



CFR
A CORUÑA

centro de
formación e recursos

Curso de robótica escolar con software libre



Ayuntamiento de A Coruña
Concello da Coruña

Č MUSEOS
CIENTÍFICOS



Bricolabs

PROGRAMACIÓN POR SESIONES

1. Introducción a la robótica y montaje inicial

- Montaje mecánico, instalación del software.
- Prueba y ajuste de programas básicos de movimiento.

2. Programación de Arduino y sensores

- Iniciación al entorno de Arduino.
- Circuitos y programas con sensores y actuadores.

3. Robots autónomos con sensores

- Mantener la distancia, evitar o buscar obstáculos, seguir líneas.

4. Control sin hilos. Proyectos y concursos

- Control por bluetooth y enlace con Android.
- Proyectos colaborativos, ferias tecnológicas, concursos.

5. Conclusiones y propuestas

- Resumen de recursos.
- Propuestas de futuro.

INTRODUCCIÓN

- ¿Qué es la robótica?
Origen del nombre
- ¿Qué vamos a hacer?
- ¿Qué queréis hacer?
Volveremos al final con
la pregunta



¿Qué se puede llegar a hacer?

Alumnos del CEIP de Os Dices ganan un premio de robótica de la Agencia Espacial Europea

Publicado en 25 de octubre de 2013



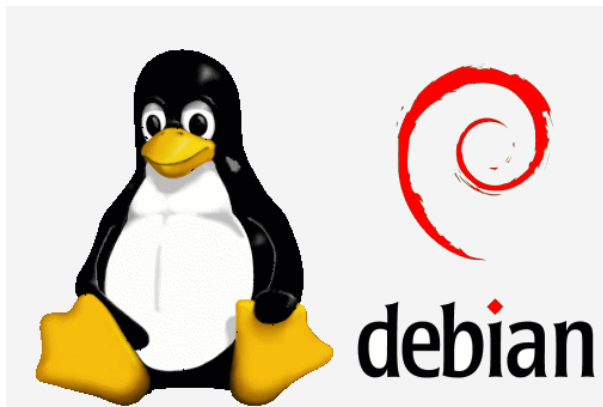
SOFTWARE LIBRE

El término se refiere al software que puede ser copiado, estudiado, modificado, utilizado libremente con cualquier fin y redistribuido con o sin cambios o mejoras.

Ejemplos y proyectos:

GNU/Linux, Libre Office, Open SCAD, R y R-Studio, Arduino IDE

Libre no es lo mismo que gratis.



HARDWARE LIBRE

Se llama hardware libre a aquellos dispositivos cuyas especificaciones y diagramas son de acceso público.

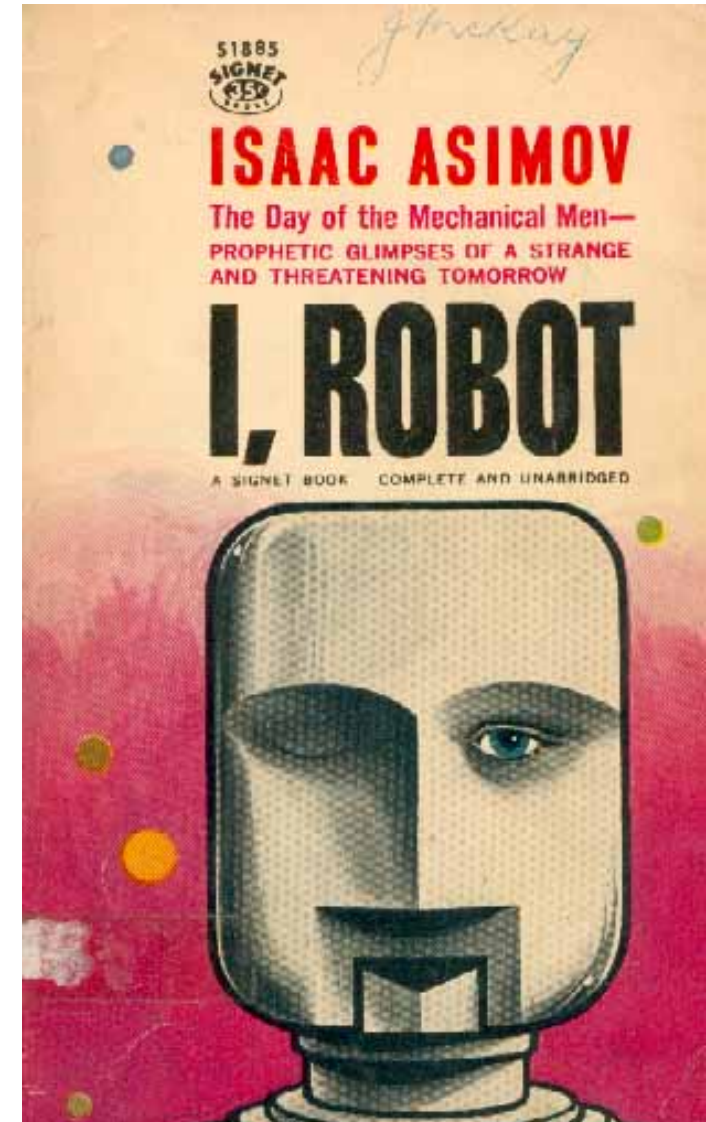
Ejemplos y proyectos: Interacción con el mundo físico

- Microcontroladores: Arduino, Beagle, pcDuino...
- Ordenador de placa única: Raspberry Pi, Beaglebone Black...



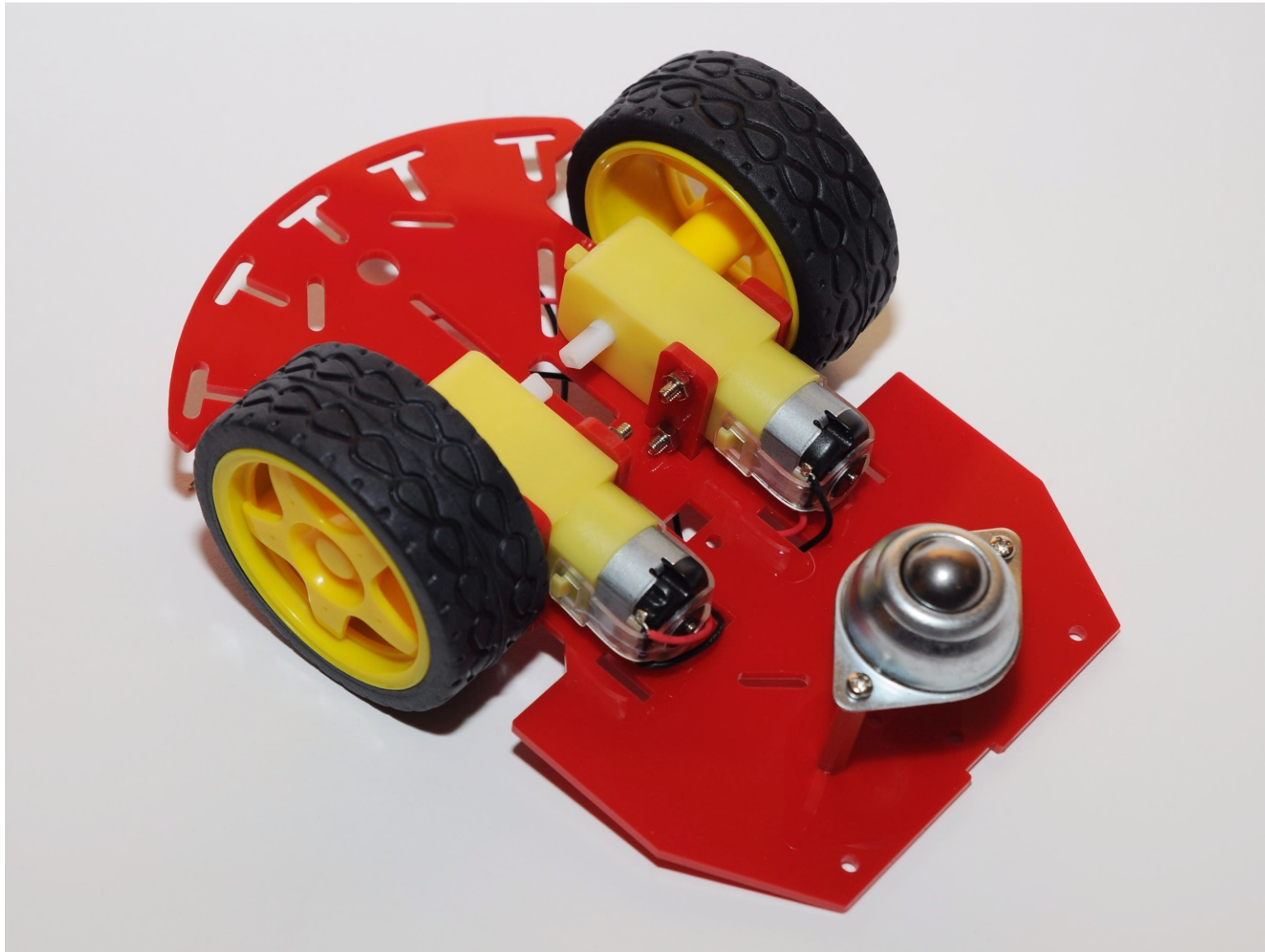
LEYES DE LA ROBÓTICA. YO ROBOT

1. Un robot no puede hacer daño a un ser humano o, por inacción, permitir que un ser humano sufra daño.
2. Un robot debe obedecer las órdenes dadas por los seres humanos, excepto si estas órdenes entrasen en conflicto con la 1ª Ley.
3. Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la 1ª o la 2ª Ley.

