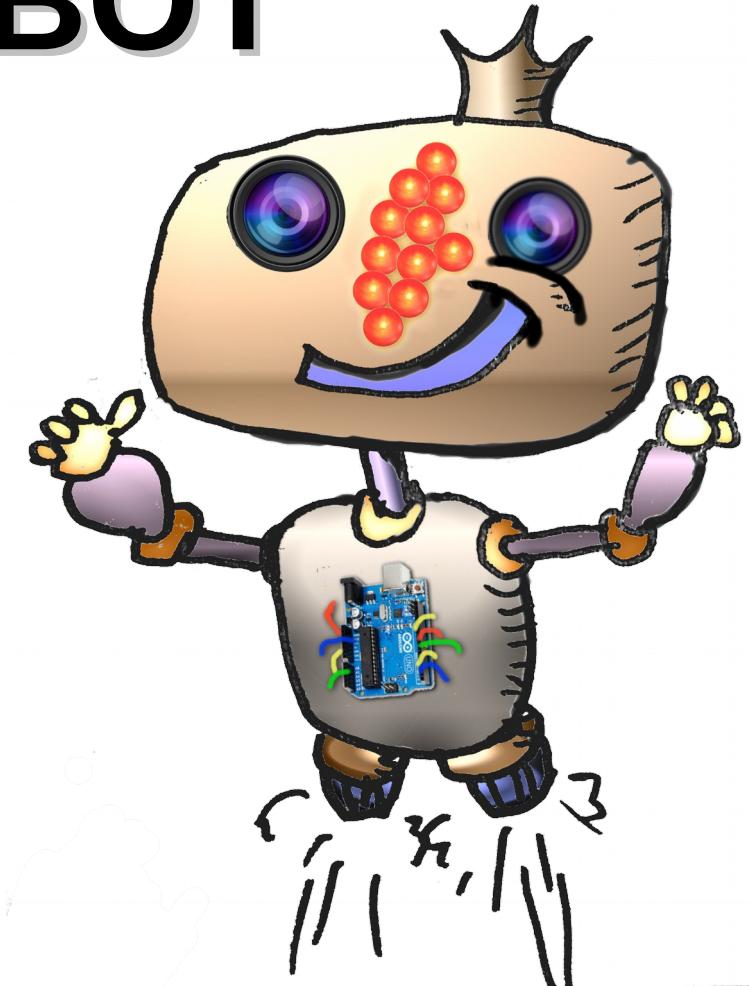


TALLER DE ESCORNABOT

**Club de Robótica de
Granada**

<https://clubroboticagranada.github.io/>



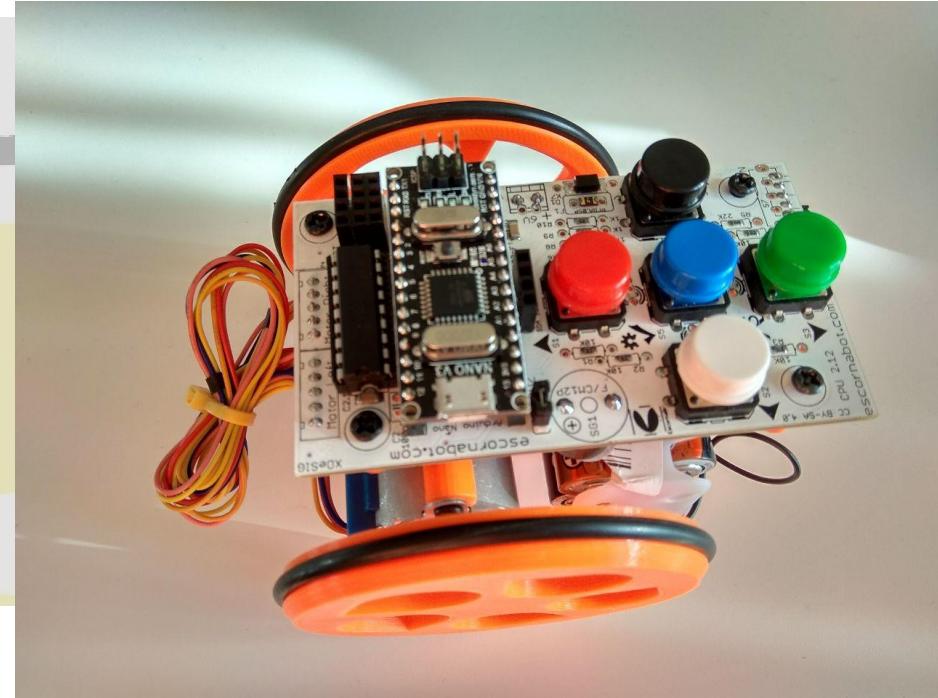
¿Escorna... qué?

