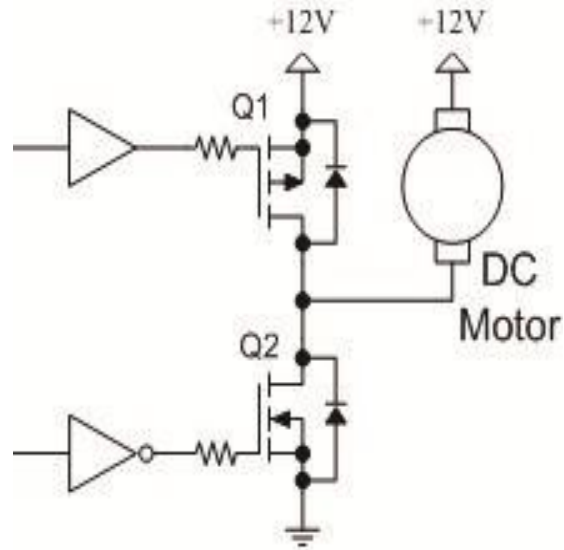
ARDUMOTO



Motor DC

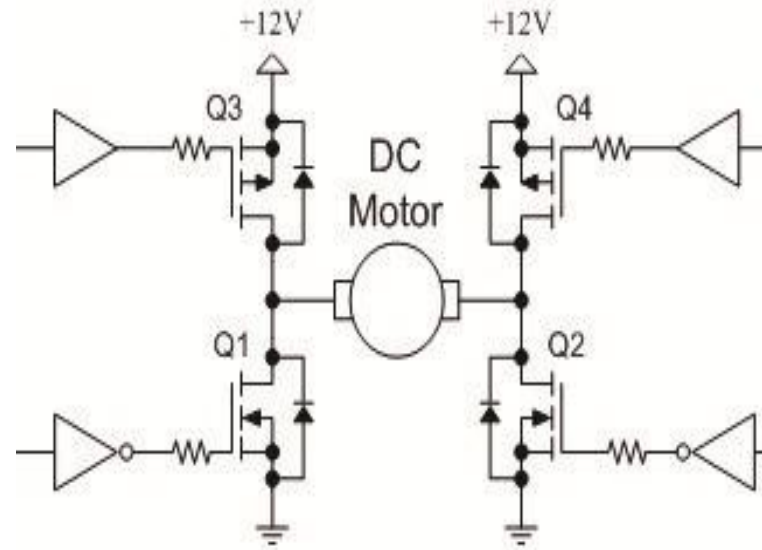


## MEDIO PUENTE H



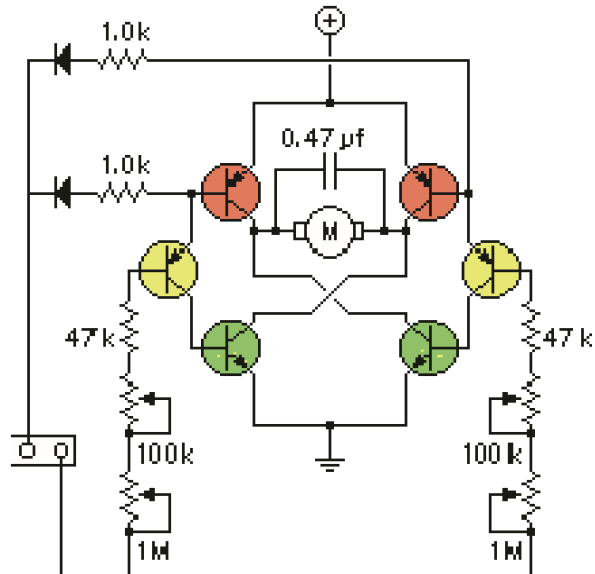
Los medios puentes son usados en diseños en los cuales no es necesario revertir la dirección de rotación del motor. Es decir, lo único que se requiere es un control de la velocidad en un solo sentido.

## FULL BRIDGE



Los puentes H completos pueden ser usados en los casos en los que revertir esta dirección es o no necesaria. Con este tipo de puentes H es posible realizar control de posición y/o velocidad además de permitir realizar frenado dinámico.

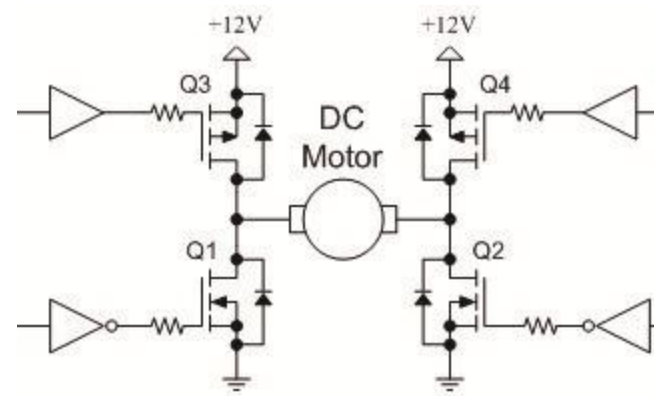
# PUENTE H - con BJT



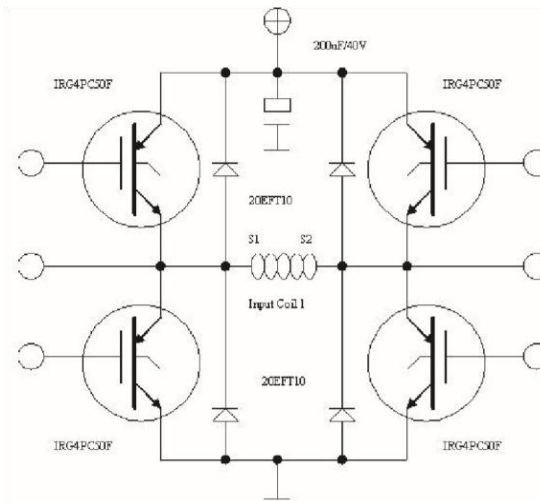
Los puentes H con BJTs son la opción rápida, son robustos, fáciles de diseñar, y fáciles de controlar. Su única limitación radica en la baja potencia que soportan y en su muy baja eficiencia

# PUENTE H - con IGBTs

# PUENTE H - con MOSFETs



Los puentes H con MOSFETs, son una de las opciones preferidas para el control de motores de mediano tamaño como los usados en robótica, poseen una muy baja resistencia de encendido, haciendo al diseño más eficiente y dándole la capacidad de soportar mayores corrientes.

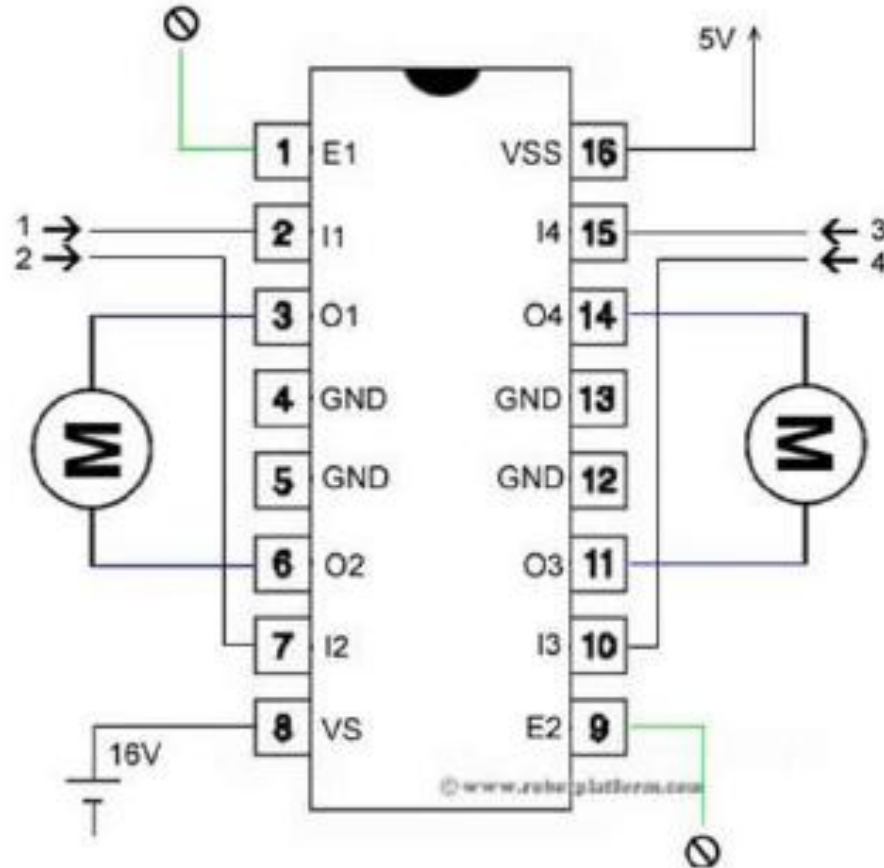


Los puentes H con IGBTs representan la elección cuando de potencia se trata, estos son los preferidos en la industria, debido al alto voltaje que pueden soportar y a su alta potencia

