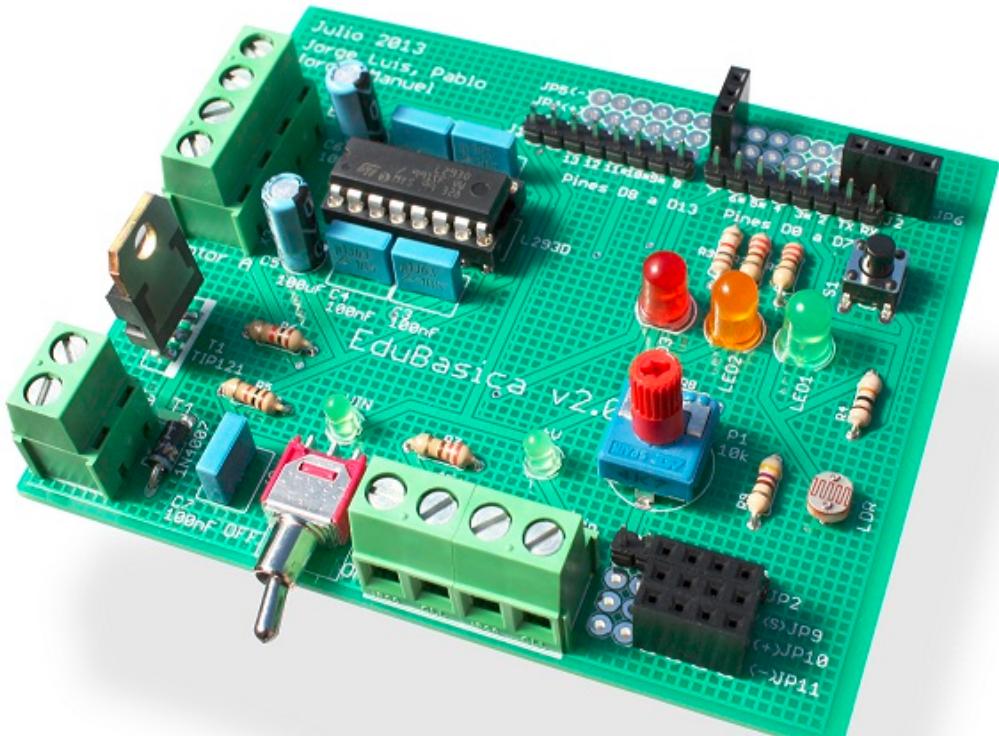


Recurso Tecnológico

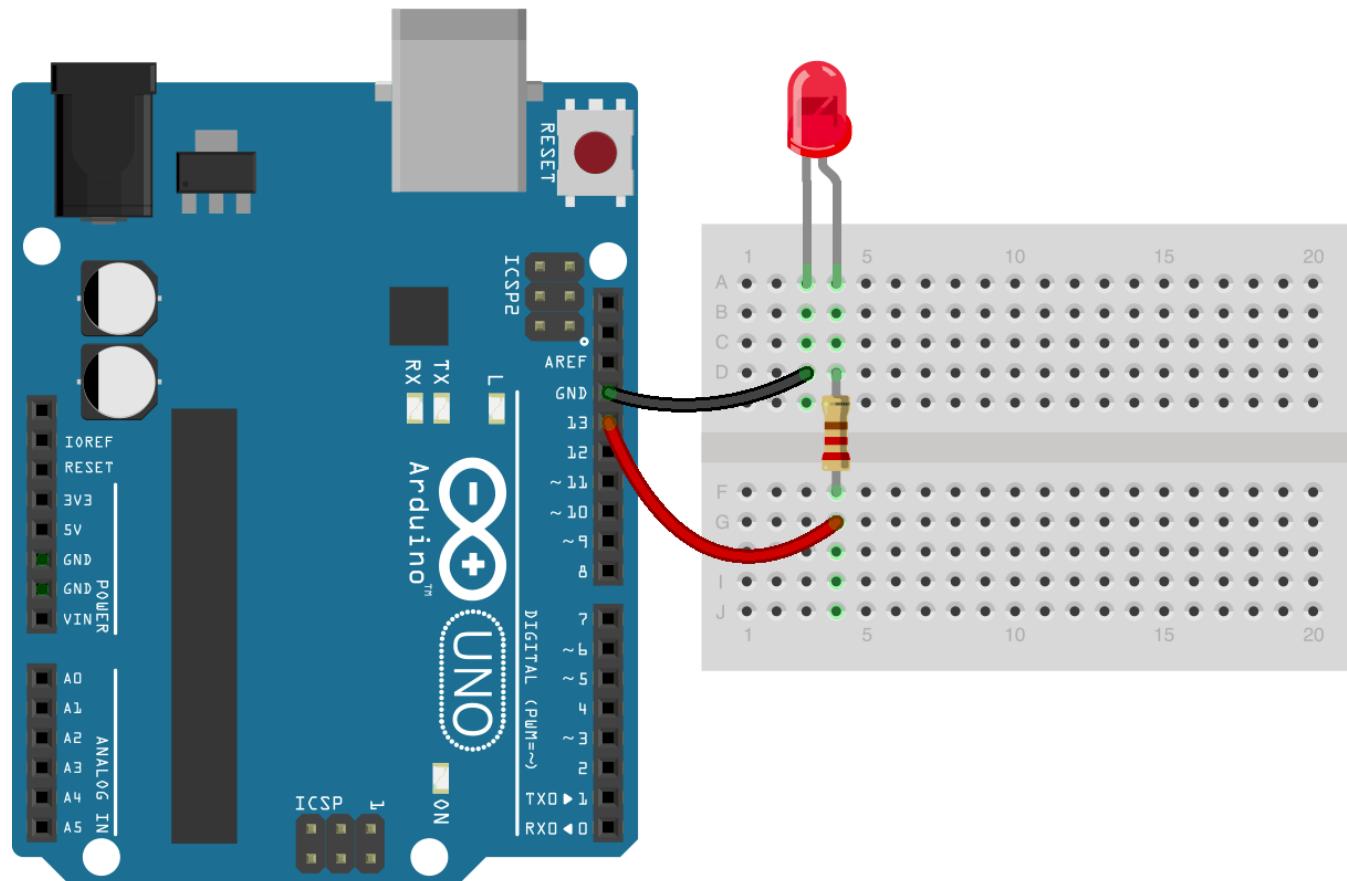


RED ANDALUZA
DE ROBÓTICA
Y TECNOLOGÍA
EDUCATIVA



COITTA/AAGIT
Colegio Oficial / Asociación Andaluza
de Graduados & Ingenieros
Técnicos de Telecomunicación