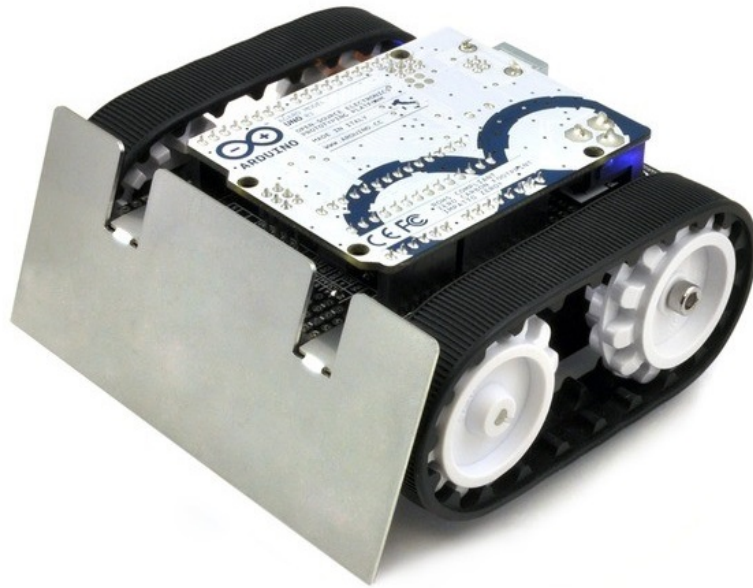


MONTAJE MECÁNICO DEL ROBOT

Montaje de la rueda loca inferior

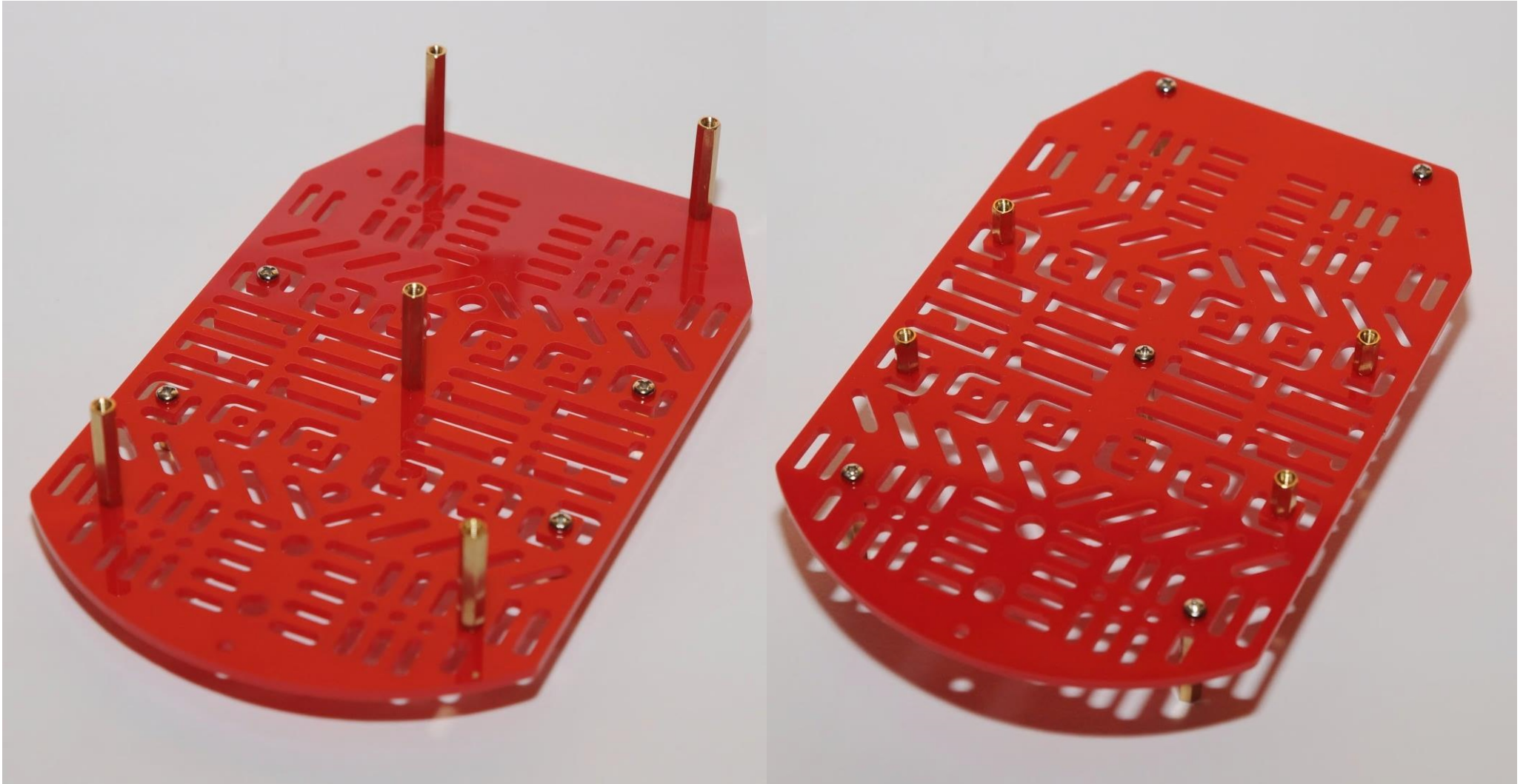


- Porqué dos motores y una rueda loca
- Otras opciones: cuatro o más ruedas, cadenas.