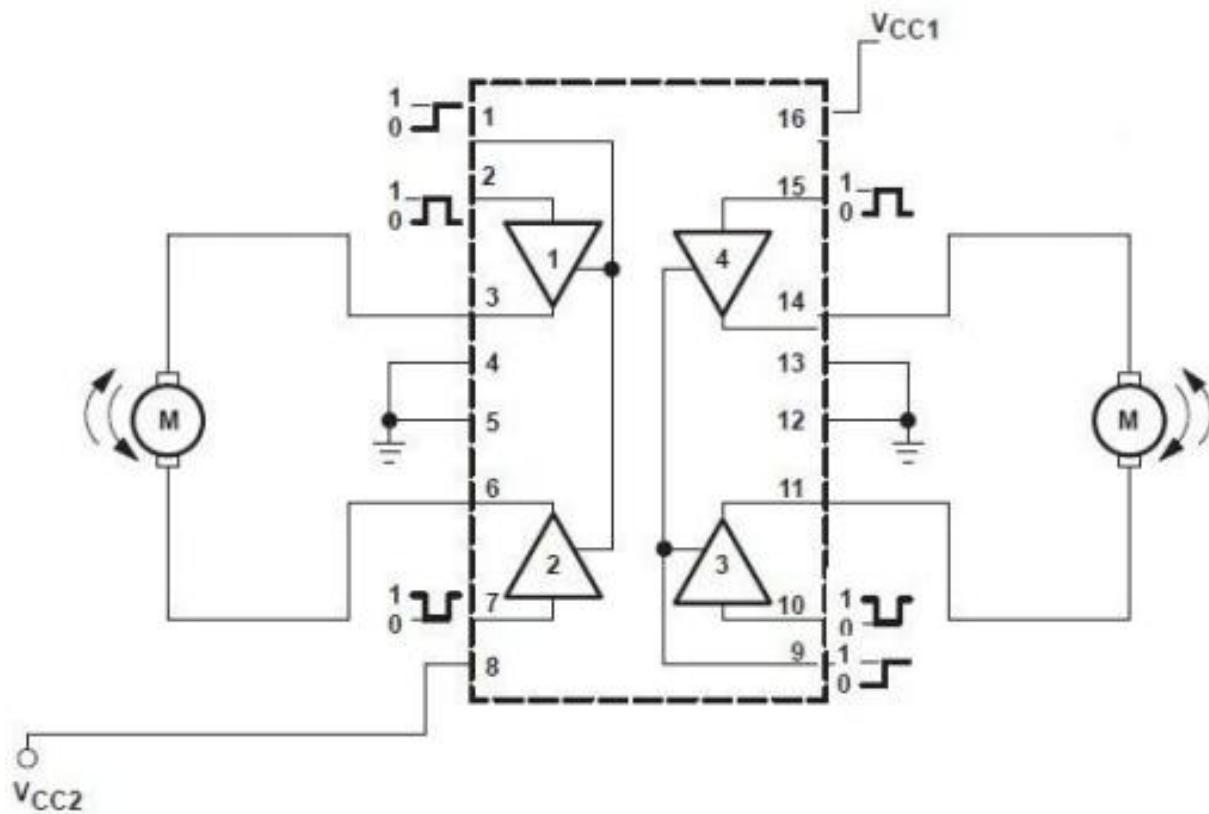
# L293D

Este chip permite controlar motores DC (cuatro de forma unidireccional o dos de forma bidireccional).

La diferencia entre los modelos L293D y L293B es que el primero viene con diodos de protección que evita los daños producidos por los picos de voltaje que puede producir el motor.



- Input from Microcontroller : 4 Inputs - I1, I2, I3, I4
- ⊖ Enable Pins : E1 & E2
- M Output to 2 Motors : O1, O2 & O3, O4
- GND Connect to microcontroller ground



**El L293D** tiene:

4 pines digitales (2,7,10,15) para controlar la dirección de los motores.

Los pines “enable” (1,9) admiten como entrada una señal PWM y se utiliza para controlar la velocidad de los motores con la técnica de modulación de ancho de pulso.

Los motores van conectados entre uno de los pines 3-6 o 11-14.

La tensión Vss es la que alimentará o dará potencia al motor.

**Pin 1:** activa o desactiva un motor (HIGH o LOW). También sirve para especificar la velocidad de giro si recibe una señal PWM.

**Pin 2:** envía la señal de giro (HIGH o LOW) en un sentido para un motor

**Pin 3:** donde se conecta uno de los terminales del motor

**Pin 4 y 5:** GND

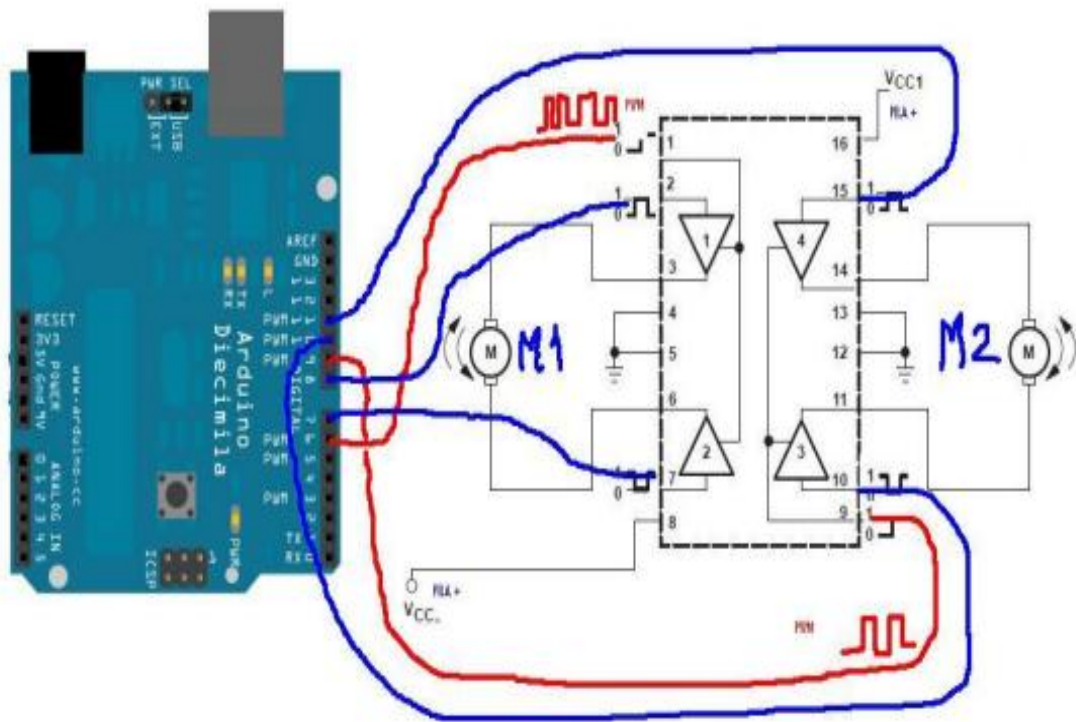
**Pin 6:** donde se conecta el otro terminal de un motor