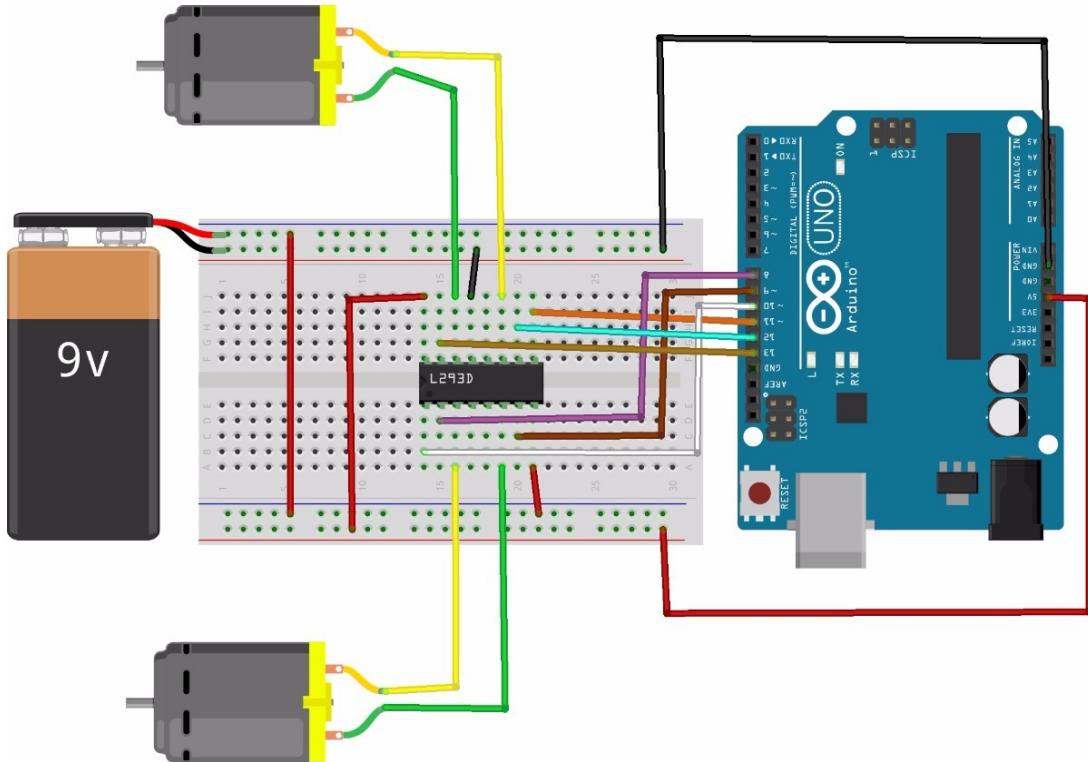
Recurso Tecnológico



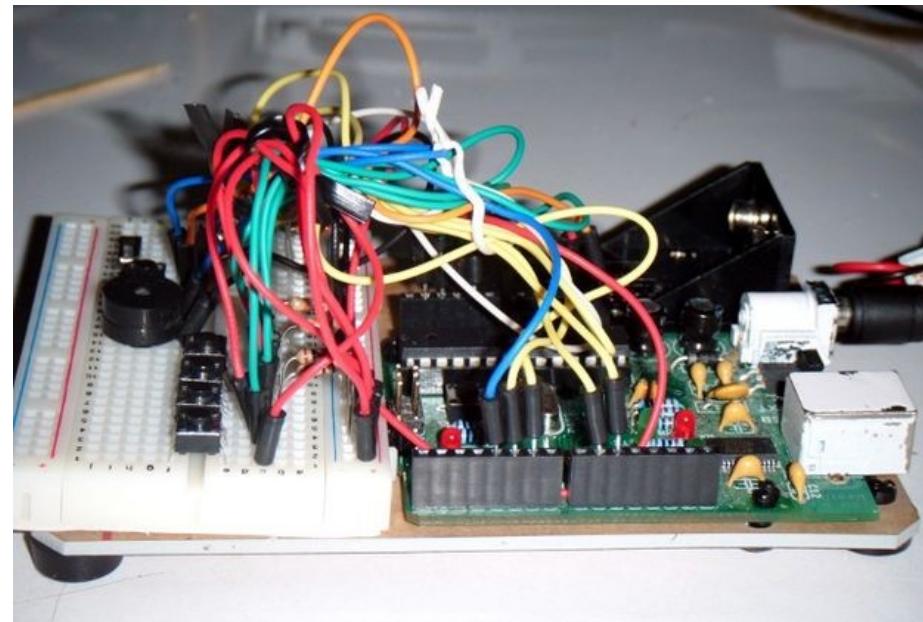
fritzing



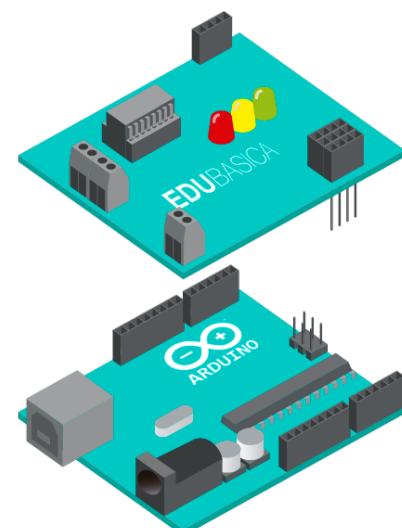
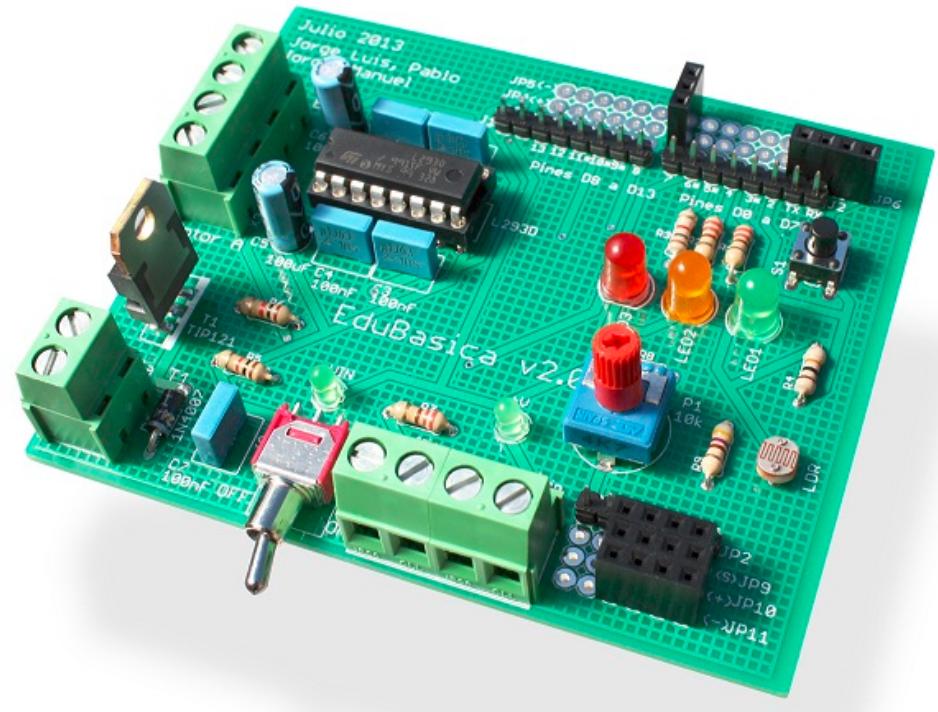
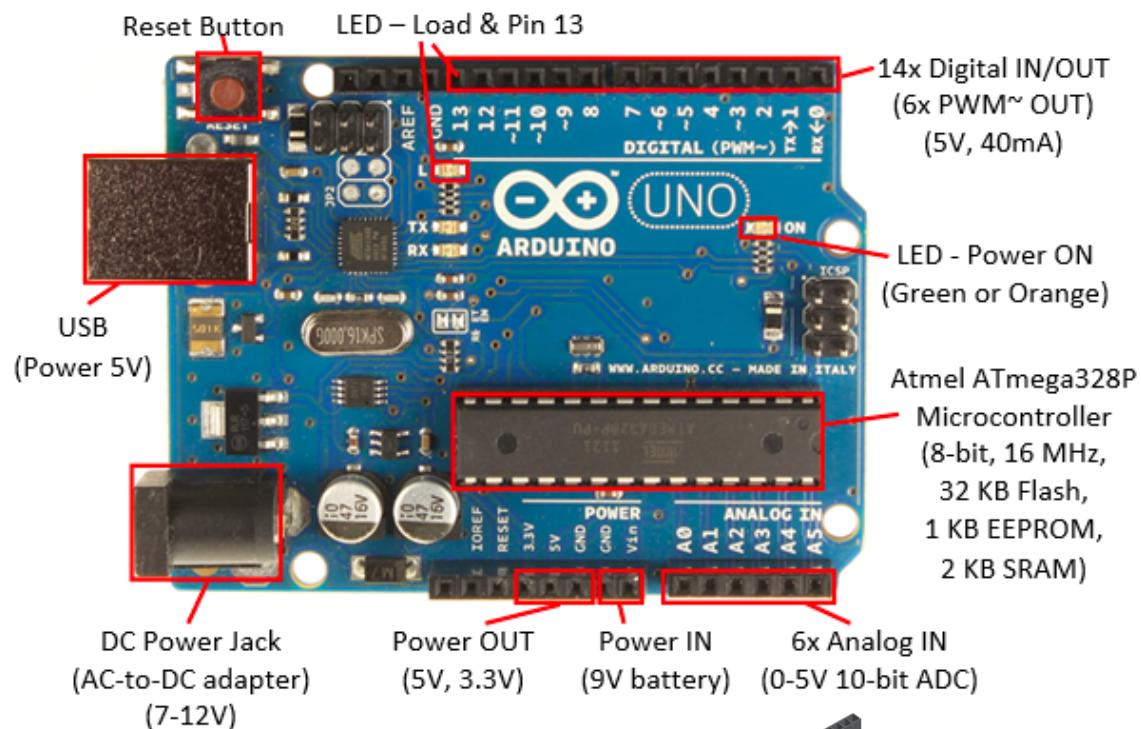
Recurso Tecnológico



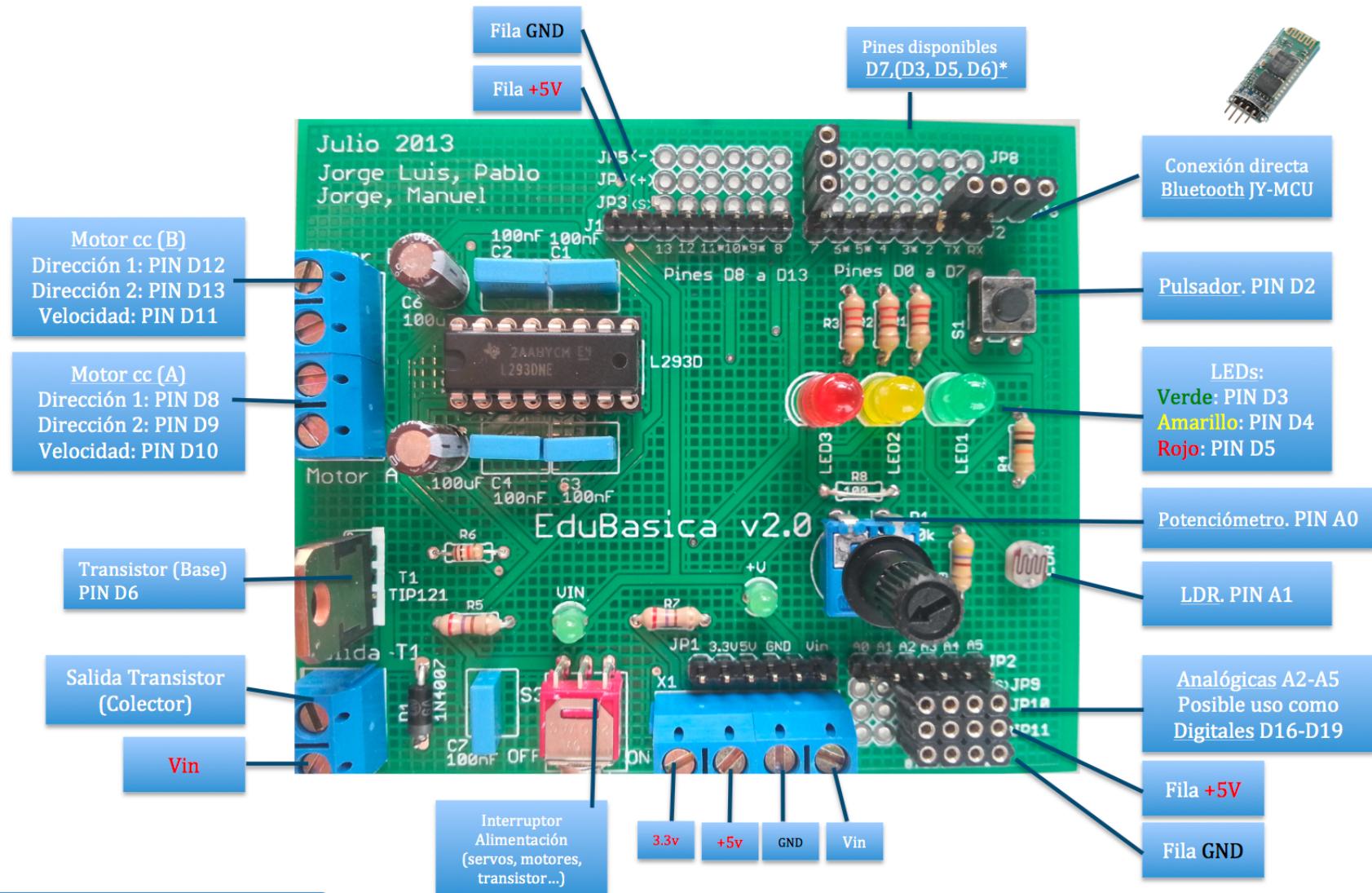
fritzing



Recurso Tecnológico



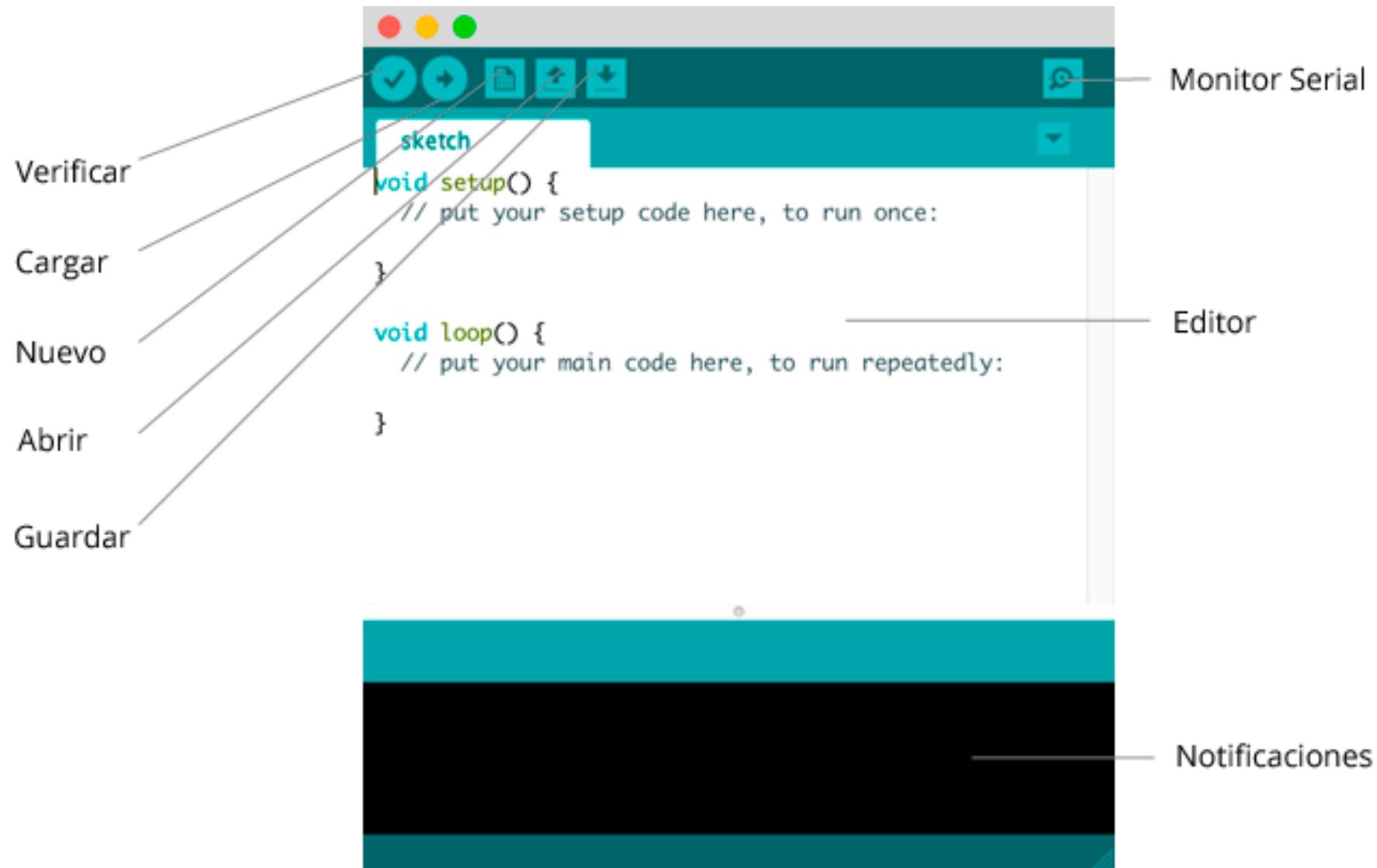
EduBasica v2.0 (rev. Nov/2013)



*Usar esos pines anulan el transistor, y los LEDs



Entorno de programación (IDE)





Estructura de programa

Knob

```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>
```

Descripción del programa

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin
```

Módulos y declaración de variables

```
void setup()
{
    myservo.attach(9); // attaches the servo on pin 9 to the servo object
}
```

Bloque de configuración

```
void loop()
{
    val = analogRead(potpin);
    val = map(val, 0, 1023, 0, 179);
    myservo.write(val);
    delay(15);
}
```

Bloque de ejecución continua