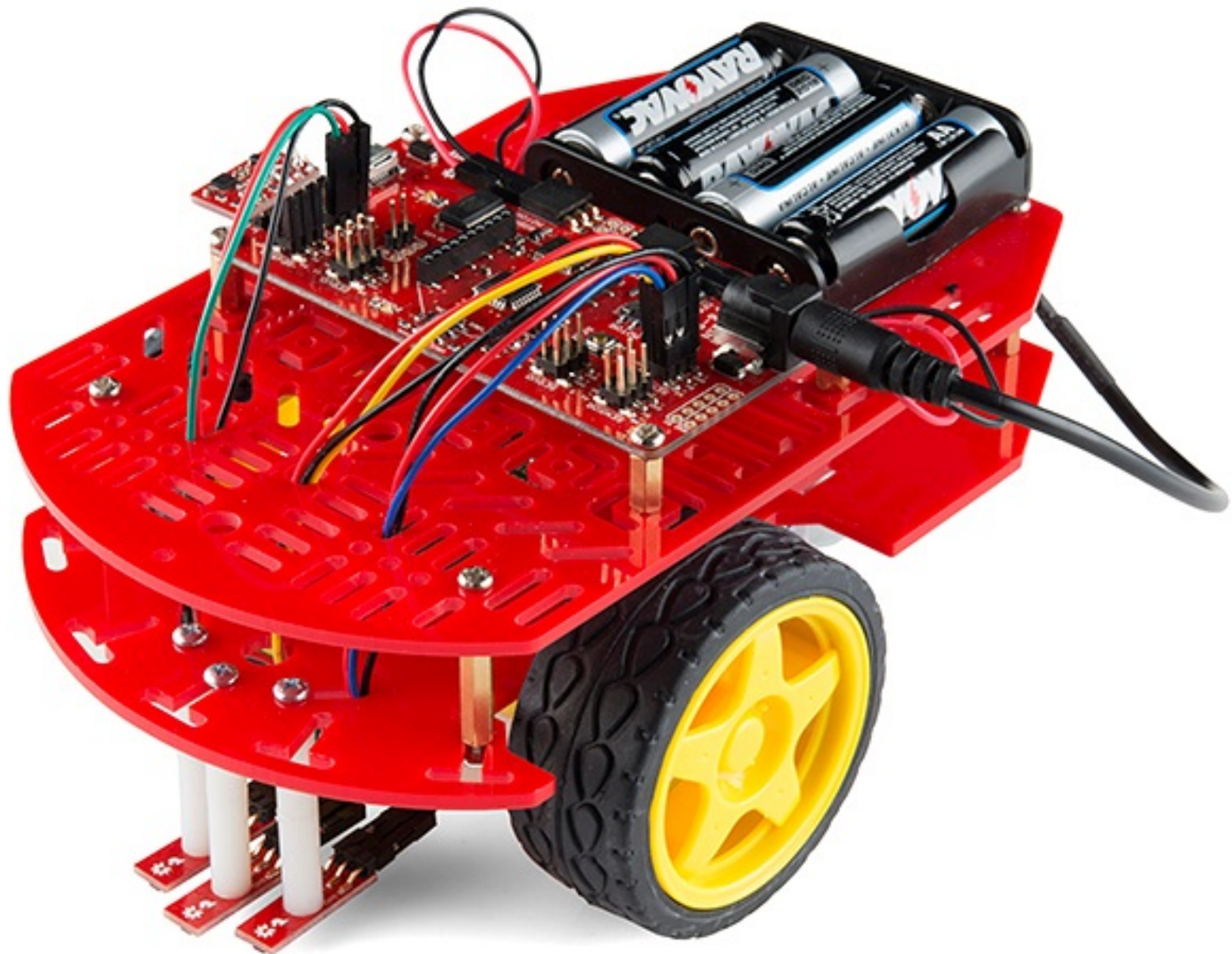


- Fotos por pasos. 97 piezas!!!

Montaje de la placa superior del chasis



Montaje final



MONTAJE ELECTRÓNICO DEL ROBOT

Arduino: Código y electrónica modular



```
robotwithlibrary
#include <RobotMovil.h>
RobotMovil robot;

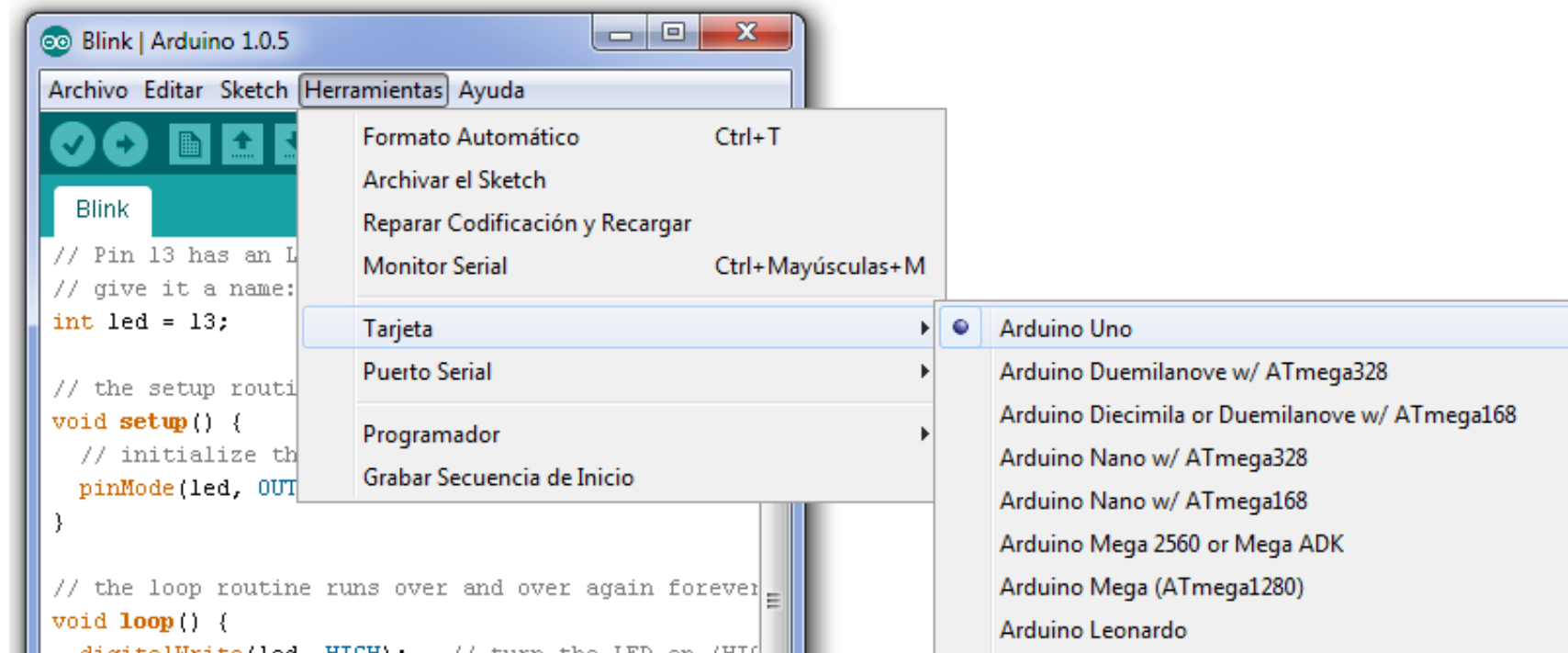
void setup()
{
}

void loop()
{
  robot.forward(80,80,1000);
}
```