**Pin 7:** envía la señal de giro (HIGH o LOW) en el otro sentido para un motor

**Pin 8:** alimentación del motor

**Pin 9-11:** si no se usa un segundo motor, pueden estar desconectados

**Pin 16:** alimentación del propio chip (5V)



<i>Conexiones arduino</i>	<i>Conexiones circuito L293</i>	<i>Funciones</i>
Pin 6 , salida PWM motor A (Tren de pulsos modulados en anchura)	Entrada Pin 1, habilita la transferencia de potencia a motor A	Para controlar velocidad Motor A
Pin 7 (dirección borna 1 motor A)	Pin 7	(dirección borna 1 motor A)
Pin 8 (dirección borna 2 motor A)	Pin 2	(dirección borna 2 motor A)
Pin 9 , salida PWM motor B (Tren de pulsos modulados en anchura)	Entrada Pin 9, habilita la transferencia de potencia a motor A	Para controlar velocidad Motor B
Pin 10 (dirección borna 1 motor B)	Pin 10	(dirección borna 1 motor B)
Pin 11 (dirección borna 2 motor B)	Pin 15	(dirección borna 2 motor B)
Negativo arduino (gnd)	4,5,12 y 13 unidas al gnd del arduino y al negativo de la pila	Los negativos del arduino, L293D y Pila deben estar unidos.
	Pines 8 y 16	Conectar al + de la pila, <b>no conectar al + del Arduino</b> pues crea interferencias y el usb del ordenador no tiene suficiente potencia para mover los motores
	Salidas 3 y 6 del L293D al motor A	Conexión del motor A
	Salidas 14 y 11 del L293D al motor B	Conexión del motor B

```

char val;
int PWMA = 6; //velocidad motor A
int dirmotorA1 = 7; // direccion motor a borna 1
int dirmotorA2= 8; // direccion motor a borna2

int PWMB = 9; //velocidad motor B
int dirmotorB1 = 10; // direccion motor b borna 1
int dirmotorB2= 11; // direccion motor b borna2

int velocidad = 120;

void adelante(){
digitalWrite (dirmotorA1,HIGH);// gira motor A derecha
digitalWrite (dirmotorA2,LOW);
analogWrite (PWMA, velocidad);

digitalWrite (dirmotorB1,LOW);// gira motor B izquierda
digitalWrite (dirmotorB2,HIGH);
analogWrite (PWMB, velocidad);
}

void atras(){
digitalWrite (dirmotorA1,LOW);// gira motor A izquierda
digitalWrite (dirmotorA2,HIGH);
analogWrite (PWMA, velocidad);

digitalWrite (dirmotorB1,HIGH);// gira motor B derecha
digitalWrite (dirmotorB2,LOW);
analogWrite (PWMB, velocidad);
}

void izquierda(){
digitalWrite (dirmotorA1,HIGH)// gira motor A derecha
digitalWrite (dirmotorA2,LOW);
analogWrite (PWMA, velocidad);

digitalWrite (dirmotorB1,HIGH)// gira motor B derecha
digitalWrite (dirmotorB2,LOW);
analogWrite (PWMB, velocidad);
}

```

```

void derecha(){
digitalWrite (dirmotorA1,LOW);// gira motor A izquierda
digitalWrite (dirmotorA2,HIGH);
analogWrite (PWMA, velocidad);

digitalWrite (dirmotorB1,LOW)// gira motor B izquierda
digitalWrite (dirmotorB2,HIGH);
analogWrite (PWMB, velocidad);
}

void paro(){
digitalWrite (dirmotorA1,HIGH)// para motor A
digitalWrite (dirmotorA2,HIGH);
analogWrite (PWMA, 0);

digitalWrite (dirmotorB1,HIGH)// para motor B
digitalWrite (dirmotorB2,HIGH);
analogWrite (PWMB, 0);
}

void setup()
{
int i;
for(i=6;i<12;i++)
{
pinMode(i, OUTPUT); //poner pin 6,7,8,9,10,11 de salida
}
Serial.begin(9600);

paro();
}

```

```

void loop()
{
if (Serial.available() )
{
val = Serial.read();
}
switch (val) {

case 's':{
paro();
break;

}

case 'a':{
adelante();
break;
}

case 'r':{
atras();
break;
}

case 'l':{
izquierda();
break;
}

case 'd':{
derecha();
break;
}
}
}

```



# Tabla de Verdad

Enable	Input 1	Input 2	Output 1	Output 2	Comportamiento del motor
1	1	1	0	0	Parado
1	1	0	1	0	Avanza
1	0	1	0	1	Retrocede
1	0	0	0	0	Parado
0	dc	dc	0	0	Parado

Esta sería la tabla de verdad de la mitad del integrado L293. La otra es simétrica a esta.

# GIRO DE UN MOTOR AUTOMÁTICAMENTE

Gira el motor automáticamente en un tiempo determinado

```
int input1 = 9; // entrada 1
int input2 = 6; //entrada 2
int velocidad = 7; // pin de habilitacion
void setup()
{
  pinMode(input1, OUTPUT);
  pinMode(input2, OUTPUT);
  pinMode(velocidad, OUTPUT);
}
void loop()
{
  //avanza por dos segundos
  digitalWrite(velocidad, HIGH);
  digitalWrite(input1 , HIGH);
  digitalWrite(input2, LOW);
  delay(2000);

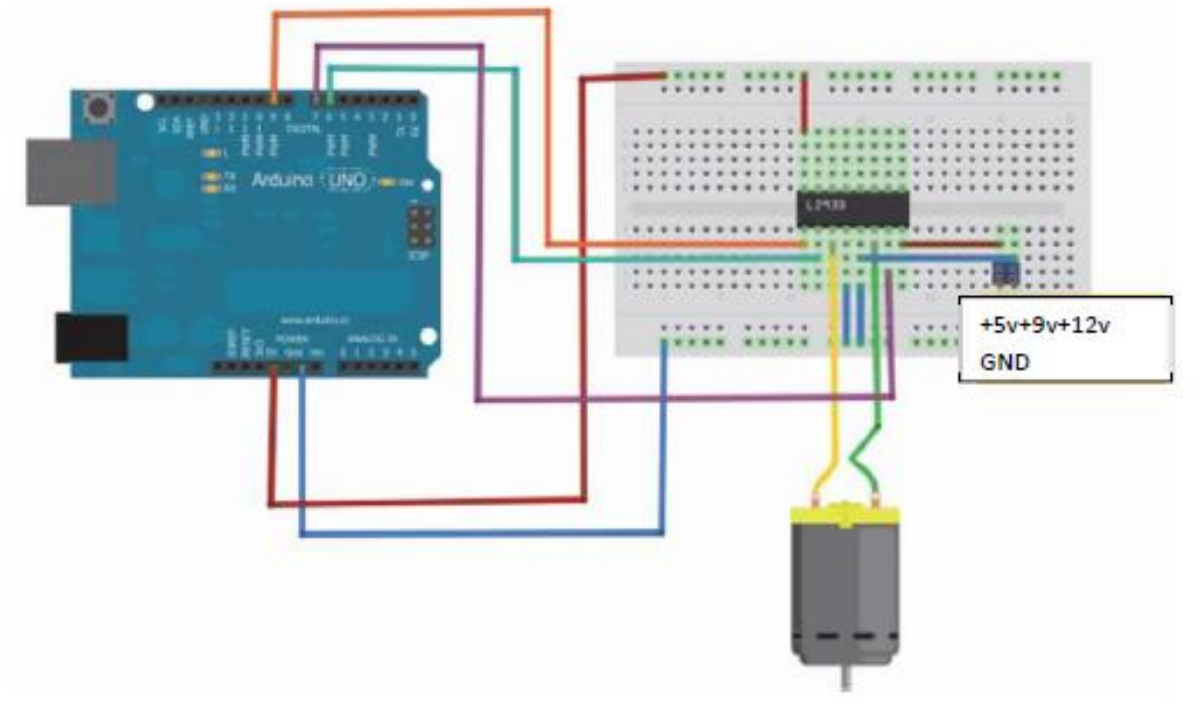
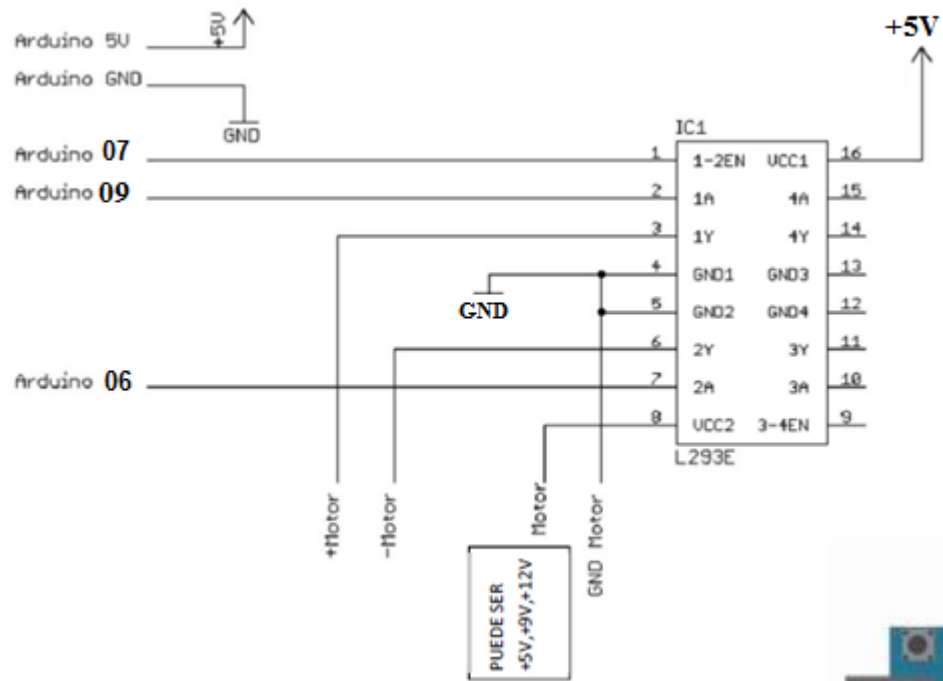
  //retrocede por 2 segundos
  digitalWrite(input1 , LOW);
  digitalWrite(input2, HIGH);
  delay(2000);
  // para el motor por 1 segundo
  digitalWrite(velocidad, LOW);
  delay(1000);
}
```