```
// reads the value of the potentiometer (value between 0 and 1023)
// scale it to use it with the servo (value between 0 and 180)
// sets the servo position according to the scaled value
// waits for the servo to get there
```



Entorno de programación visual: Visualino (<http://visualino.net/>)

The screenshot shows the Visualino environment. On the left, a sidebar lists categories: Funciones, Control, Lógica, Matemáticas, Variables, Texto, Comunicación, Zum bloqs, Octopus bloqs, Funciones PIN, LCD bloqs, and Servo. The main workspace displays a Scratch-style script for an 'Inicio' (Start) block:

- Repetir [Var valorLDR v = Sensor de Luz PIN# Pin analógico A1 v] [Imprimir por puerto serie "Valor leido del sensor LDR: "][Imprimir por puerto serie con salto de línea Var valorLDR v]
- Esperar [ms] 2000

To the right, the Arduino IDE shows the generated C++ code:

```
/** Global variables */
int valorLDR=0;

/** Function declaration */
void setup()
{

    Serial.begin(9600);
}

void loop()
{
    valorLDR=analogRead(A1);
    Serial.print("Valor leido del sensor LDR: ");
    Serial.println(valorLDR);
    delay(2000);
}

/** Function definition */
```

At the bottom, there are tabs for 'arduino:avr:uno' and '/dev/ttyACM0', and a file path: /media/manual/TOSHIBA EXT/00DellFebrero2018/00_carpetasActivas/EduBasicaManuel/EduBasicaTest_GranadaMarzo2017/progTaller2017diciembre/visualino/prog06_sensorLDR.bly



Entorno de programación visual: mBlock (<https://www.makeblock.com>)
<https://www.makeblock.es>)

mBlock(v4.0.4)

Archivo ▾ Editar ▾ Extensiones ▾ Ayuda ▾

Programas

- Movimiento
- Apariencia
- Sonido
- Lápiz
- Datos y Bloques
- Eventos
- Control
- Sensores
- Operadores
- Robots**

Arduino ▾

Programa de Arduino

- leer pin digital 9
- leer pin analógico (A) 0
- lee el pulso del pin 13 expirado 20000
- fijar salida pin digital 9 a ALTO
- set analog(PWM) pin 5 output as 0
- reproducir tono 9 en nota C4 beat Med
- fijar ángulo del pin 9 del servo a 90
- escribir en el serial el texto hola
- bytes disponibles en el serial
- byte leido en el serial
- lee el sensor ultrasónico trig pin 13 echo
- cronómetro
- reiniciar cronómetro
- map 50 from 1 ~ 100 to 1 ~ 1000
- constrain 110 within 0 ~ 100

EduBasica
por siempre

```

    si leer pin digital 7 entonces
        fijar salida pin digital 3 a ALTO
    si no
        fijar salida pin digital 3 a BAJO
    esperar 0.1 segundos
  
```

x: 13 y: -16

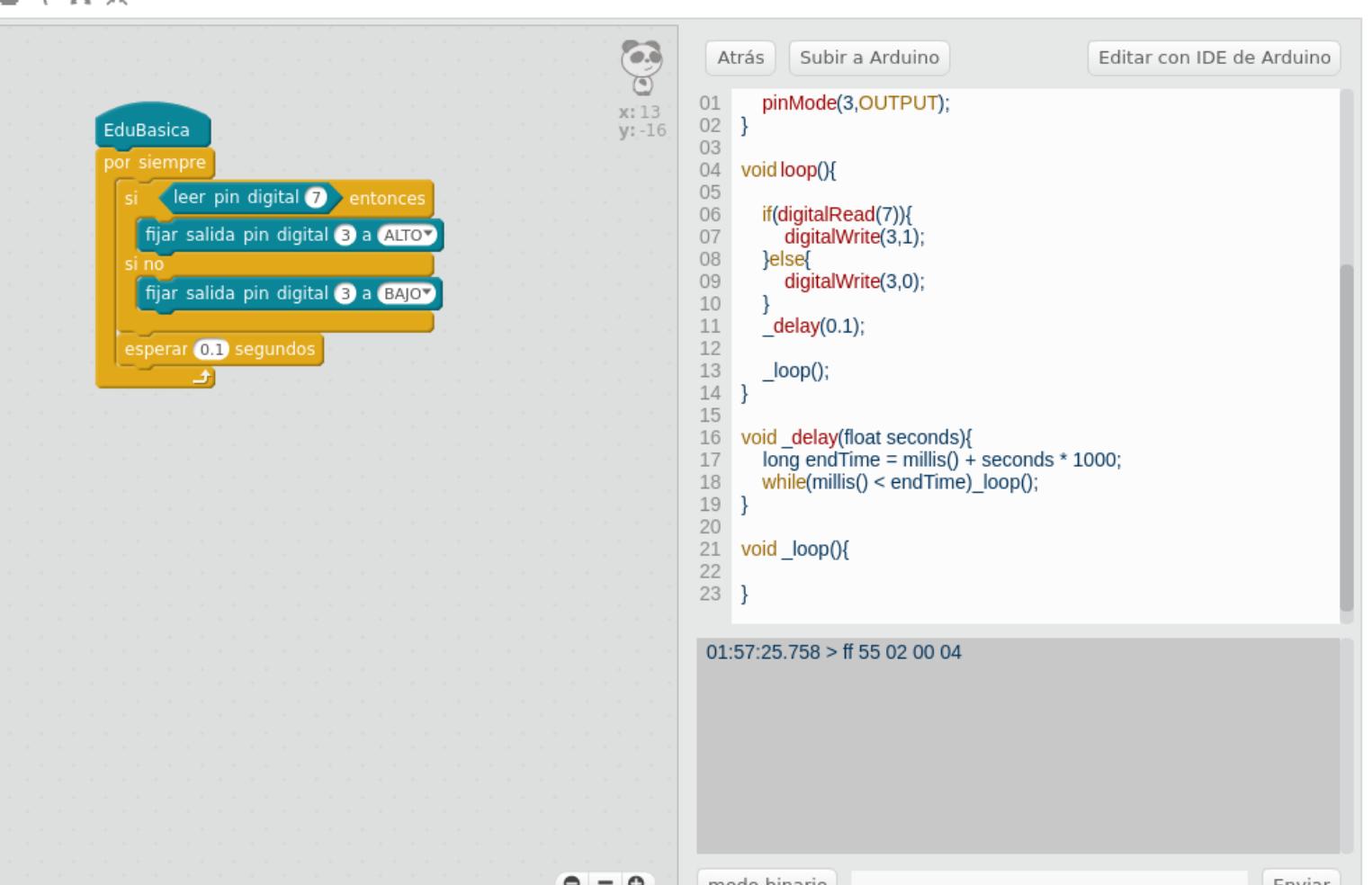
Atrás Subir a Arduino Editar con IDE de Arduino