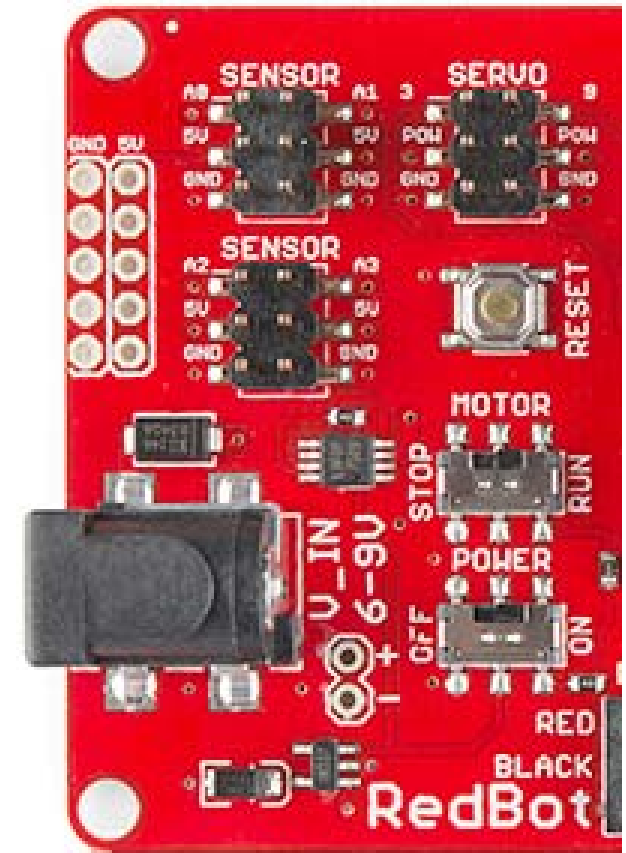
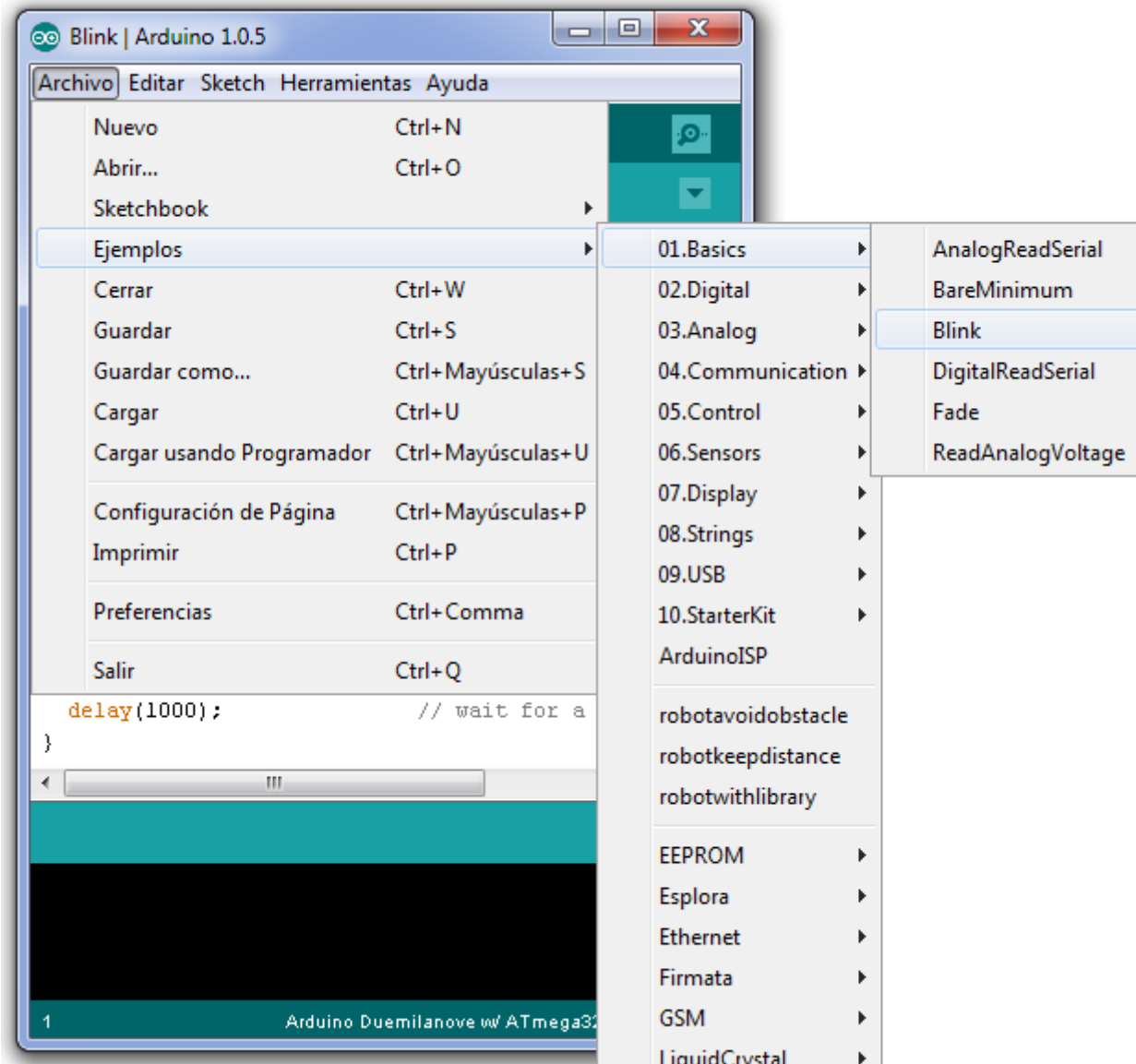
1 Arduino Uno on COM7

INSTALACIÓN Y PRUEBA DE ARDUINO IDE (software de desarrollo)

- Instalación con pendrive (monitores)