ESCORNABOI:
escarabajo
lucanus cervus
en gallego

+

BOT:
robot

lunes, 11 de noviembre de 2019

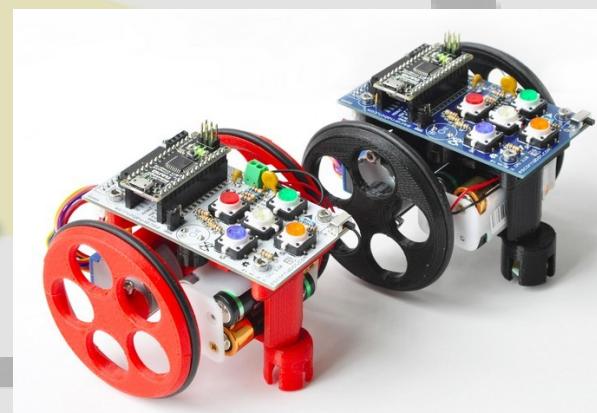


Objetivo

ROBÓTICA Y PROGRAMACIÓN



Sustituye a robots privativos



Características



- DIY: Lo haces tú
- Hardware abierto y software libre
- Asequible
- Bien documentado



¿Quién?

Proyecto liderado y soportado por la comunidad:
Profesores, diseñadores, desarrolladores,
traductores...

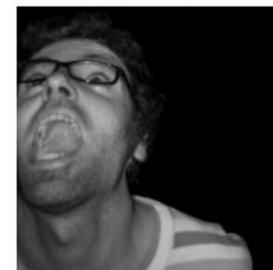


Los tres creadores

Tucho Méndez
[@procastino](#)



Xoán Sampaio
[@xoan](#)



Rafa Couto
[@caligari_pub](#)



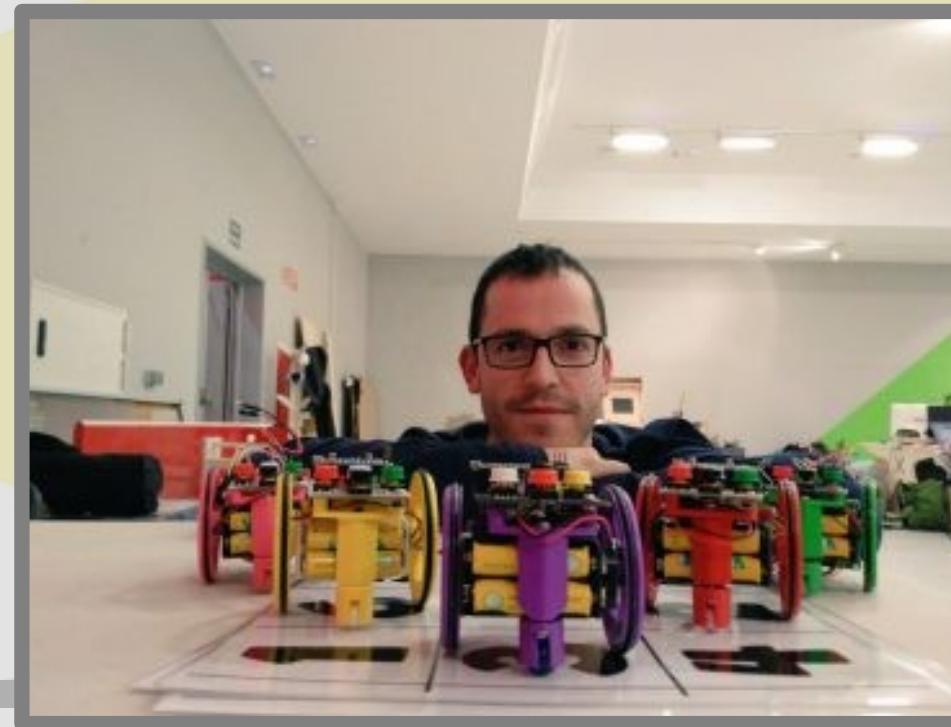
Comunidad: Grupo de Google / Telegram

Github <https://github.com/orgs/escornabot/people/>

iii Escornafan !!!