```

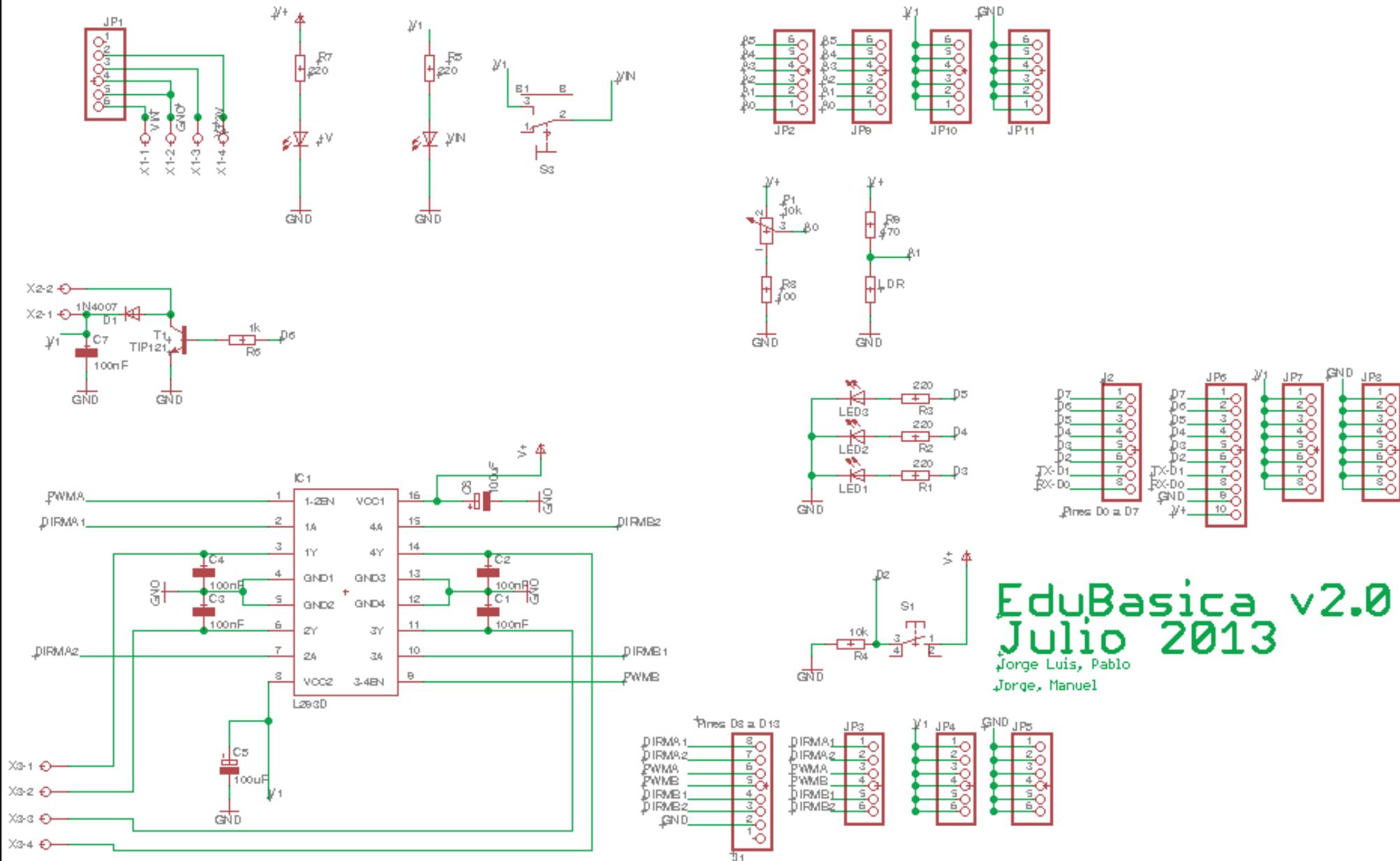
01 pinMode(3,OUTPUT);
02 }
03
04 void loop(){
05
06   if(digitalRead(7)){
07     digitalWrite(3,1);
08   }else{
09     digitalWrite(3,0);
10   }
11   _delay(0.1);
12
13   _loop();
14 }
15
16 void _delay(float seconds){
17   long endTime = millis() + seconds * 1000;
18   while(millis() < endTime)_loop();
19 }
20
21 void _loop(){
22 }
23
  
```

01:57:25.758 > ff 55 02 00 04

modo binario Enviar

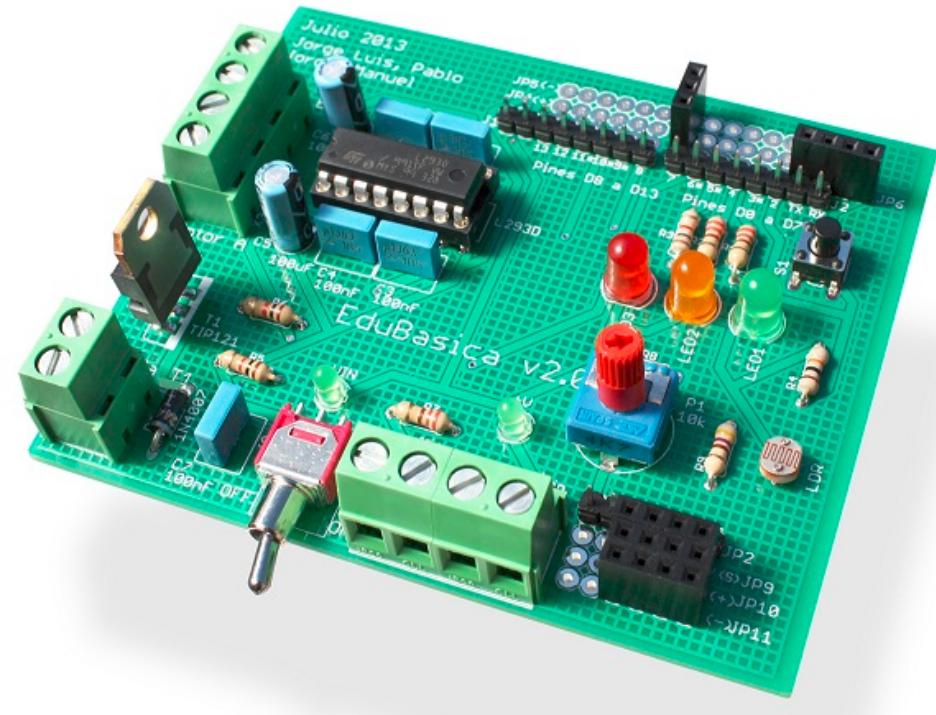
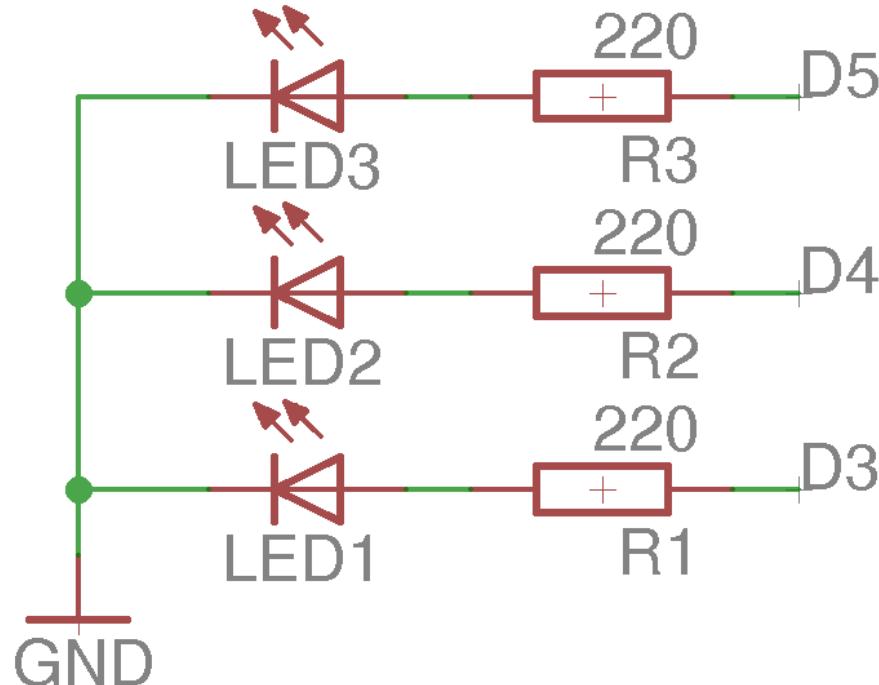


Esquemático de EduBasica





Salidas Digitales



Programas de test:

- test00_blink.ino
- test01_leds.ino

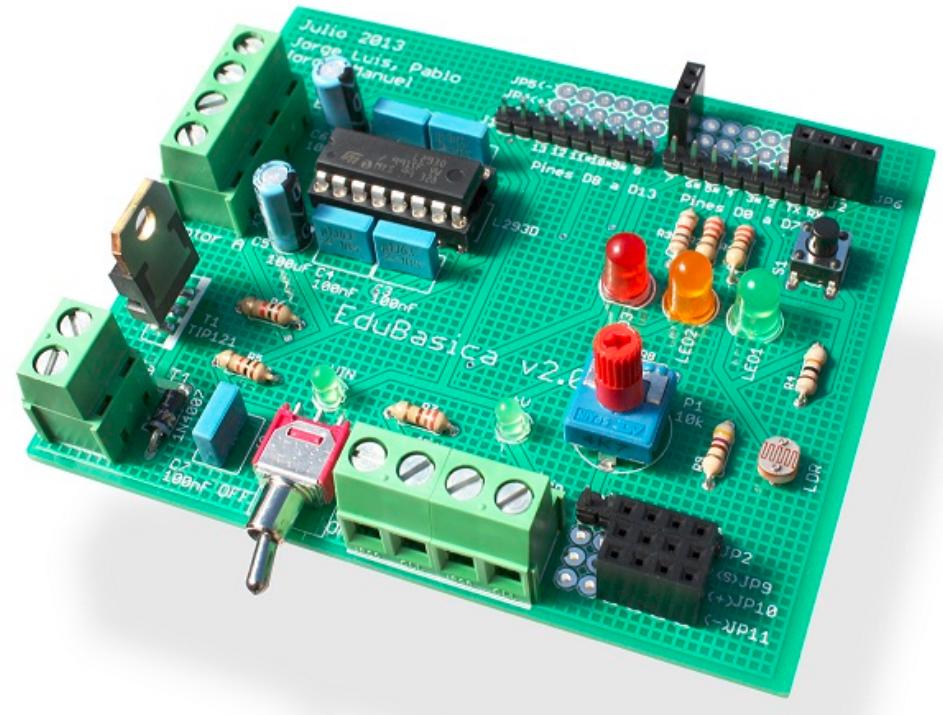
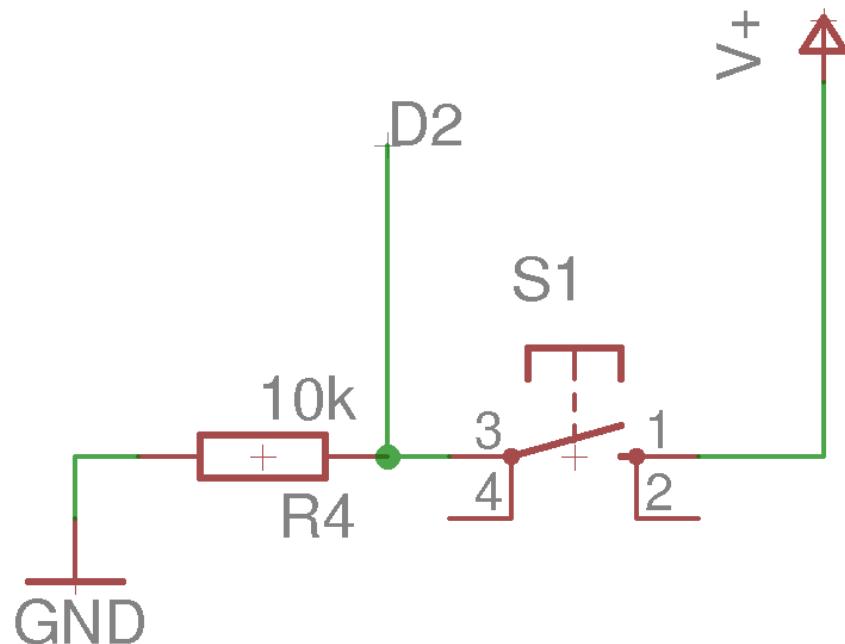
Programa Visualino

- prog01_Parpadeo.bly
- prog02_Secuencia.bly

Actividades:

- Programar secuencias con tiempos diferentes.

Entrada Digital



Programa de test:

- test02_pulsador.ino

Programa Visualino:

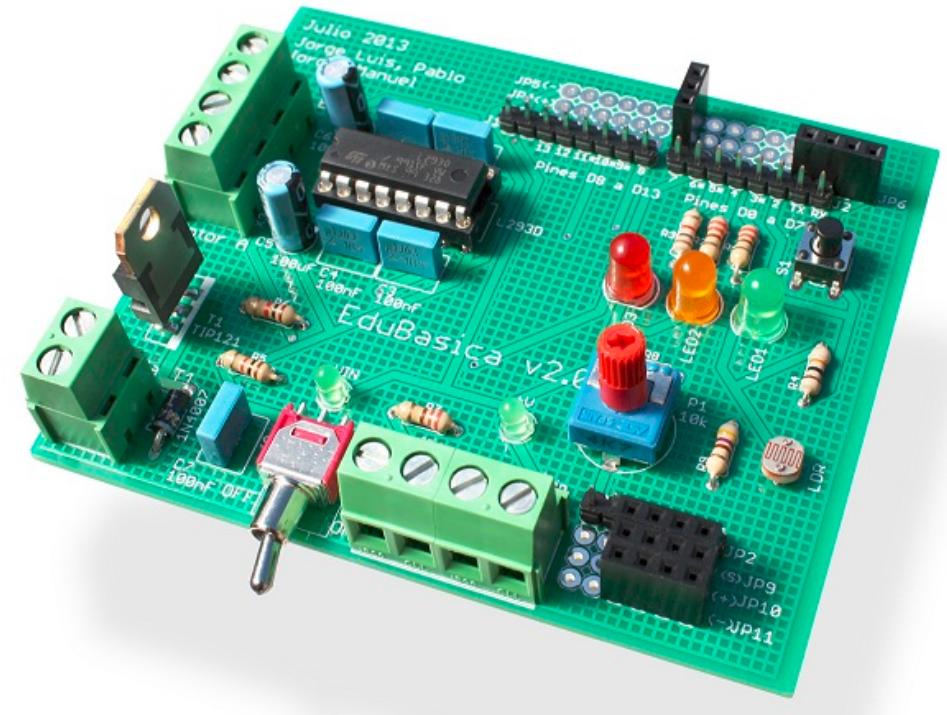
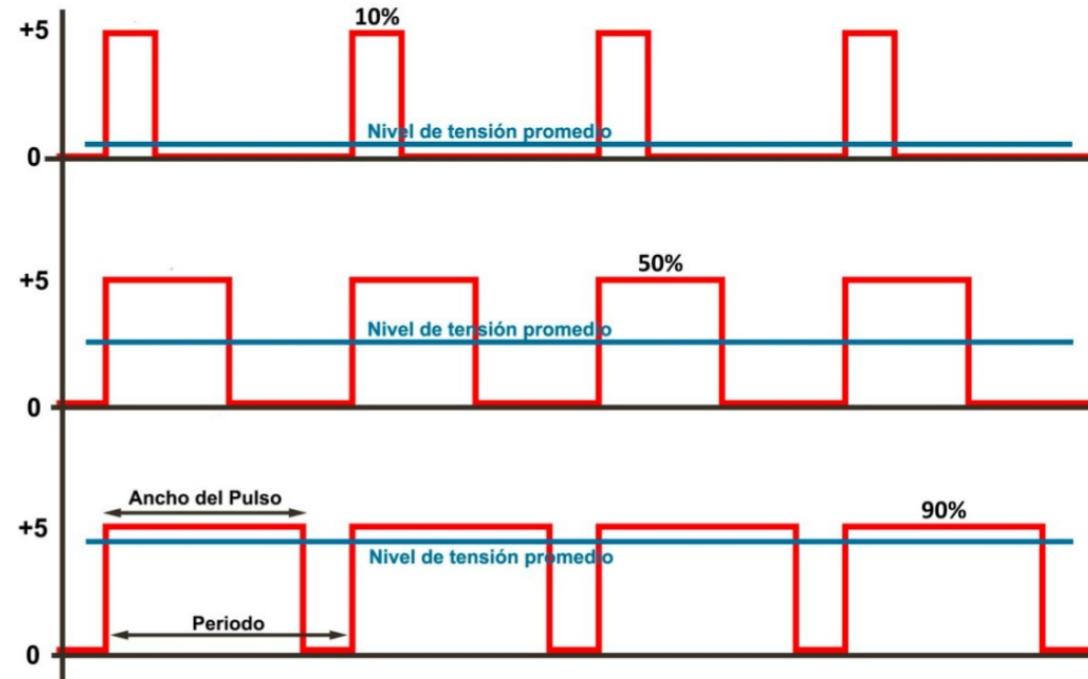