# ENVÍO DE SEÑAL POR EL PUERTO SERIAL PARA CONTROLAR EL MOTOR

```
int input1 = 9; // entrada 1
int input2 = 6; //entrada 2
int velocidad = 7; // pin de habilitacion
void setup()
{
  Serial.begin(9600);
  pinMode(input1, OUTPUT); //M1
  pinMode(input2, OUTPUT); //M1
  pinMode(velocidad, OUTPUT); //M1
}
void loop()
{
  char i;
  if (Serial.available())
  {
    i = Serial.read(); //leer dato y lo guarada en i
    switch (i)
    {
      case 'a': //adelante
        Serial.println("adelante");
        digitalWrite(velocidad, HIGH);
        digitalWrite(input1 , HIGH);
        digitalWrite(input2, LOW);
        break;

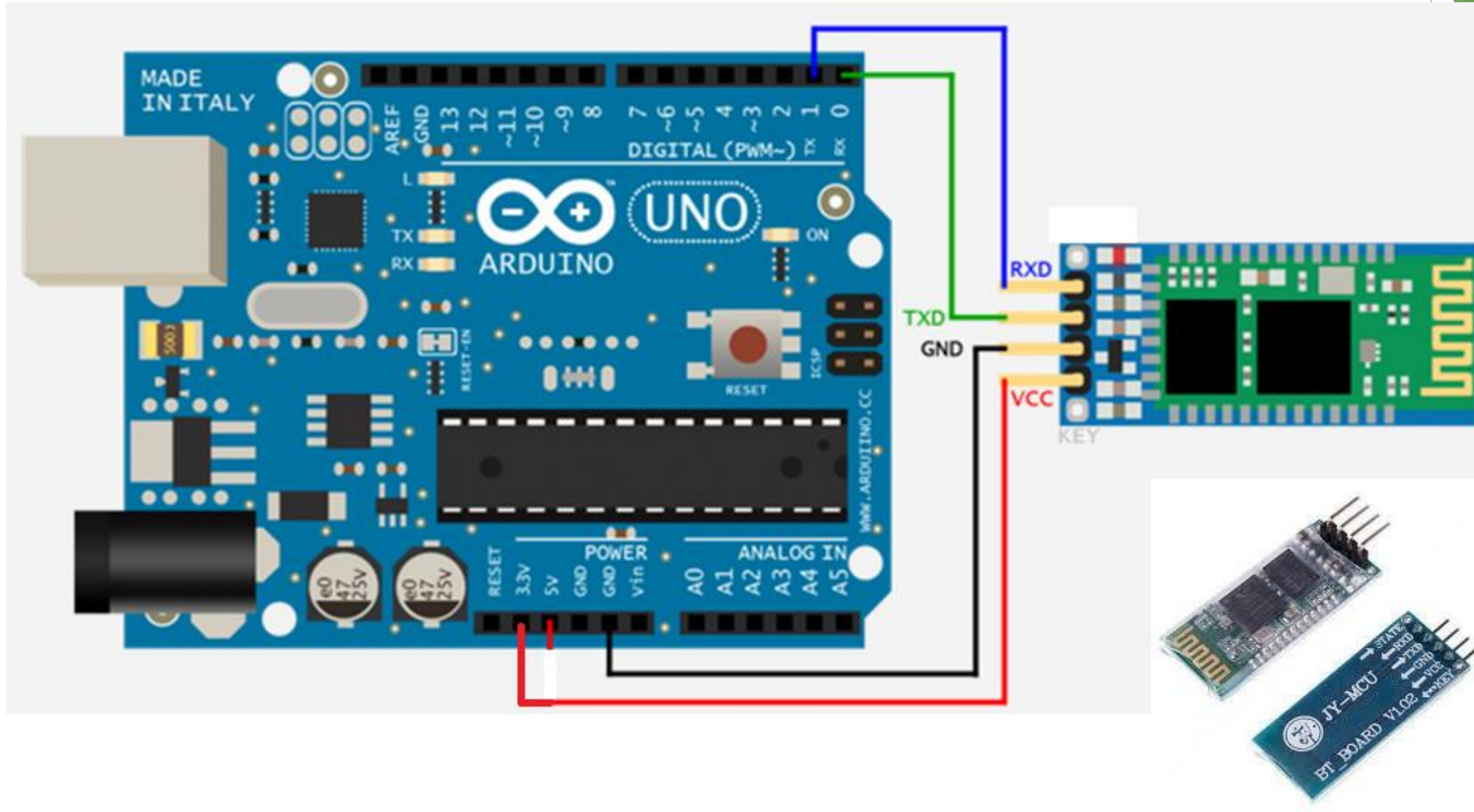
      case 'd': //atras
        Serial.println("atras");
        digitalWrite(velocidad, HIGH);
        digitalWrite(input1 , LOW);
        digitalWrite(input2, HIGH);
        break;

      case 'p': //parada
        Serial.println("parada");
        digitalWrite(velocidad, LOW);
        break;
    }
  }
}
```



# **RETO 5: CONTROL DEL MOTOR POR BLUETOOTH**

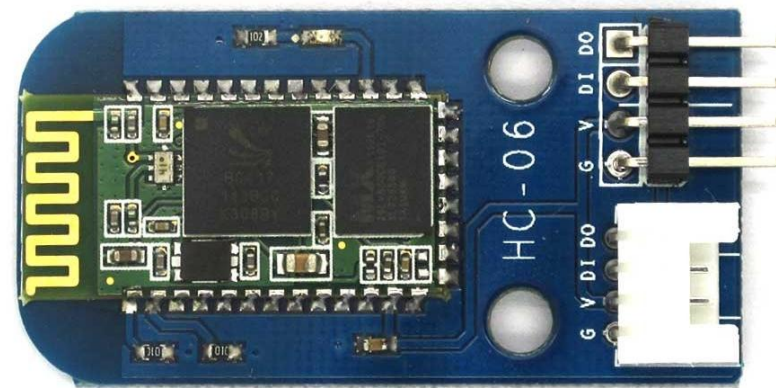
# Configuración y Conexión Bluetooth

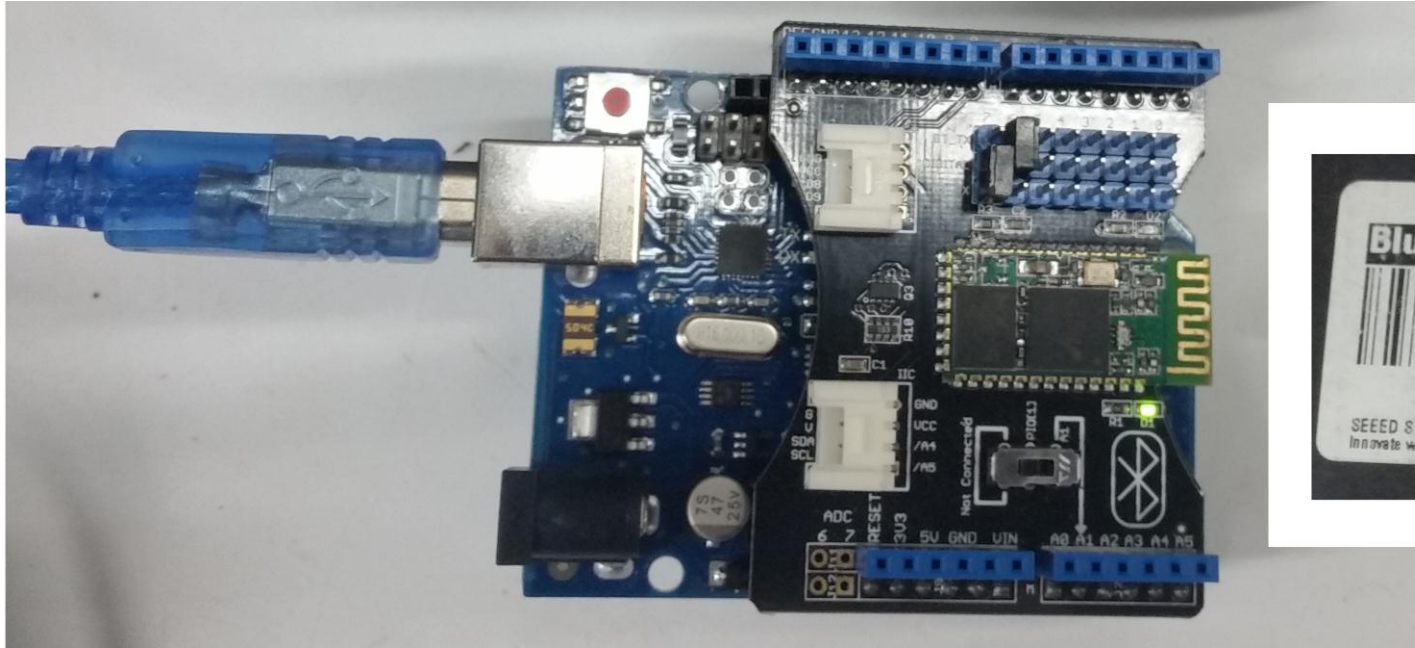


<http://www.youtube.com/watch?v=fokloKBNCOE#t=17>



Alcance de hasta 10m en línea de vista.  
Este módulo de la marca Roving Networks es poderoso, pequeño y muy fácil de utilizar. Se encuentra diseñado para reemplazar los cables seriales. El bluetooth se encuentra totalmente encapsulado e incluye antena.  
Operación de 3.3V a 5V.