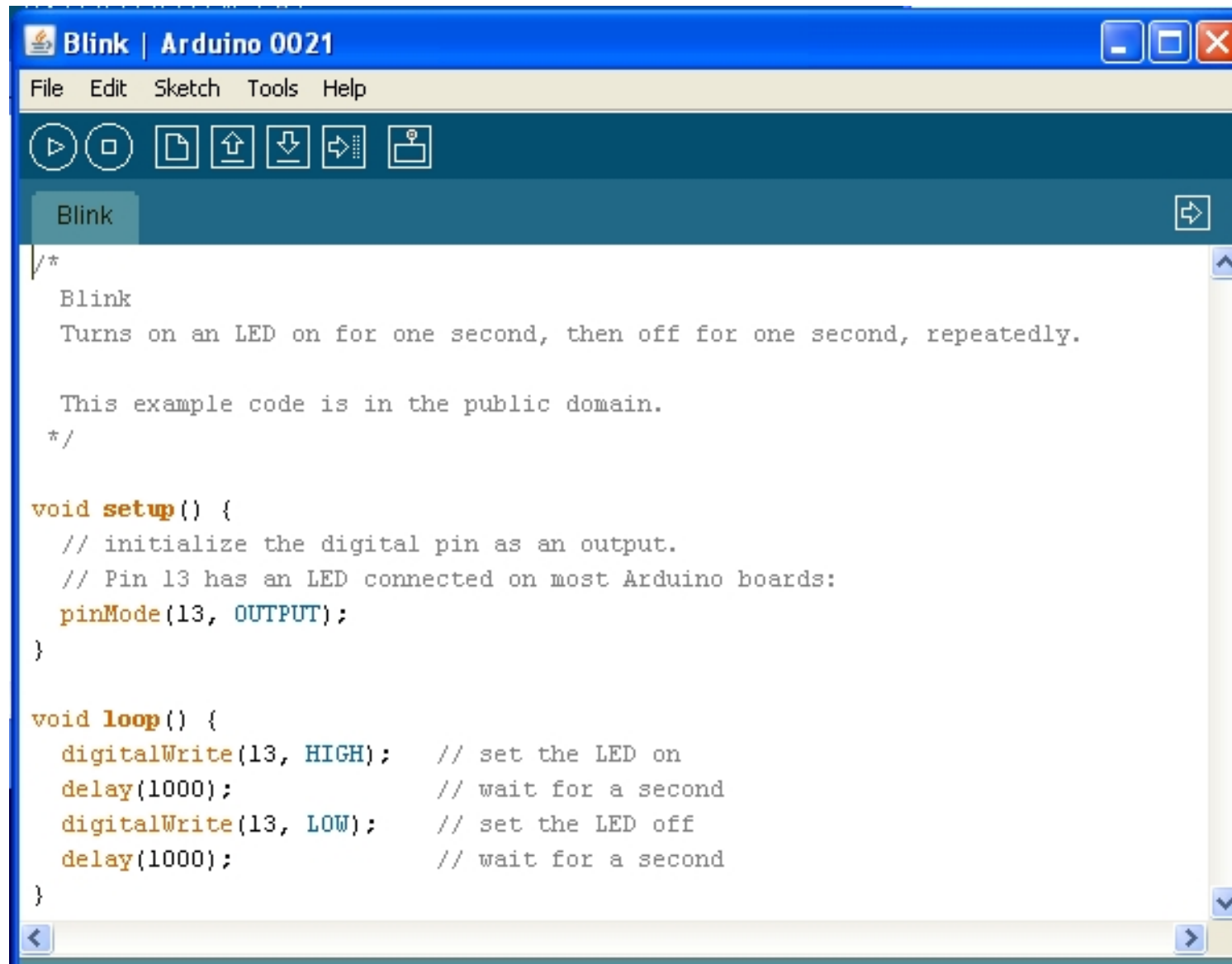


- Prueba de comunicación con Blink



Power ON
Motor STOP

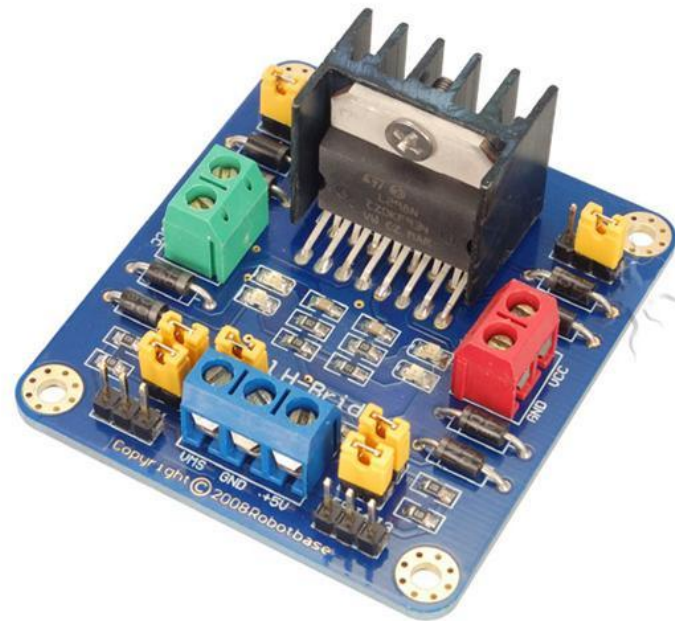
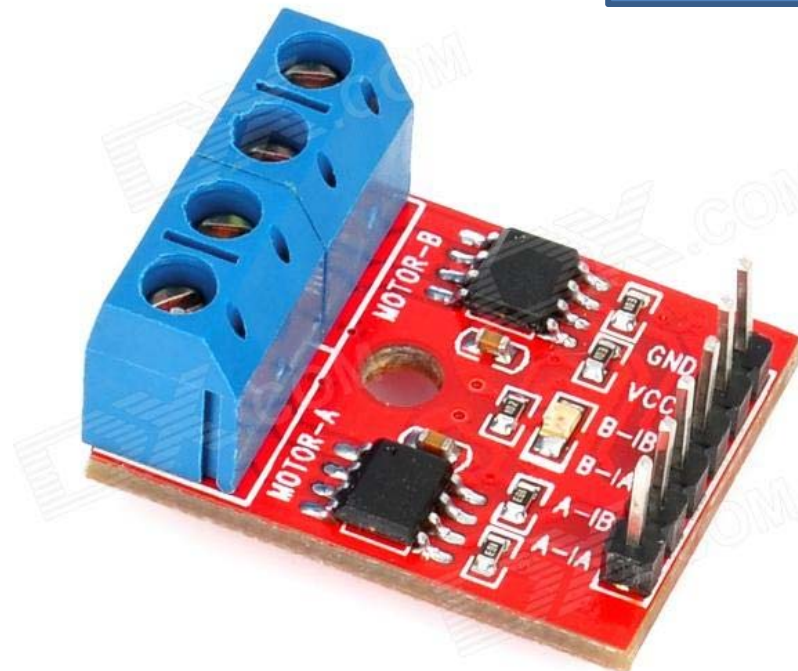
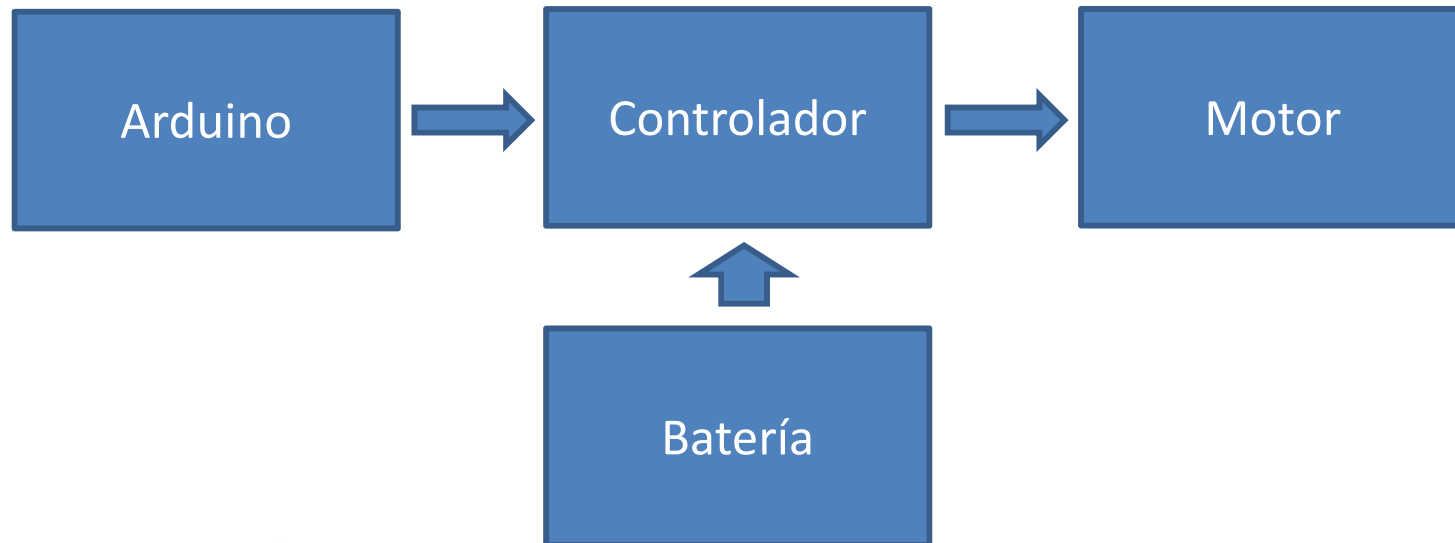
Código de Blink: salida de información y órdenes



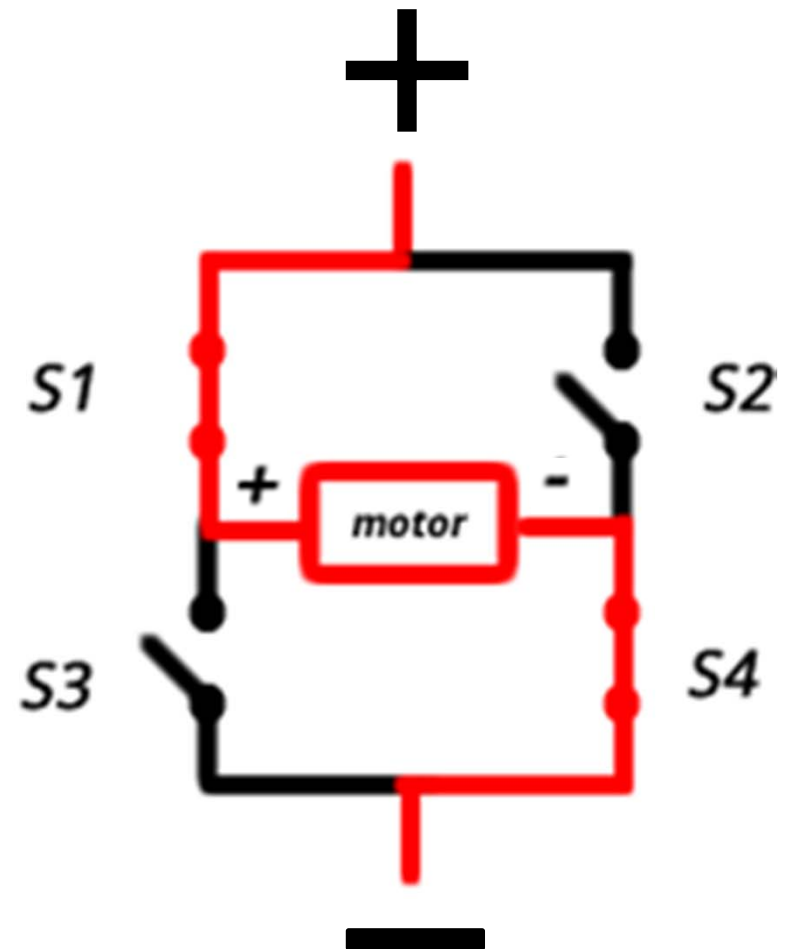
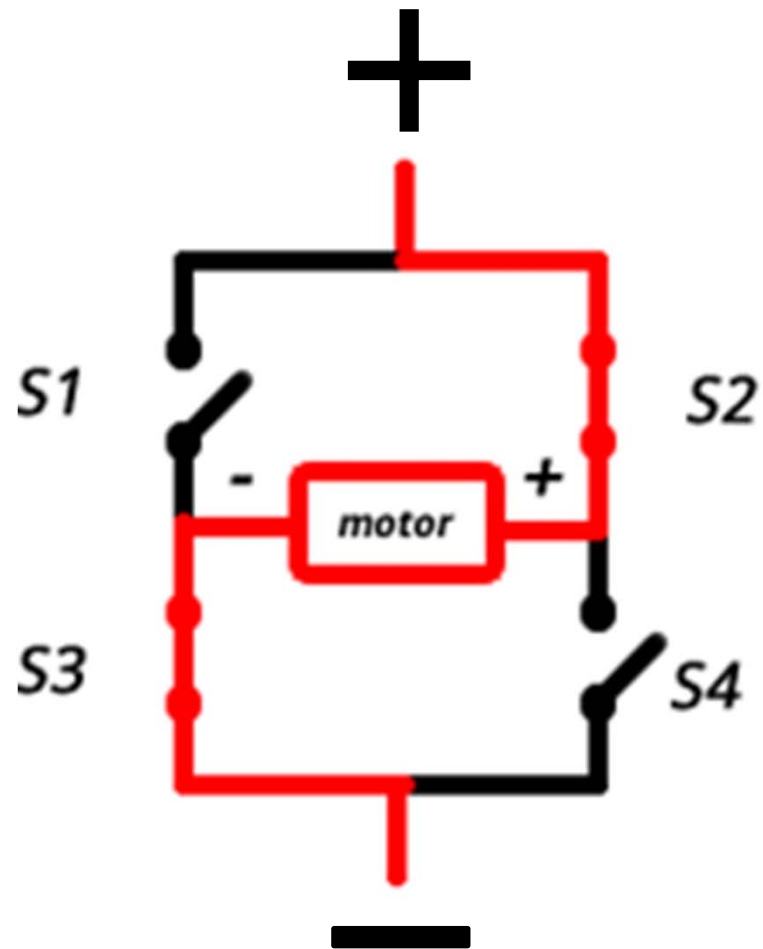
The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 0021". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, opening, and other sketch-related actions. The main text area displays the code for the "Blink" sketch. The code includes a multi-line comment describing the sketch's function and a license statement, followed by the `void setup()` and `void loop()` functions. The `setup()` function initializes pin 13 as an output. The `loop()` function sets the LED on for 1000ms and off for 1000ms in a repeating cycle.