prog03_Pulsador.bly

Actividades:

- Programar secuencias cambiantes al pulsar el pulsador.
- Semáforo activado con el pulsador.



Salidas Analógicas

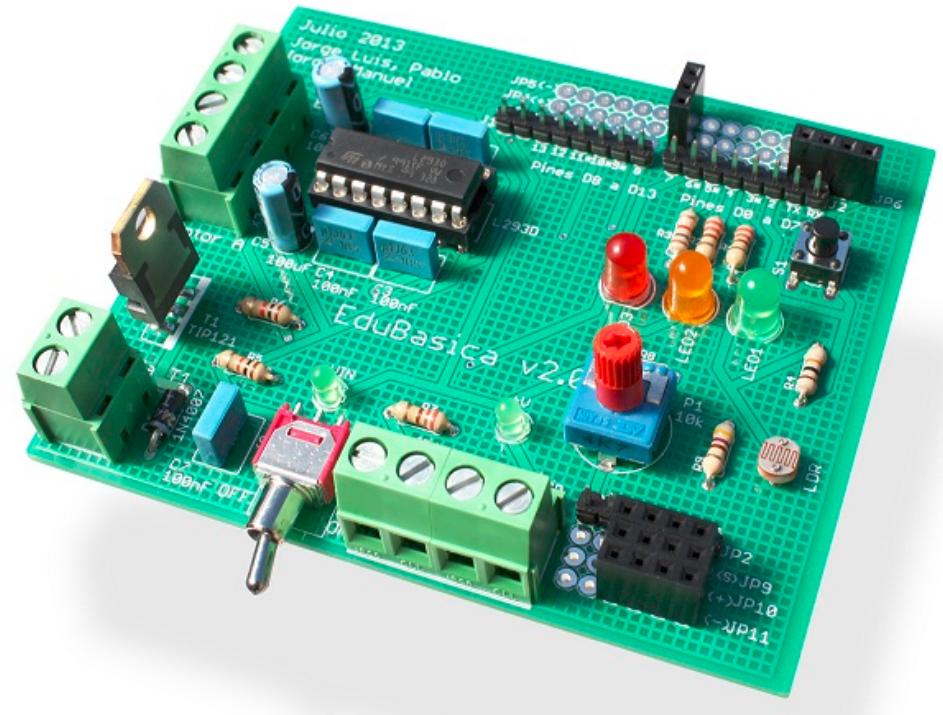
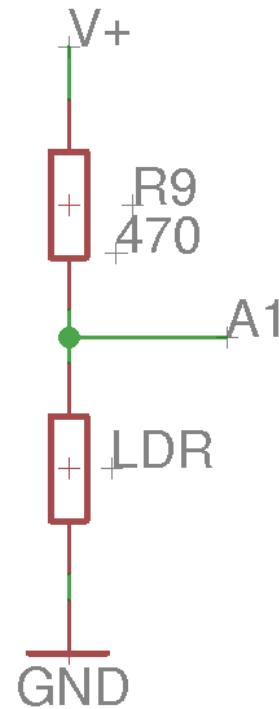
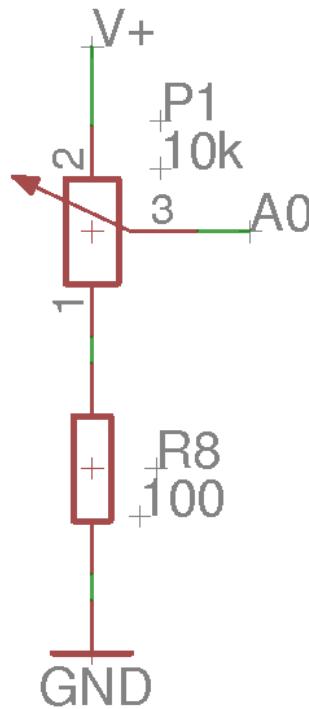


La aplicación de la señal PWM (señal ~ 500Hz) se ve con las entradas analógicas.

prog04_pwm.bly



Entradas Analógicas



Programas de test:

- test03_potenciometro.ino
- test04_LDR.ino

Programas Visualino:

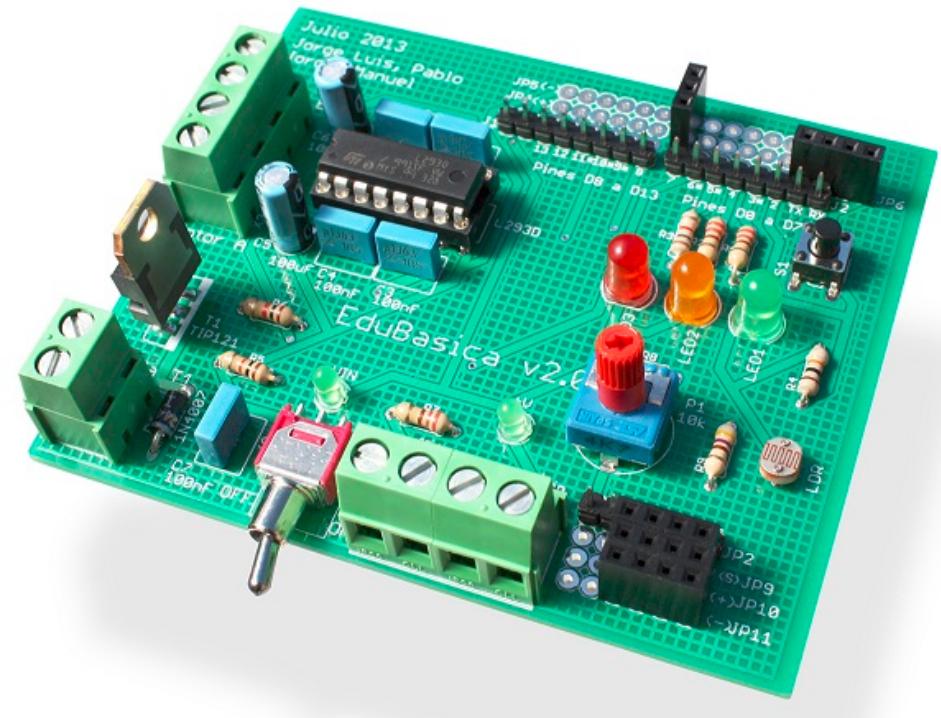
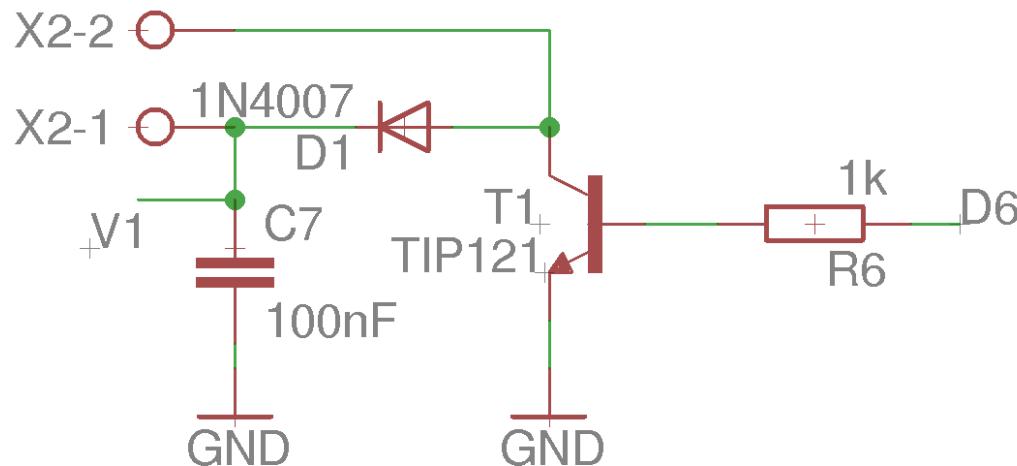
- prog05_potenciometro.bly
- prog06_sensorLDR.bly

Actividades:

- Programar el encendido de un led con el LDR (día-noche).



Salida de potencia



Programas de test:

- test05_Transistor.ino

Programa Visualino

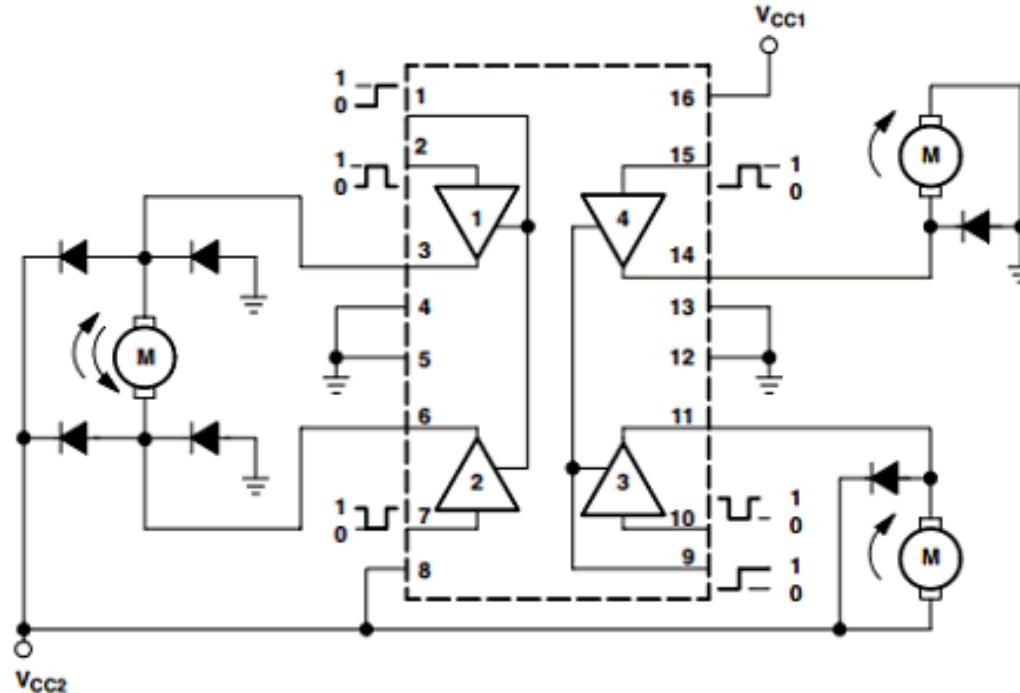
- prog07_salidaT1.bly

Actividades:

- Programar el control de giro del motor dc con el potenciómetro.



L293. Integrado de control de motores



NOTE: Output diodes are internal in L293D.