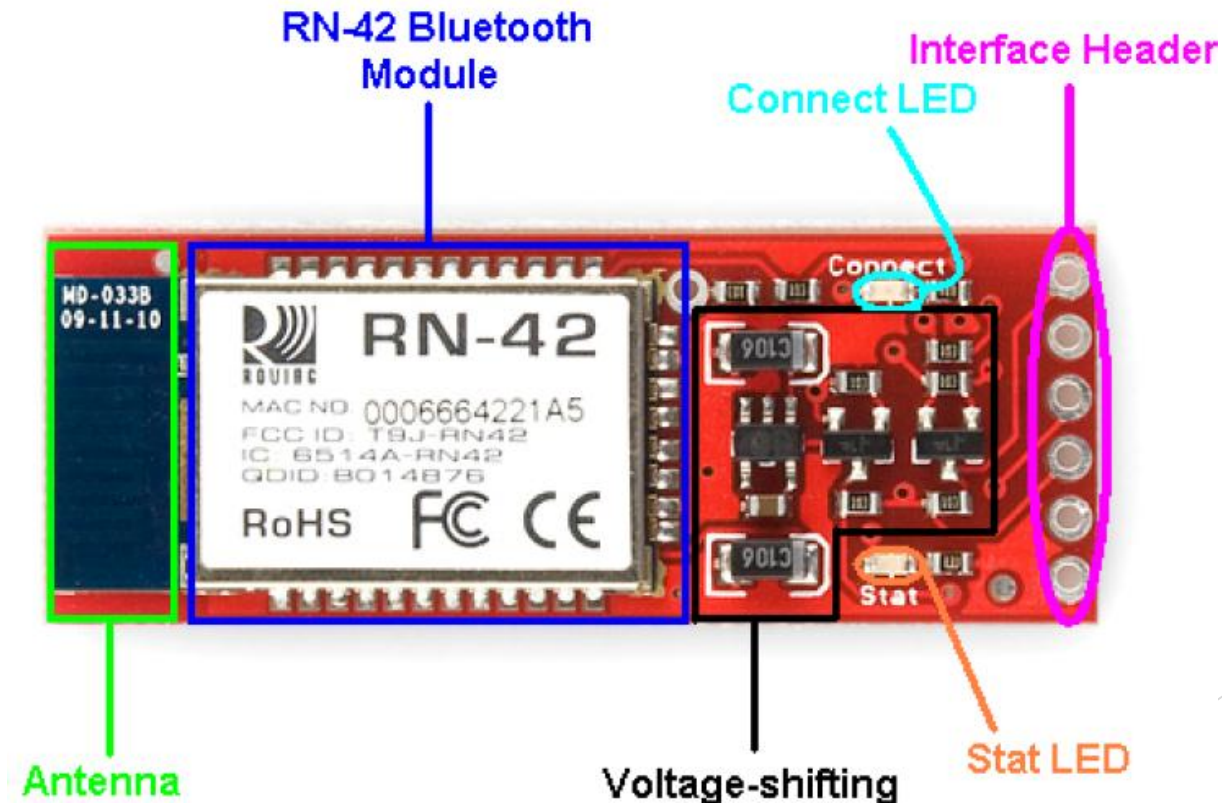
# RN-42

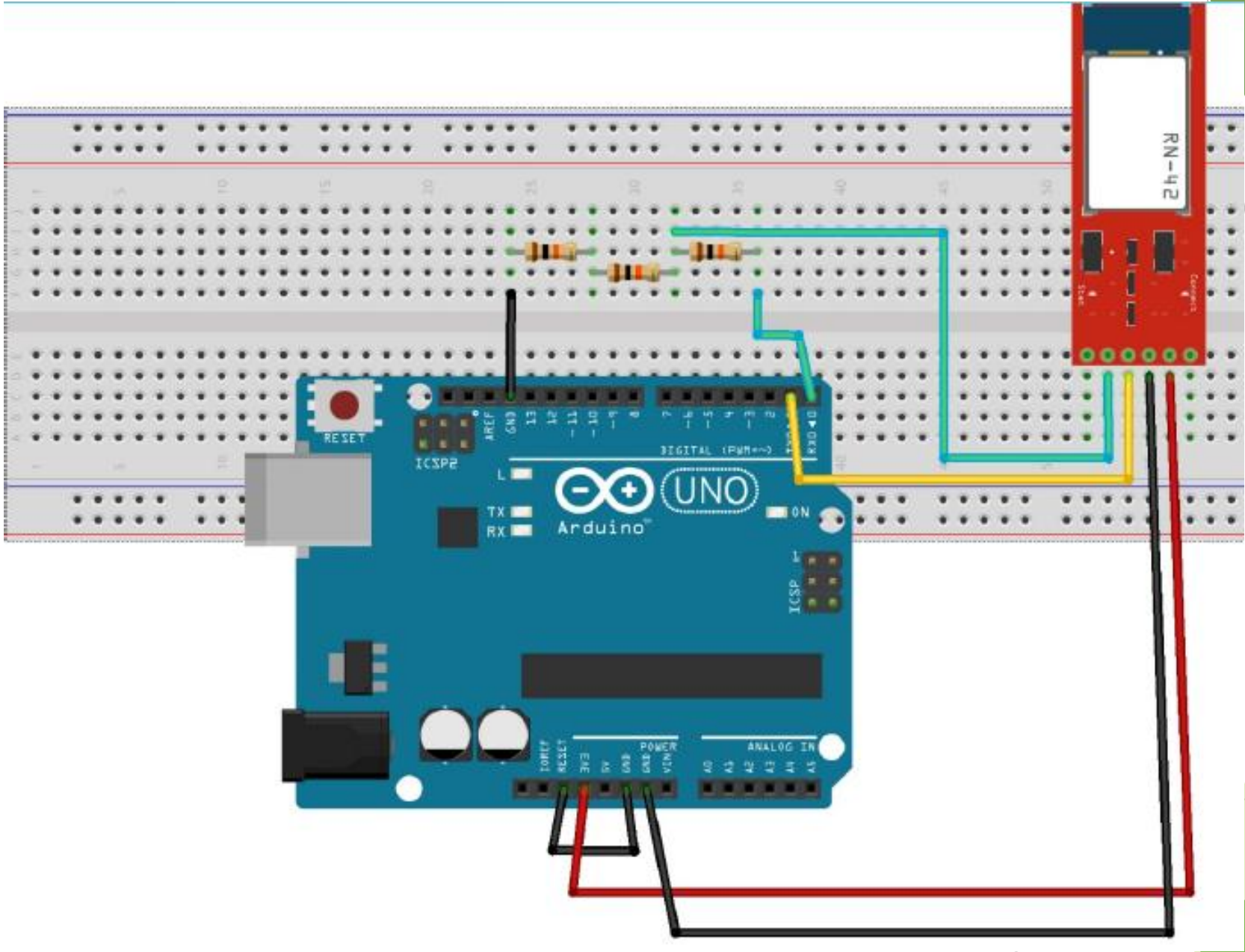
Voltaje de Operación de 3V a 3,6V

Consumo de 26  $\mu$ A en modo inactivo, 3 mA en conexión establecida y 30 mA hasta 50 mA transmitiendo.

Máxima distancia de transmisión: 20 m el RN-42 y 100 m el RN-41.

Puerto UART Local y USB.





Archivo Editar Programa Herramientas Ayuda

Auto Formato Ctrl+T

Archivo de programa.

Reparar codificación & Recargar.