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  // set the LED on  
  delay(1000);             // wait for a second  
  digitalWrite(13, LOW);   // set the LED off  
  delay(1000);             // wait for a second  
}
```

CONEXIONADO DE MOTORES

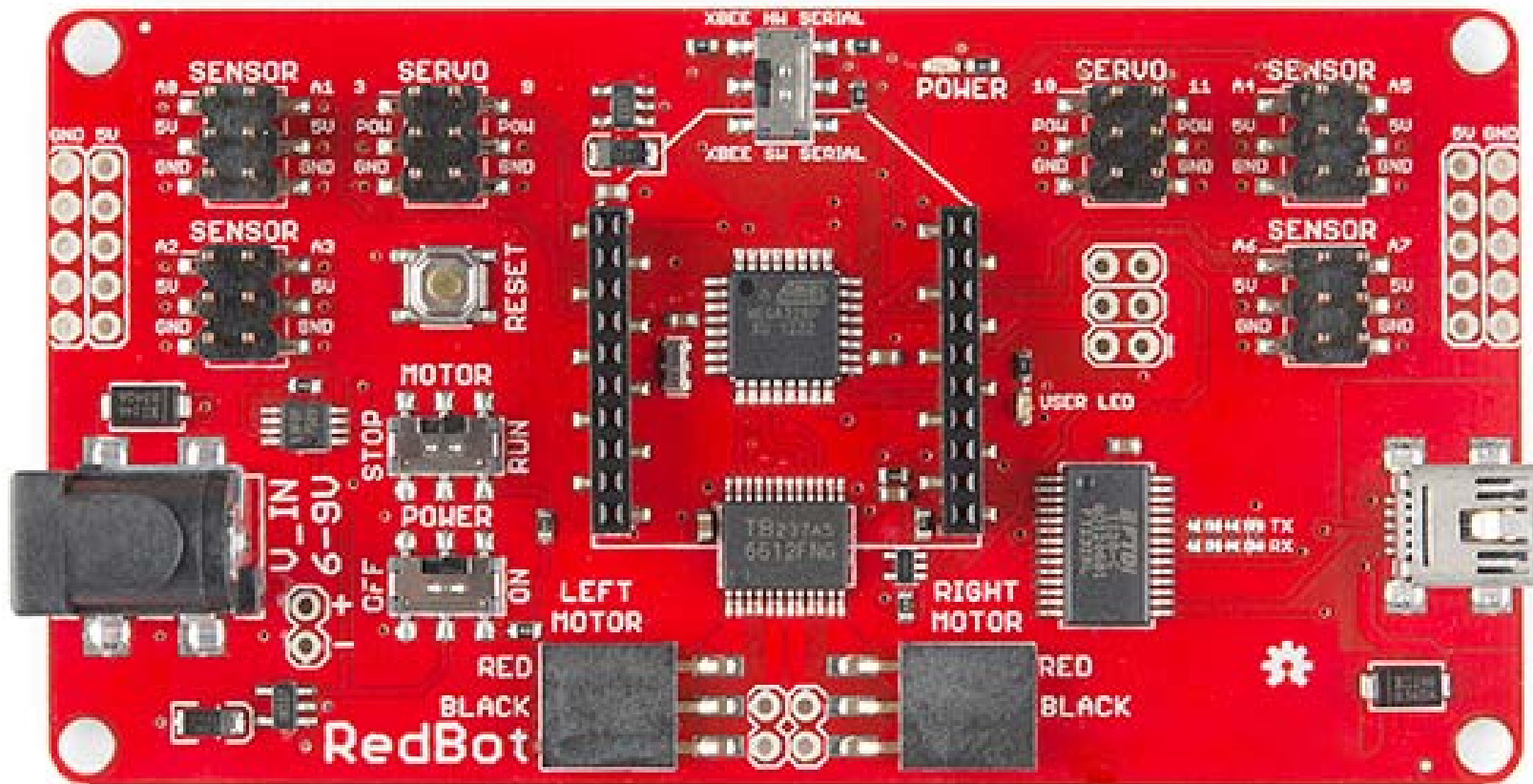


Puente en H para invertir el giro

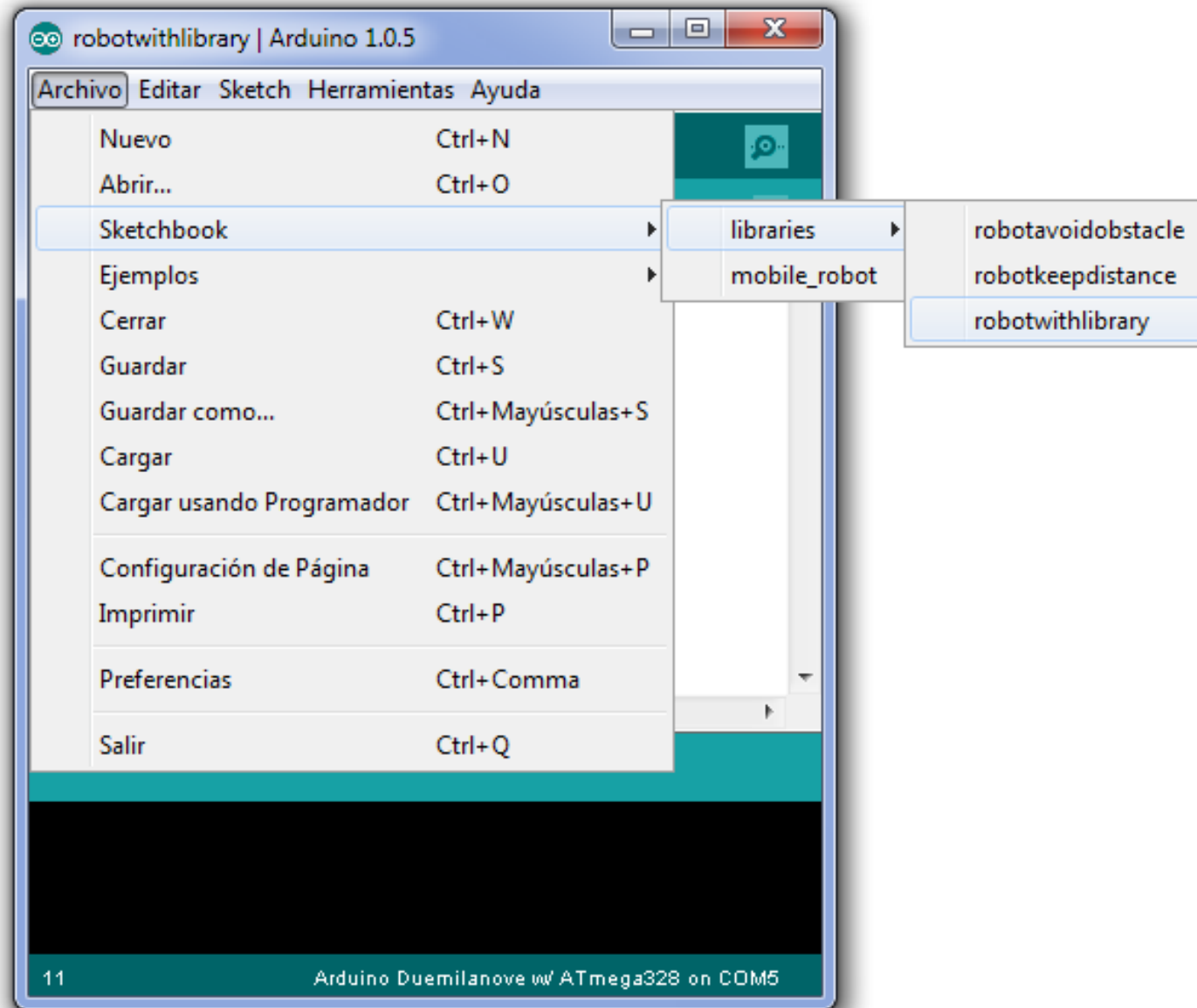


Controlador integrado en placa

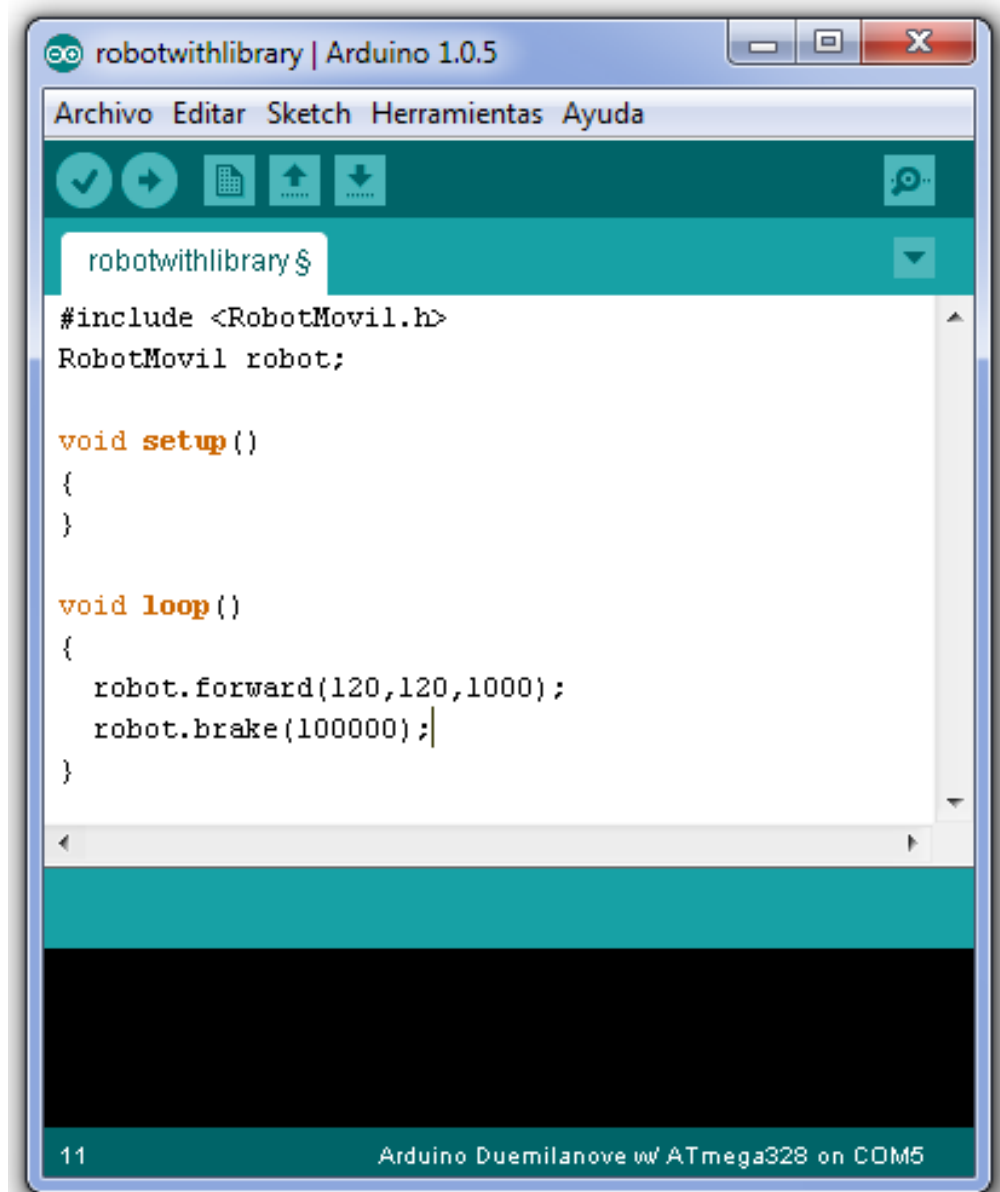
- Conexionado y revisión de polaridad



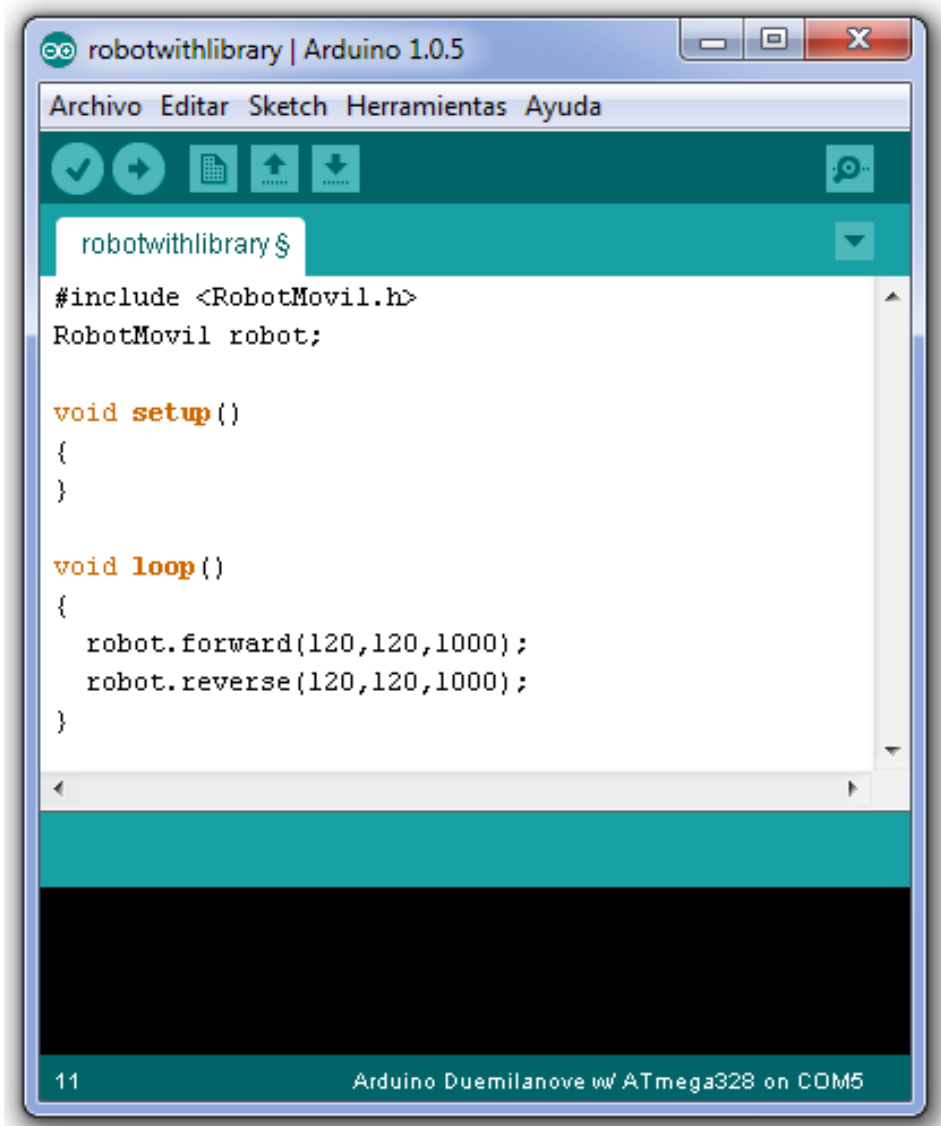
LIBRERÍA ROBOTMOVIL Y EJEMPLOS



- Comprobación de movimiento adelante



- Adelante y atrás
- Más rápido, más lento. Límites.
- Caminar recto. Ajuste fino (tunning).



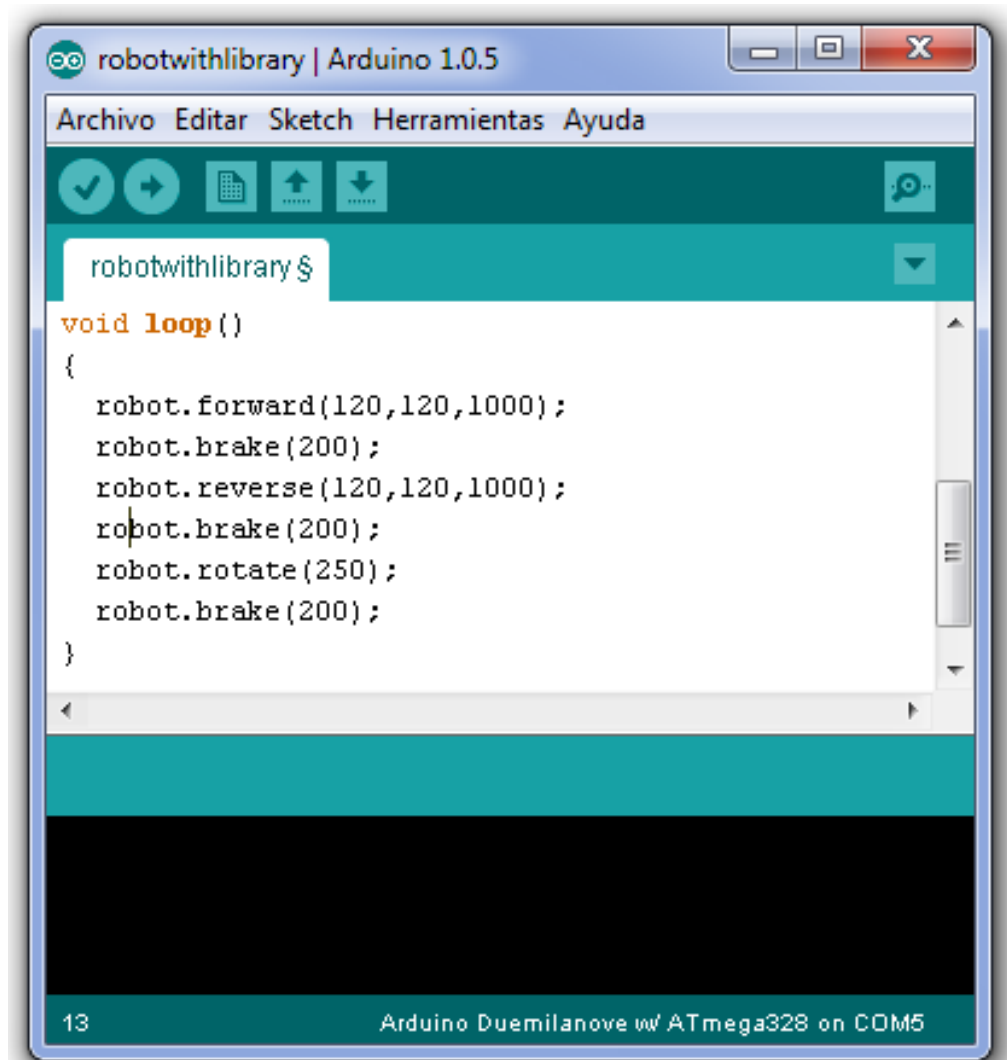