FUNCTION TABLE
(each driver)

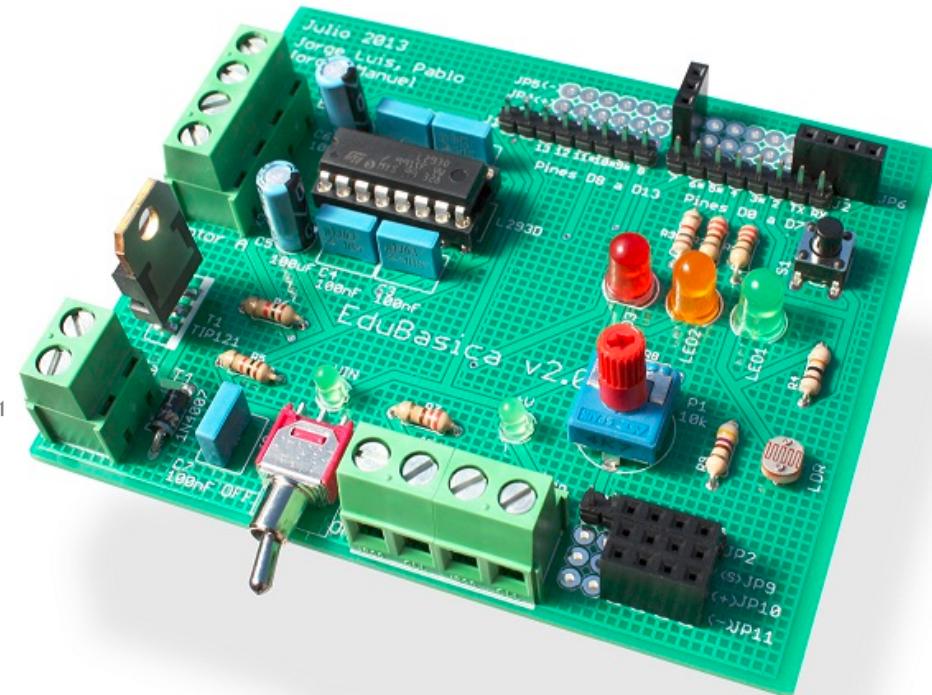
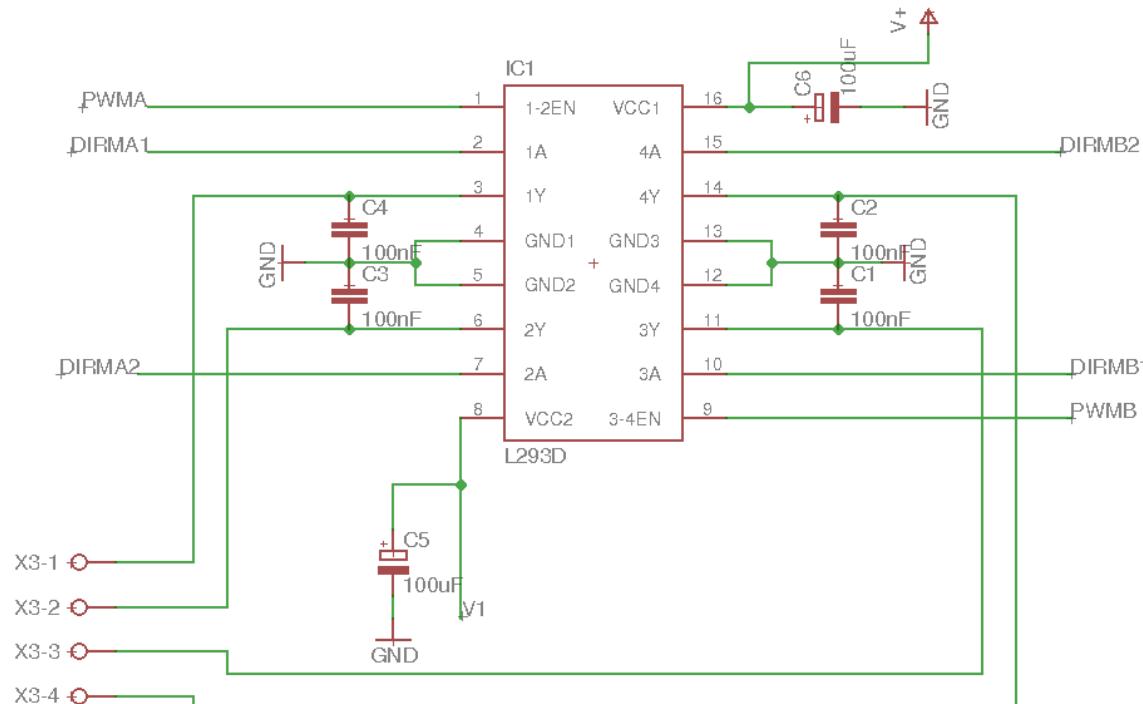
INPUTS [†]		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant,
Z = high impedance (off)

[†] In the thermal shutdown mode, the output is
in the high-impedance state, regardless of
the input levels.



L293. Integrado de control de motores



Programas de test:

- test06_MotorA.ino
- test07_MotorB.ino

Programas Visualino:

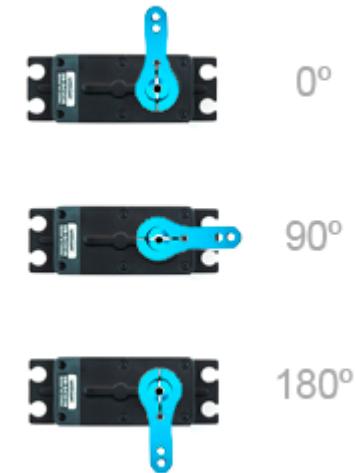
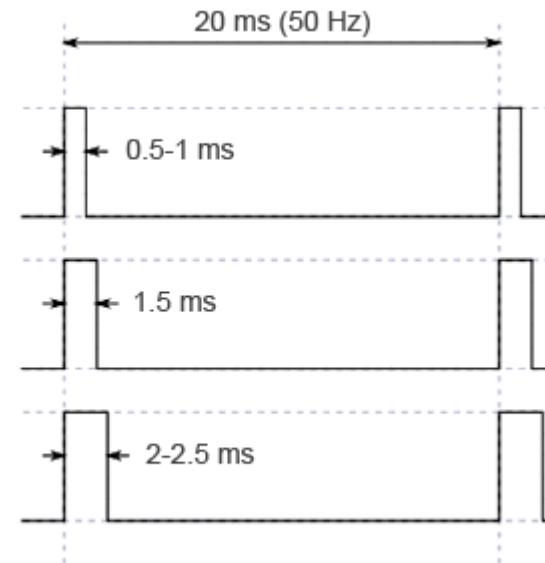
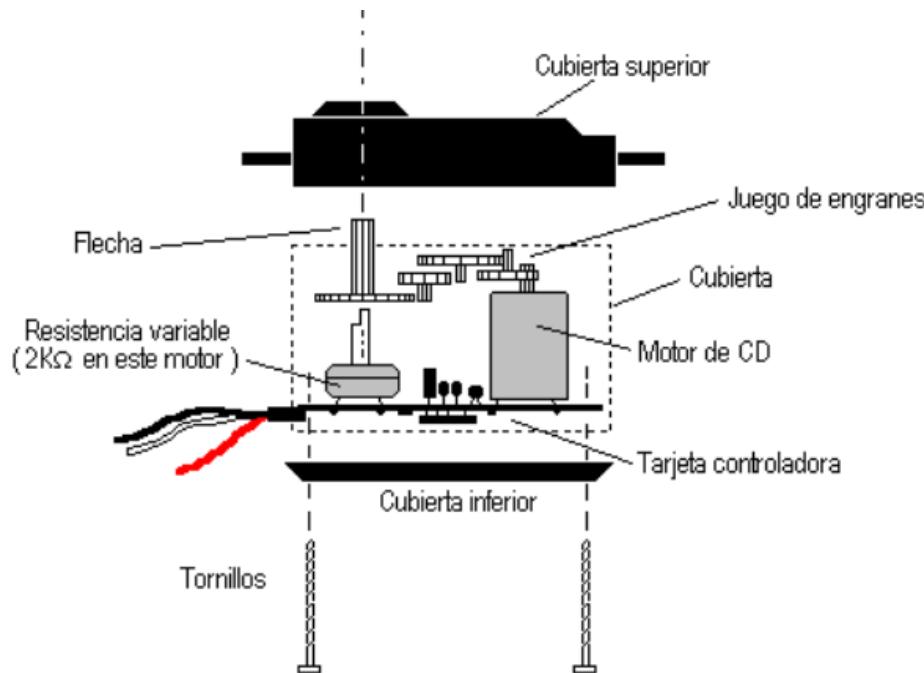
- prog08_motorA01.bly
- prog08_motorA02.bly
- prog08_motorA03.bly
- prog08_motorA4.bly

Actividades:

- Programar el control de sentido de giro de los motores con el pulsador.
- Programar el control de velocidad de los motores con el potenciómetro.



Servomotores.



Fuente: - [Servomotor](#)

Programas Visualino:
prog09_servo01.bly
prog09_servo02.bly

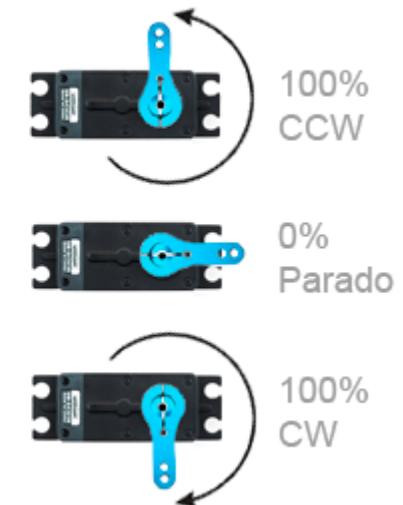
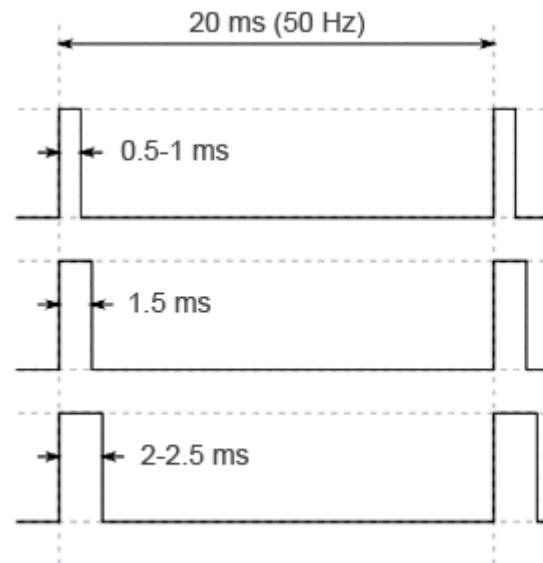