Monitor Serie Ctrl+Mayúsculas+M

Board

Processor

Puerto Serie

Programador

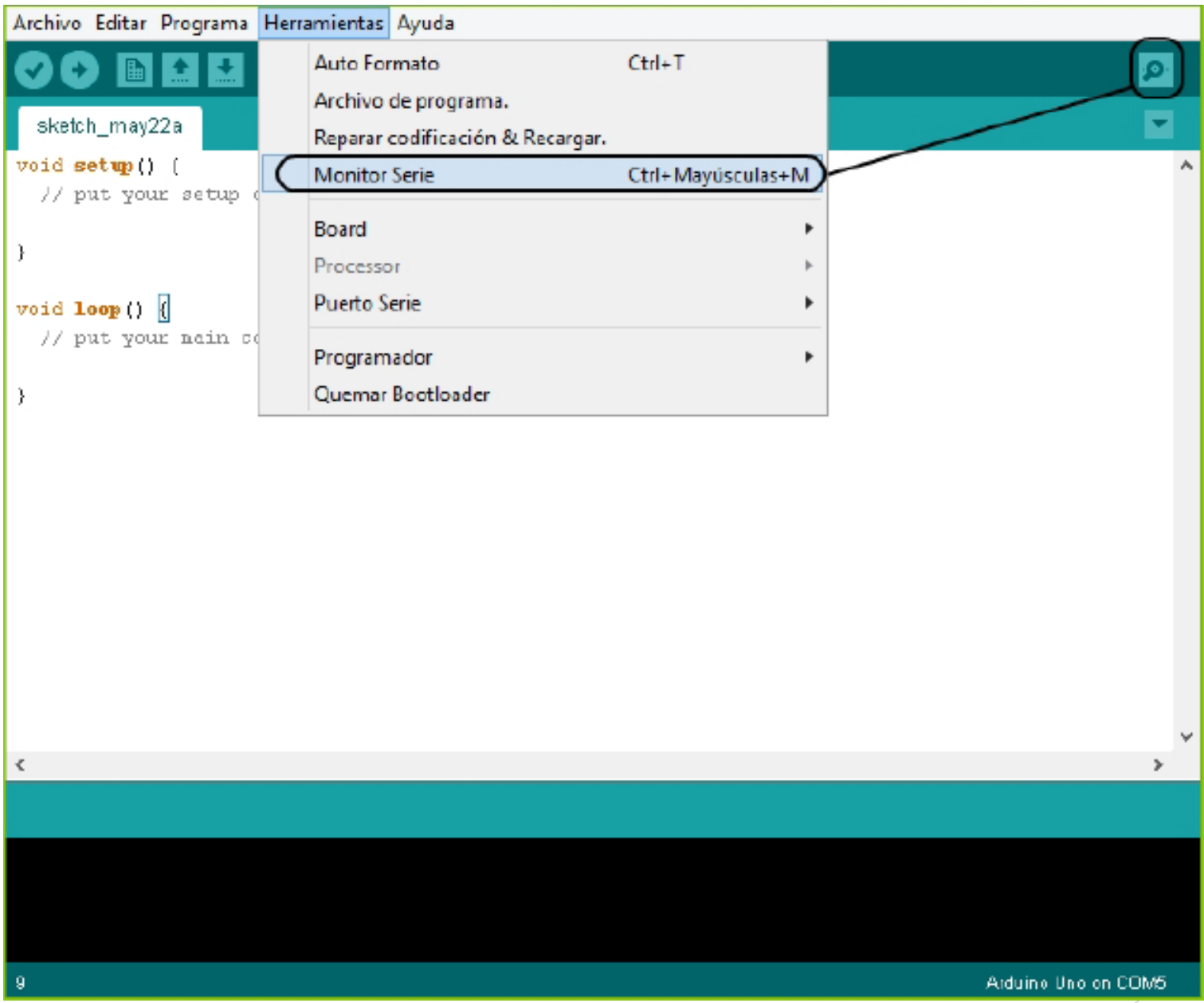
Quemar Bootloader

- Arduino AVR Boards
  - Arduino Uno
  - Arduino Duemilanove or Diecimila
  - Arduino Nano
  - Arduino Mega 2560 or Mega ADK
  - Arduino Mega (ATmega1280)
  - Arduino Leonardo
  - Arduino Micro
  - Arduino Esplora
  - Arduino Mini
  - Arduino Ethernet
  - Arduino Fio
  - Arduino BT
  - LilyPad Arduino USB
  - LilyPad Arduino
  - Arduino Pro or Pro Mini
  - Arduino NG or older
- Arduino ARM (32-bits) Boards
  - Arduino Due (Programming Port)
  - Arduino Due (Native USB Port)

```
sketch_may22a
void setup() {
  // put your setup code here, to configure pins
}

void loop() {
  // put your main code here, to run repeatedly
}
```

Arduino Uno on COM5



```
void setup() {  
  // put your setup code here  
}  
  
void loop() {  
  // put your main code here  
}
```

COM5

Enviar

depende

9600 baudio

Autoscroll

- Sin ajuste de línea
- Nueva línea
- Retorno de carro
- Ambos NL & CR

The screenshot shows the Arduino IDE interface. On the left, a code editor window titled 'sketch\_may22a' contains the following code:

```
void setup() {  
  // put your setup code here, to run once:   
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

In the center, a serial monitor window titled 'COM5' is open. The input field at the top contains '\$\$\$' and is highlighted with a black circle. Below the input field, the text 'Enviamos tres signos pesos para entrar a modo configuración del modulo' is displayed in red. At the bottom of the serial monitor, there is a checkbox for 'Autoscroll' which is checked, and two dropdown menus: 'Sin ajuste de línea' and '115200 baudio'. The status bar at the bottom right of the IDE indicates 'Arduino Uno on COM5'.



sketch\_may22a

```
void setup() {  
  // put your setup  
}  
  
void loop() {  
  // put your main c  
}
```

COM5

Enviar

CMD

**Si los datos  
fueron leídos el  
imprime CMD**

Autoscroll   Sin ajuste de línea   115200 baudio



Archivo Editar Programa Herramientas Ayuda

sketch\_may22a

```
void setup() {  
  // put your setup  
}  
  
void loop() {  
  // put your main c  
}
```

COM5

SU,96

Enviar

CMD

**“SU, ” es para  
cambiar la  
velocidad de  
transmisión(solo  
necesita los 2  
primeros dígitos)**

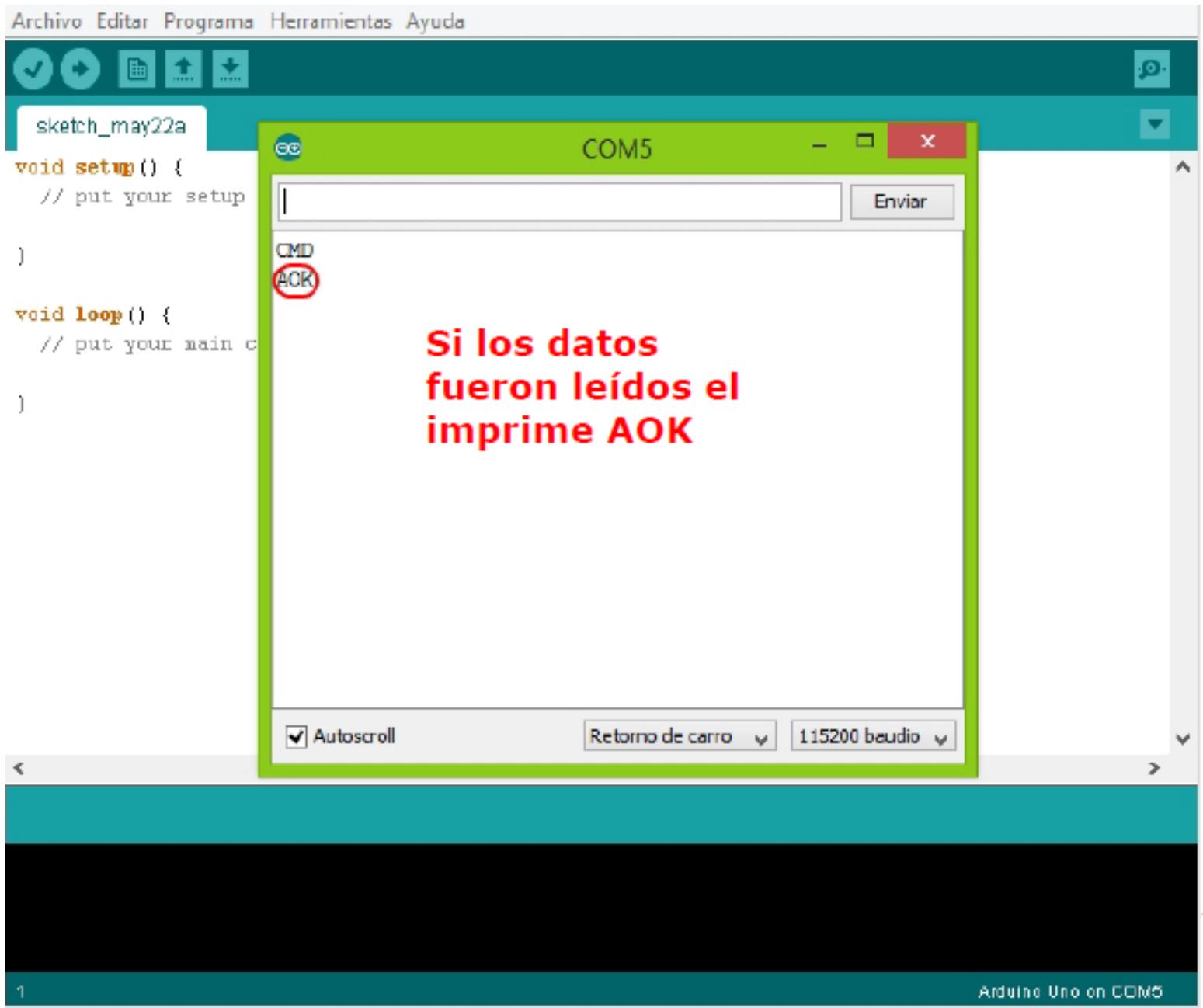
Autoscroll

Retorno de carro

115200 baudio

Se puede configurar en cualquiera de las siguientes velocidades:  
1200, 2400, 4800, 9600, 19.2,  
28.8, 38.4, 57.6, 115K, 230K,  
460K, 921K





sketch\_may22a