SECUENCIAS DE MOVIMIENTOS

robot.forward

robot.reverse

robot.brake

robot.rotate



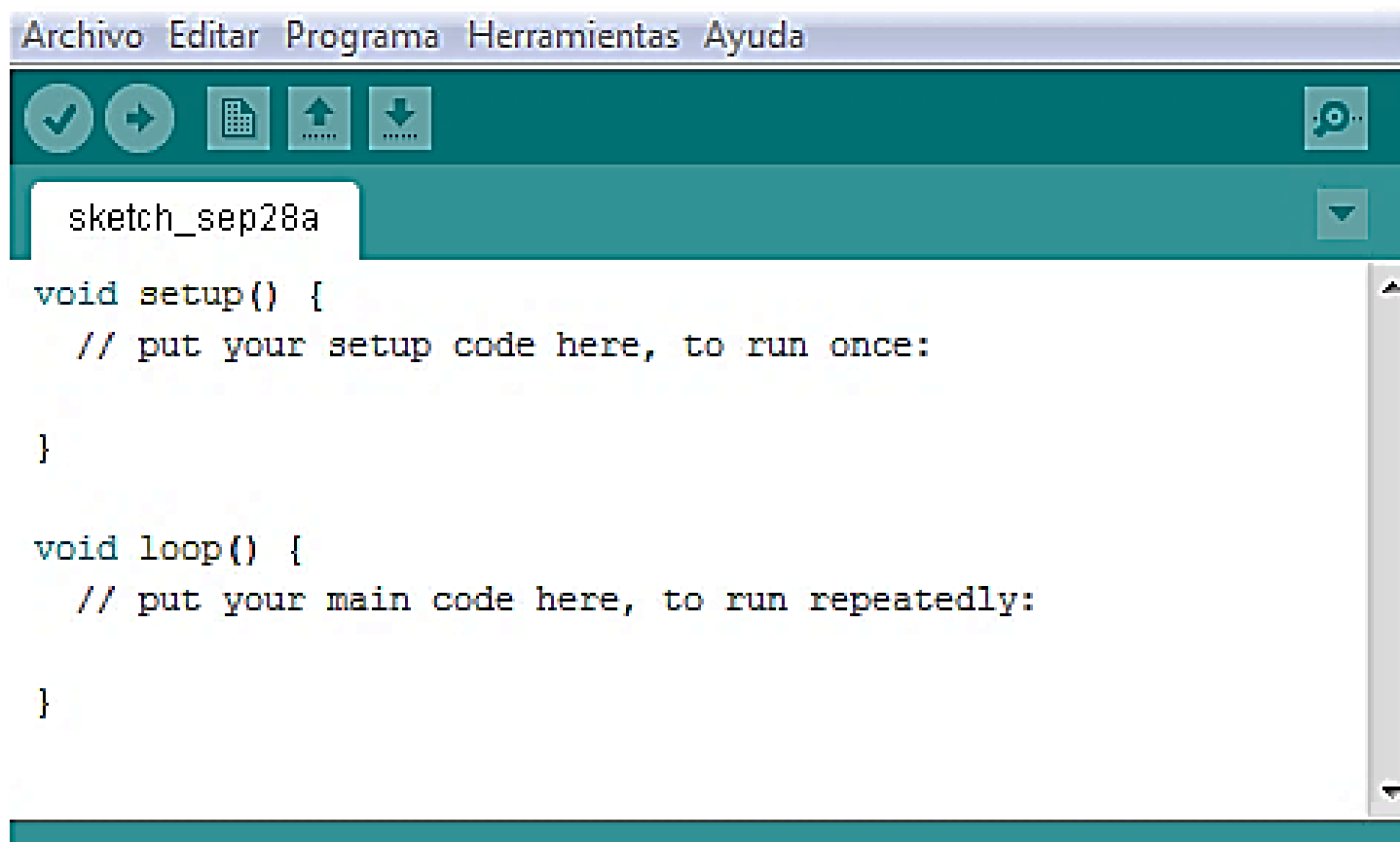
The screenshot shows the Arduino IDE interface with the title bar 'robotwithlibrary | Arduino 1.0.5'. The menu bar includes 'Archivo', 'Editar', 'Sketch', 'Herramientas', and 'Ayuda'. The toolbar contains icons for checking, running, uploading, and downloading. The main text area shows the following code:

```
robotwithlibrary $
void loop()
{
  robot.forward(120,120,1000);
  robot.brake(200);
  robot.reverse(120,120,1000);
  robot.brake(200);
  robot.rotate(250);
  robot.brake(200);
}
```

The status bar at the bottom indicates '13' and 'Arduino Duemilanove w/ ATmega328 on COM5'.

ARDUINO SETUP Y LOOP

- La parte inicial (setup) se ejecuta una vez.
- La segunda parte (loop) se ejecuta en bucle indefinido

A screenshot of the Arduino IDE interface. The menu bar at the top includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. Below the menu is a toolbar with icons for checking, running, uploading, and downloading. The main text area shows the code for a sketch named 'sketch_sep28a'. The code defines two functions: 'void setup()' and 'void loop()'. The 'setup' function contains a comment indicating where to place code that runs once. The 'loop' function contains a comment indicating where to place code that runs repeatedly.

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