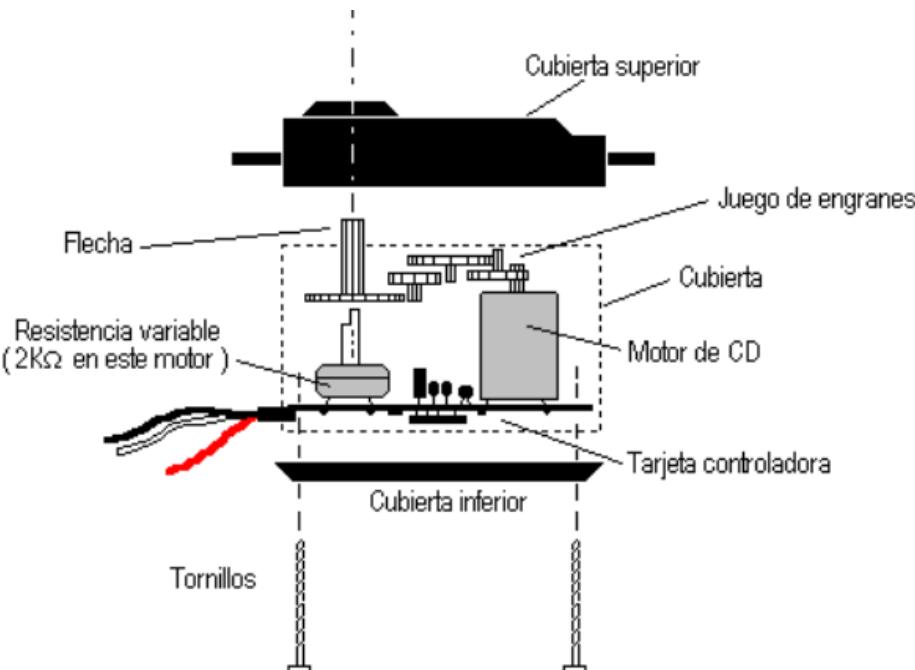
Programas de test:

- test09_ServoPin7.ino
- test10_ServoPin7pot.ino



Servomotores de rotación continua.

Tiene los mismos elementos que el servomotor, diferenciándose sólo en que la resistencia variable es fija. Este cambio hace que se llame "servomotor trucado".



Fuente: - [Servomotor de rotación continua](#)

Programas de test:

- test09_ServoPin7.ino
- test10_ServoPin7pot.ino

Programas Visualino:

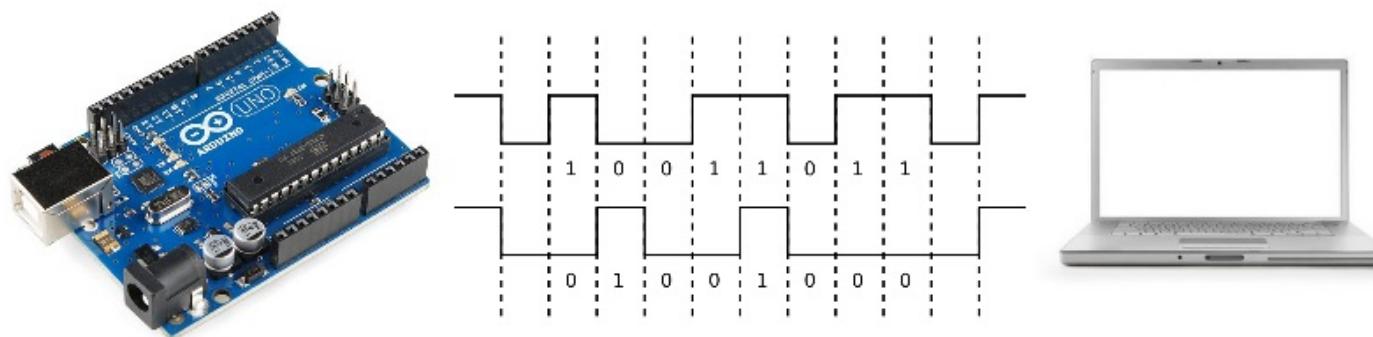
- prog10_servo01RC.bly
- prog10_servo02RC.bly



Comunicaciones Serie (USB).

El puerto serie de Arduino se usa para transmitir y recibir datos de otros dispositivos como un PC, teléfono móvil, tablet o microcontrolador.

El puerto serie del Arduino Uno usa los pins 0(RX) y 1(TX). Estos están conectados al controlador FTDI (ATmega 16u2) que es el que permite la traducción del formato serie TTL a USB. Estos pins no pueden ser utilizados mientras se usa la comunicación serie.