```
void setup() {  
  // put your setup  
}  
  
void loop() {  
  // put your main c  
}
```

COM5

Enviar

CMD  
ACK

**Si los datos  
fueron leídos el  
imprime AOK**

Autoscroll Retorno de carro 115200 baudio



sketch\_may22a

```
void setup() {  
  // put your setup  
}  
  
void loop() {  
  // put your main c  
}
```

COM5

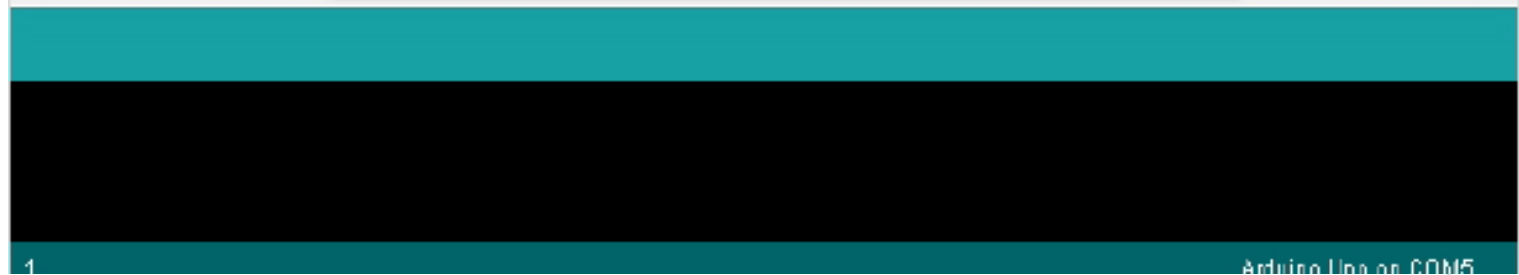
SN,JEPP

Enviar

CMD  
AOK

**"SN,"este comando es para cambiar el nombre (preferible cuatro letras).**

Autoscroll    Retorno de carro    115200 baudo





sketch\_may22a

```
void setup() {  
  // put your setup  
}  
  
void loop() {  
  // put your main c  
}
```

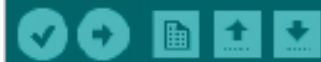
COM5

Enviar

CMD  
ACK  
ACK

**Si los datos  
fueron leídos el  
imprime AOK**

Autoscroll    Retorno de carro    115200 baudio



sketch\_may22a

```
void setup() {  
  // put your setup  
}  
  
void loop() {  
  // put your main c  
}
```

COM5

SP,1234

Enviar

CMD  
AOK  
AOK

**"SP," este comando es para cambiar la contraseña(preferible 4 dígitos)**

Autoscroll    Retorno de carro    115200 baudio

```
void setup() {  
  // put your setup code here, to run once:   
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

CMD  
ACK  
ACK  
ACK

**Si los datos fueron leídos el imprime AOK**

Autoscroll    Retorno de carro    115200 baudio

1    Arduino Uno on COM5



sketch\_may22a

```
void setup() {  
  // put your setup  
}  
  
void loop() {  
  // put your main c  
}
```

COM5

**R,1**

CMD  
ACK  
ACK  
ACK

**“R,1 ”este comando es para reiniciar el bluetooth y guardar la configuración.**

Autoscroll    Retorno de carro    115200 baudio

Archivo Editar Programa Herramientas Ayuda

✓ ↻ 🏠 ⬆️ ⬇️

sketch\_may22a

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

COM5

Enviar

CMD  
AOK  
AOK  
AOK  
Reboot!

**Si los datos fueron leídos el imprime Reboot!**

Nota: preferiblemente hacer todos los pasos anteriores en menos de 1 minuto.

Autoscroll    Retorno de carro ▼    115200 baudios ▼

1

Arduino Uno on COM5

# APP Inventor

## App Inventor

[ai2.appinventor.mit.edu/](https://ai2.appinventor.mit.edu/)

### MIT App Inventor

Instituto de Tecnología de Massachusetts

Desde este sitio se puede acceder Inventor MIT App, que le permite desarrollar aplicaciones para dispositivos **Android** que utilizan un navegador web y ya sea un teléfono o un emulador conectado. App Inventor fue desarrollado originalmente por Google. El sitio también incluye documentación de un contenido educativo, y esto se está licenciado para usted bajo la licencia Creative Commons Reconocimiento 3.0 Unported ( [CC BY 3.0](https://creativecommons.org/licenses/by/3.0/) ).



## Requisitos:

**1) Tener correo en gmail**

<https://mail.google.com/mail/u/0/?pli=1#inbox>

**2) Entrar a el siguiente link**

<http://appinventor.mit.edu/explore/>

**Create!**



MIT App Inventor 2  
Beta

Proyectos ▾

Conecte ▾

Construir ▾

Ayuda ▾

Iniciar nuevo proyecto

Eliminar Proyecto

### Mis proyectos

Nombre

Fecha de creación

### Crear nuevo proyecto App Inventor

Nombre del  
proyecto:

Cancelar

OKAY

**Paleta**

**Interfaz de usuario**

- Botón
- CheckBox
- DatePicker
- Imagen
- Etiqueta
- ListPicker
- ListView
- Notificador
- PasswordTextBox
- Deslizador
- Hiladero
- TextBox
- TimePicker
- WebView

Screen1

**Propiedades**

Button1

BackgroundColor  
■ Defecto

Activado

FontBold

FontItalic

Tamaño de Letra  
14.0

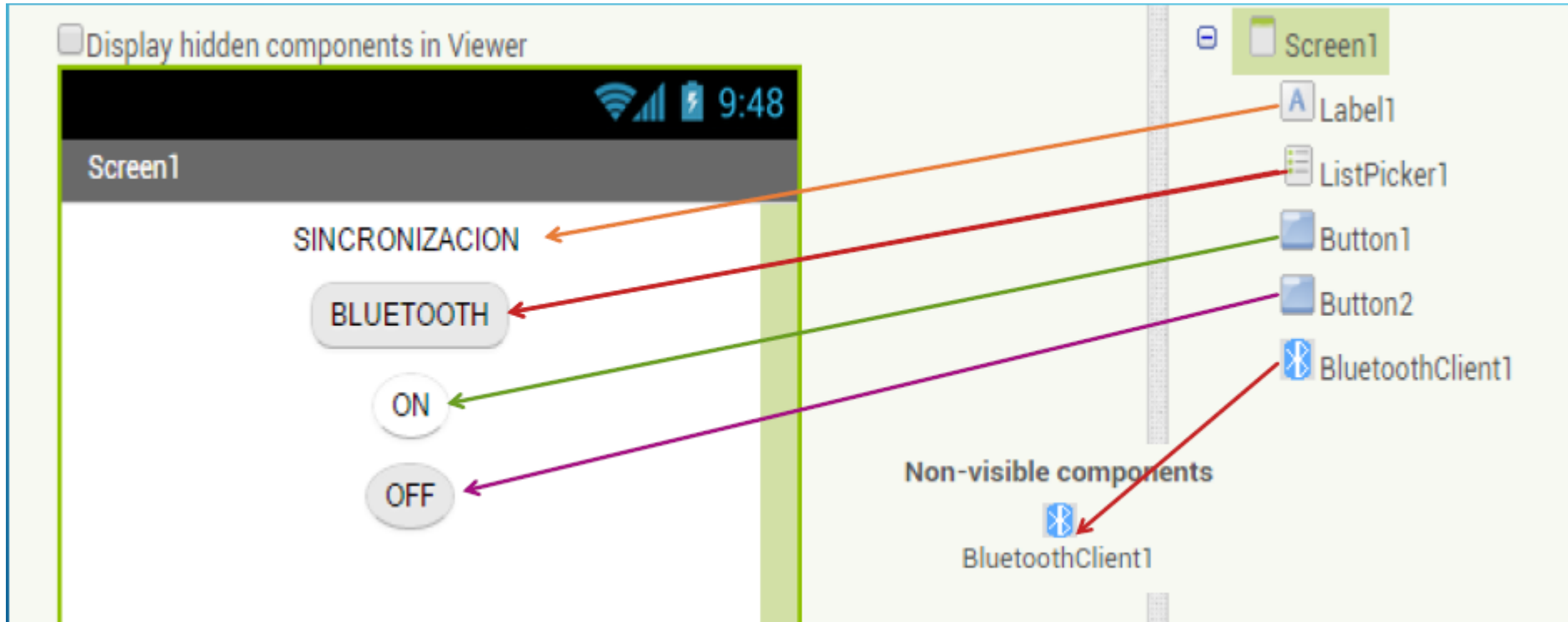
FontTypeface  
defecto

Imagen  
None...

Forma  
defecto

ShowFeedback

Texto



when ListPicker1 .BeforePicking

do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking

do set ListPicker1 . Selection to call BluetoothClient1 . Connect  
address ListPicker1 . Selection

set Label1 . Text to " CONECTADO "

when Button1 .Click

do call BluetoothClient1 .SendText  
text " E "

when Button2 .Click

do call BluetoothClient1 .SendText  
text " A "

# CREA EL EJECUTABLE .APK

The image shows two side-by-side screenshots of the Android Studio interface. The left screenshot shows the 'Build' menu open, with the option 'App ( save .apk to my computer )' highlighted. A red arrow points from this option to the title 'CREA EL EJECUTABLE .APK'. The right screenshot shows the 'Build' menu open, with the option 'Export selected project (.aia) to my computer' highlighted. The background of both screenshots shows a mobile app preview with a status bar at 9:48 and a screen titled 'Screen1' containing a 'SINCRONIZACION' section with a 'BLUETOOTH' button and 'ON'/'OFF' toggle options.

Projects ▾ Connect ▾ Build ▾ Help ▾

screen1 ▾ Add Screen ...

App ( provide QR code for .apk )  
App ( save .apk to my computer )

Viewer

Display hidden components in Viewer

Screen1

SINCRONIZACION

BLUETOOTH

ON

OFF

Projects ▾ Connect ▾ Build ▾ Help ▾

My projects

Start new project

Import project (.aia) from my computer ...

Import project (.aia) from a repository ...

Delete Project

Save project

Save project as ...

Checkpoint

Export selected project (.aia) to my computer

Export all projects

Import keystore

Export keystore

Delete keystore

9:48

SINCRONIZACION

BLUETOOTH

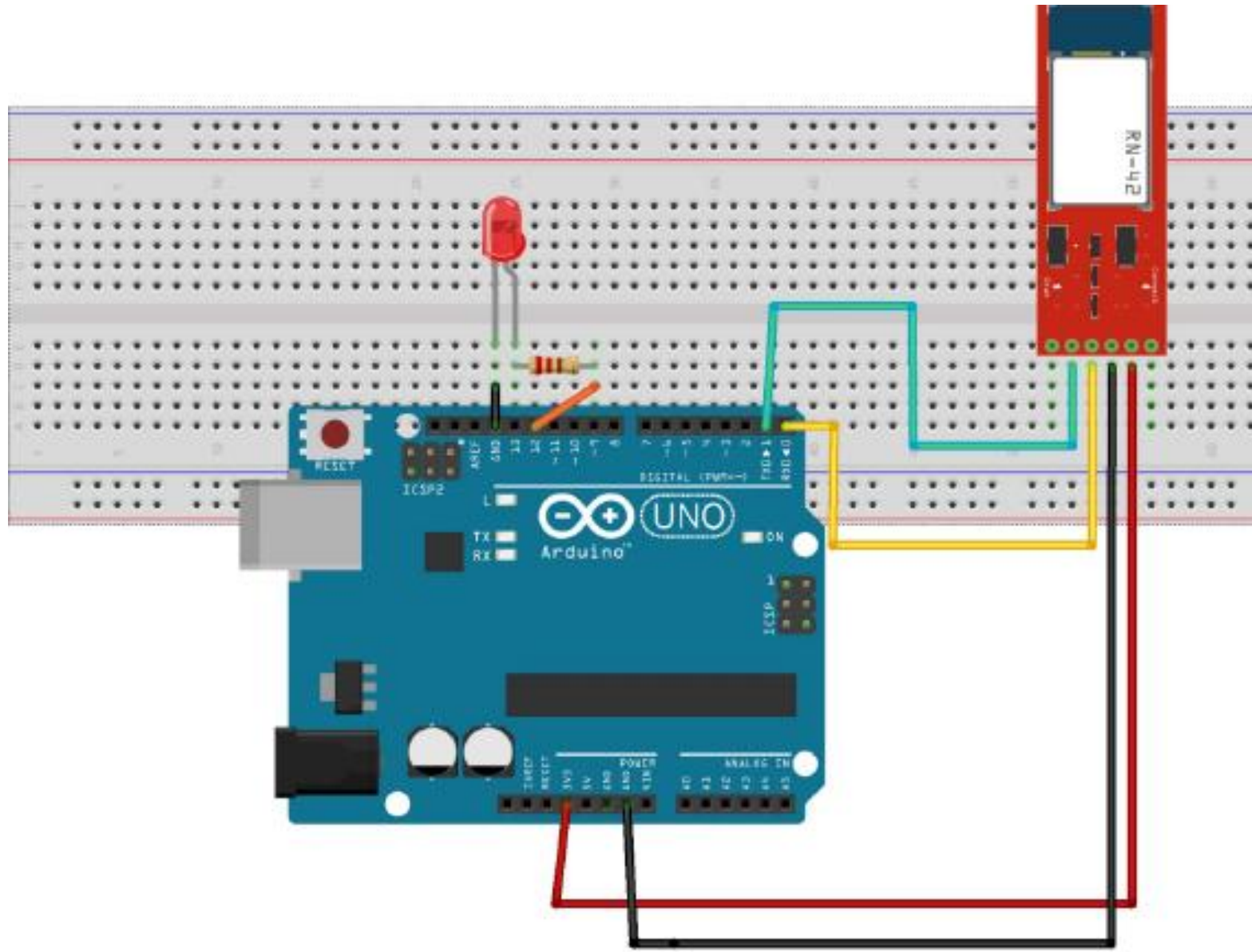
ON

OFF

```
int LED12 = 12; // LED CONECTADO AL PIN 12 DEL ARDUINO

void setup()
{
  Serial.begin(9600);
  pinMode(LED12, OUTPUT);
}
void loop()
{
  char i;
  if (Serial.available())
  {
    i = Serial.read(); //leer dato y lo guarada en i
    switch (i)
    {
      case 'E':
        digitalWrite(LED12 , HIGH);
        break;

      case 'A':
        digitalWrite(LED12 , LOW);
        break;
    }
  }
}
```





Instructables (<http://www.instructables.com>)  
Makezine (<http://blog.makezine.com/arduino>)  
MakeProjects (<http://makeprojects.com/Topic/Arduino>)  
HackADay (<http://hackaday.com/category/arduino-hacks>)  
HackNMod (<http://hacknmod.com>)  
Dangerous Prototypes (<http://dangerousprototypes.com>)  
Electronics Lab (<http://www.electronics-lab.com/blog>)  
BricoGeek (<http://www.bricogeek.com>)  
Embedds (<http://www.embedds.com>)  
Sección de proyectos Jameco (<http://www.jameco.com/Jameco/PressRoom/diy.html>)  
Sección de proyectos Arduino (<http://arduino.cc/playground/Projects/ArduinoUsers>)  
Proyectos del HLT (<http://hlt.media.mit.edu/?cat=20> , <http://hlt.media.mit.edu/?p=1283> y [?p=1314](http://hlt.media.mit.edu/?p=1314))

## Bluetooth y Arduino

<https://www.youtube.com/watch?v=Is6pJyEv8s0#t=21>

<http://www.youtube.com/watch?v=fokloKBNCOE#t=17>