Más información del [puerto serie](#), la tabla de códigos ASCII se encuentra en la información.



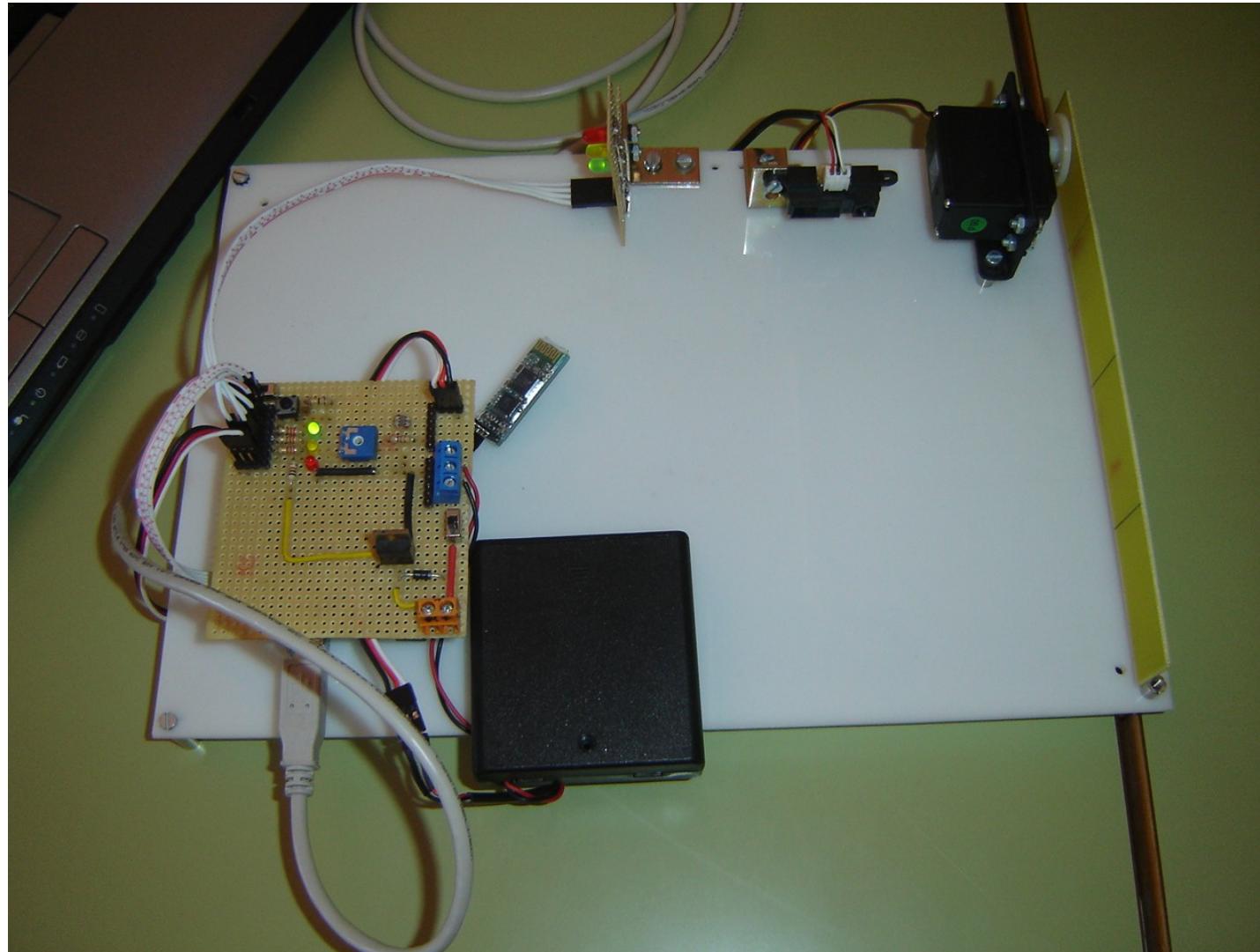
Fuente: - [Comunicaciones Serie \(USB\)](#)

Programas de test:

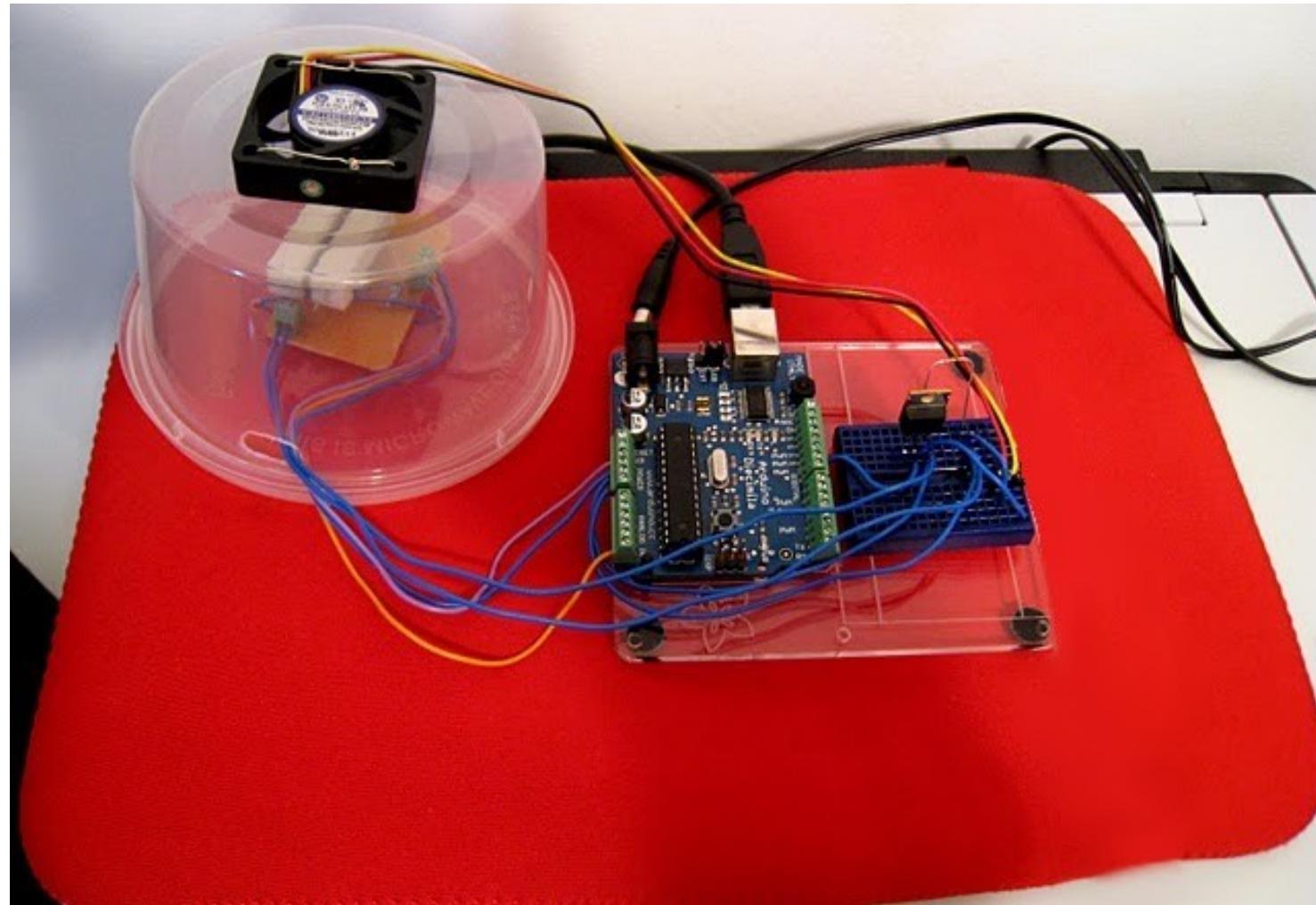
- prog11_serie01.ino
- prog11_serie02.ino

Programas Visualino:

- prog11_serie01.bly
- prog11_serie01b.bly

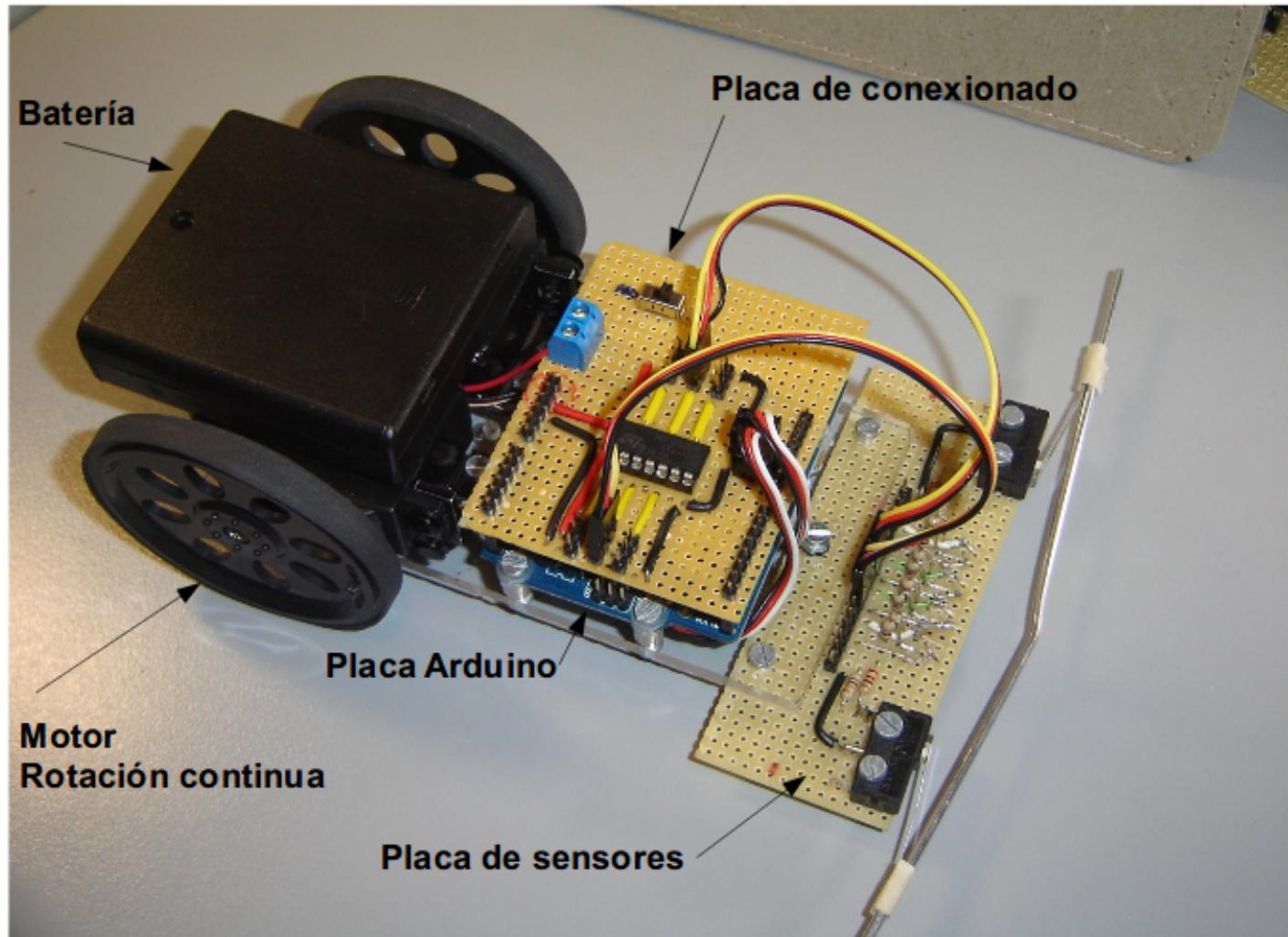


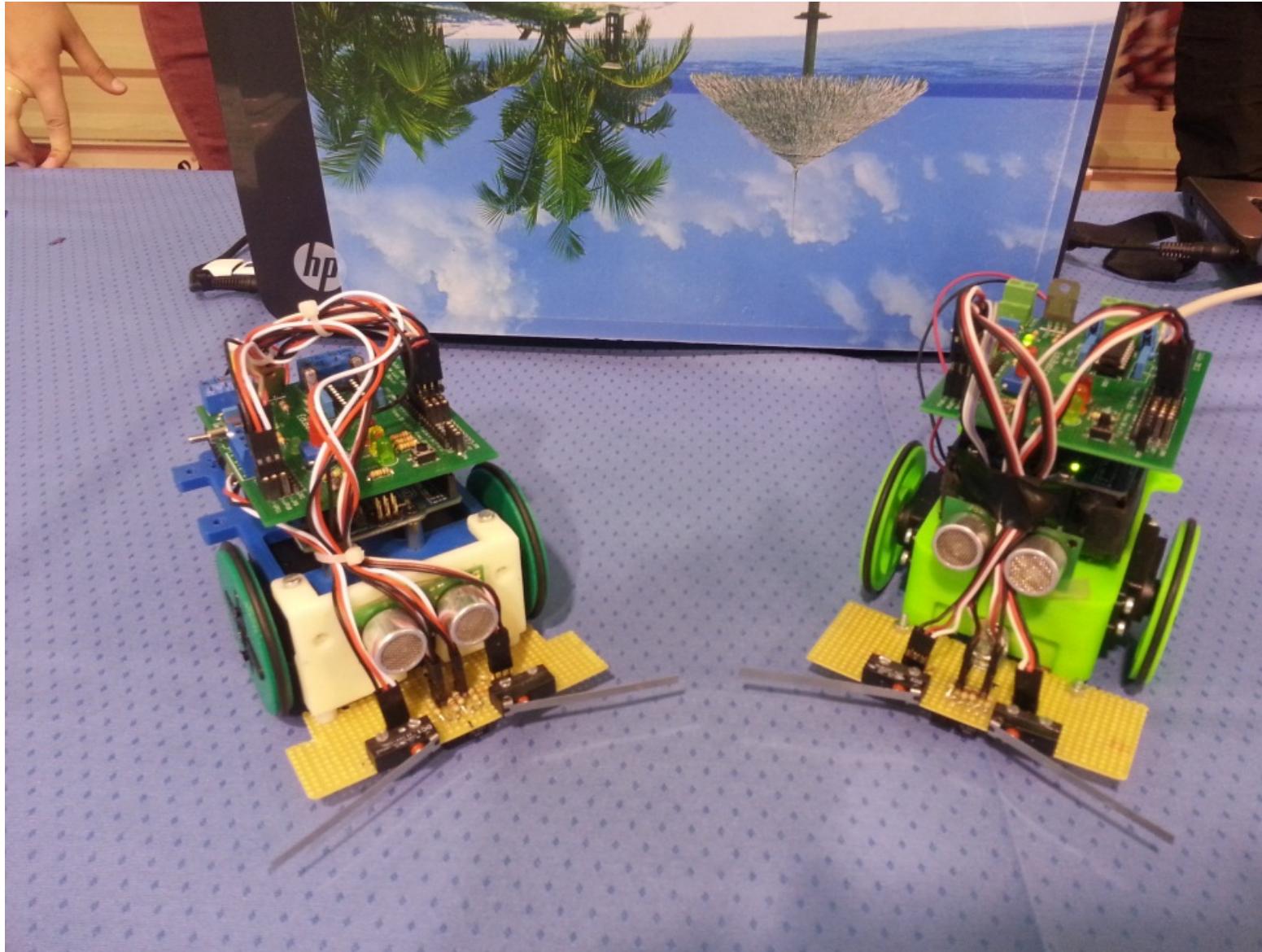


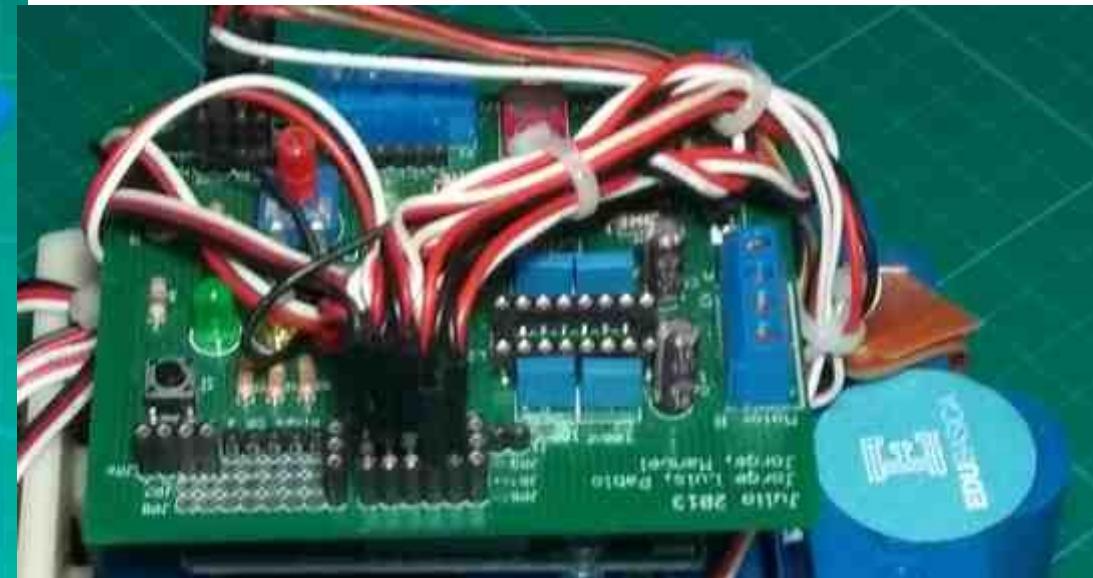
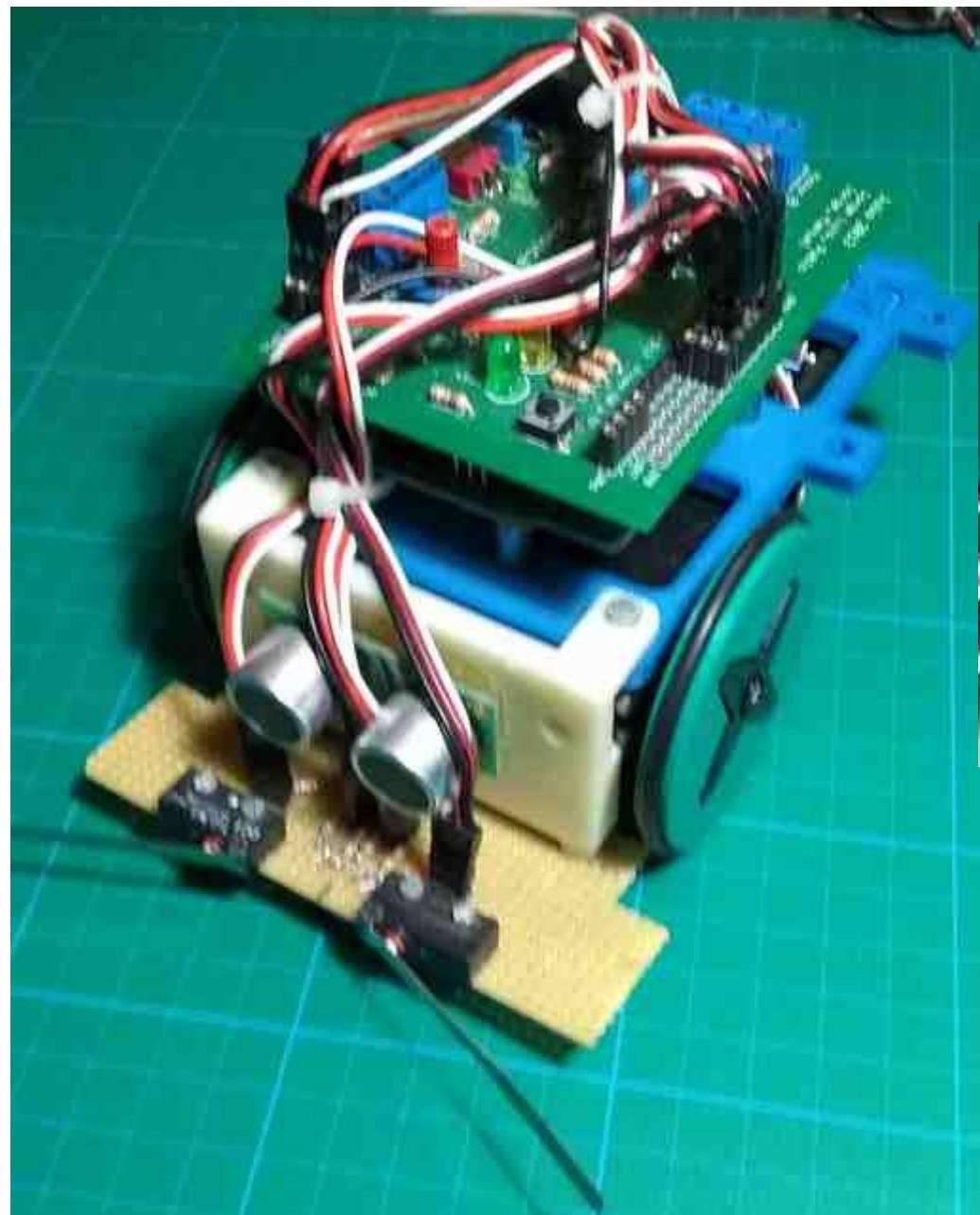


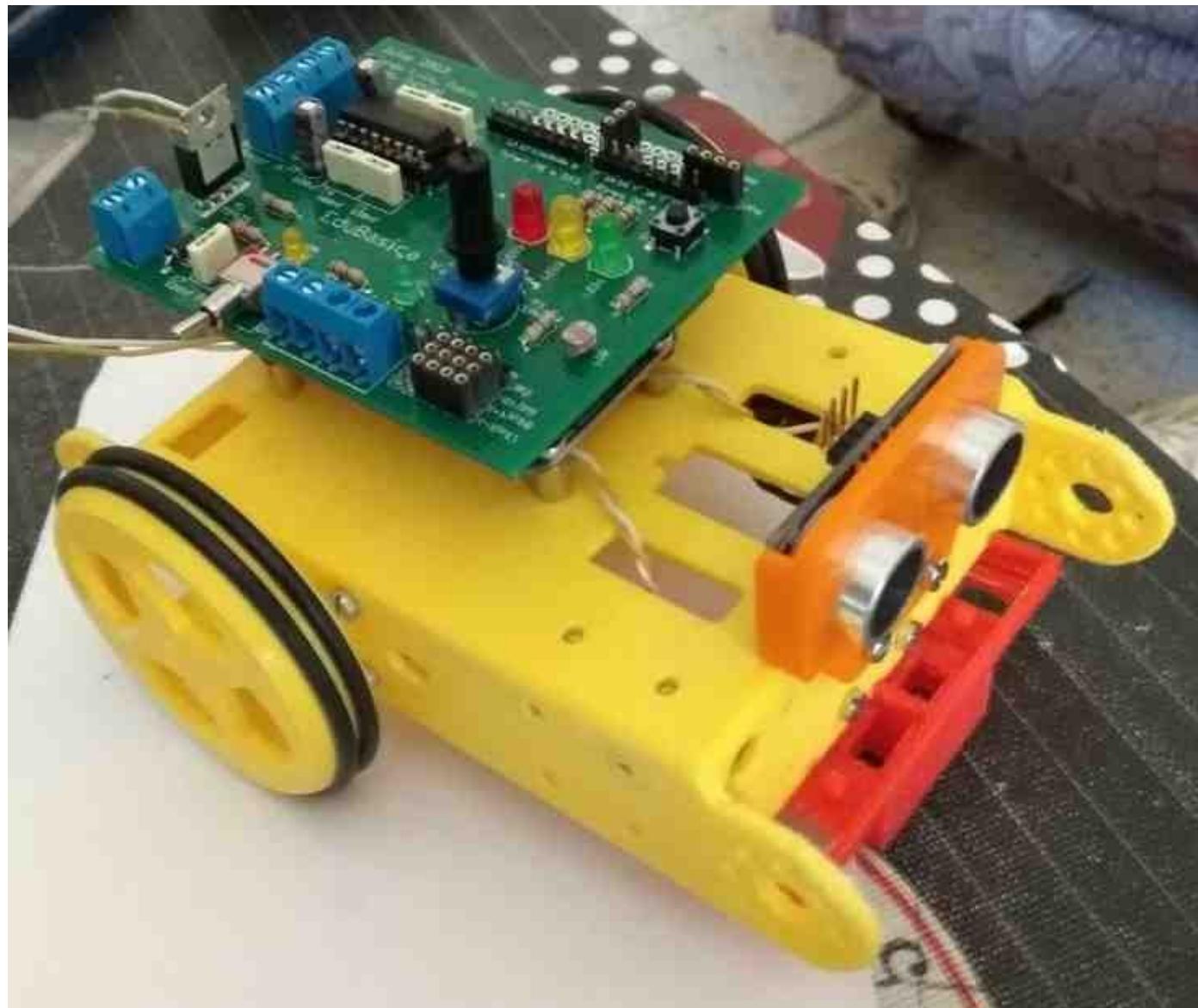


EducaBot











www.practicasconarduino.com/

Repository of EduBasica
<https://edubasica.github.io/>

Video assembly EduBasica
<https://www.youtube.com/watch?v=OG5RQnMbqs8>

Contacto
practicasconarduino@gmail.com



RED ANDALUZA
DE ROBÓTICA
Y TECNOLOGÍA
EDUCATIVA



COITTA/AAGIT
Colegio Oficial / Asociación Andaluza
de Graduados & Ingenieros
Técnicos de Telecomunicación

<http://roboticaytecnologia.org/>

MUCHAS GRACIAS

Ponente:

Manuel Hidalgo - - manuel.leobot@gmail.com