Pablo Rubio

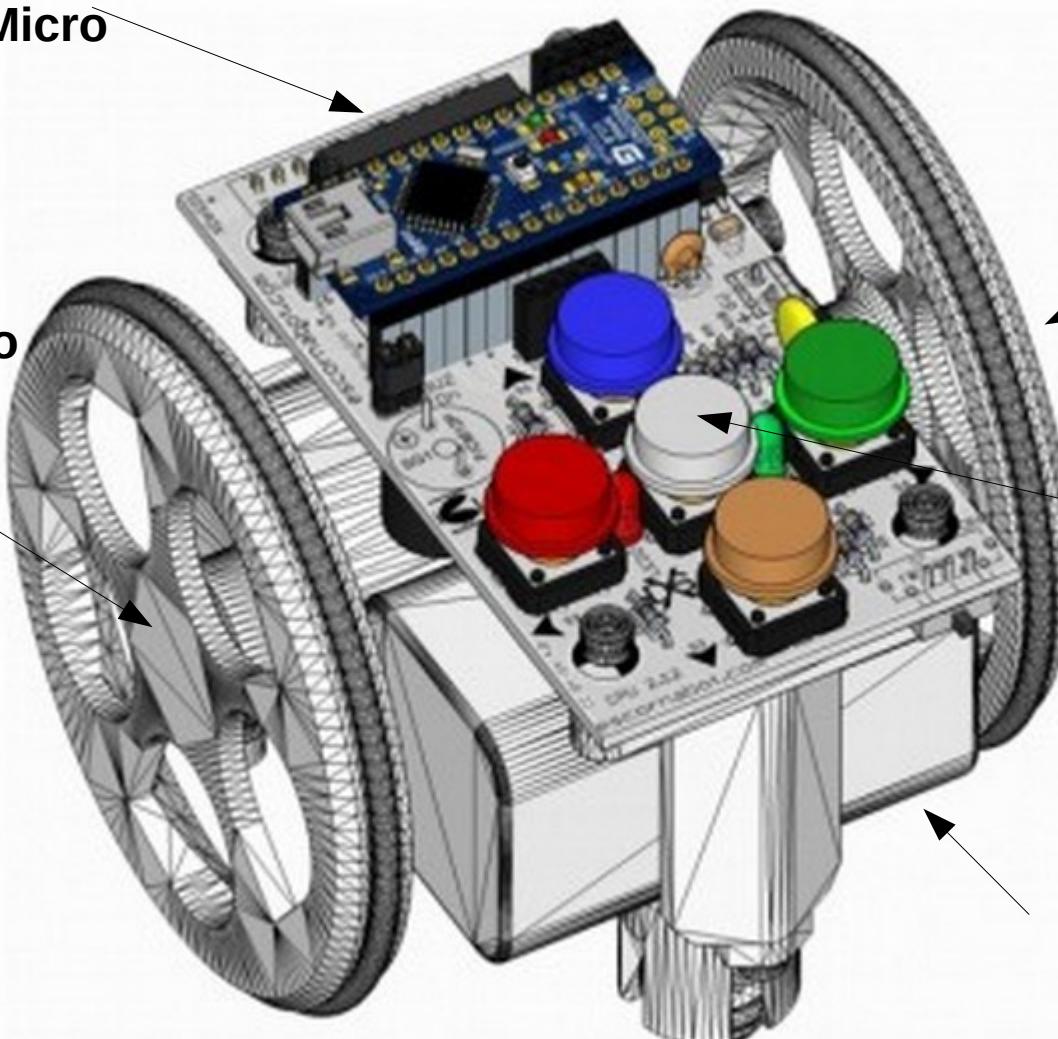
<https://pablorubma.cc/>



Escornabot

Sistema de Control
(Programación)
Arduino Nano o Micro

Actuador 2
Motor Paso a Paso



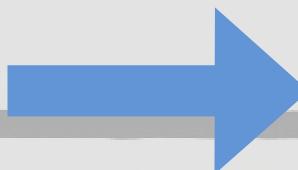
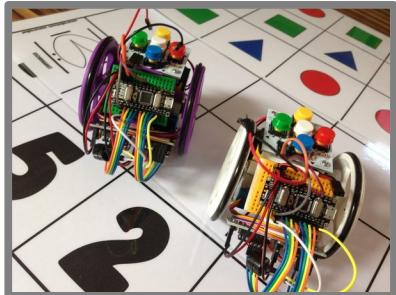
Actuador 1
Motor Paso a Paso

Sensores
5 botones

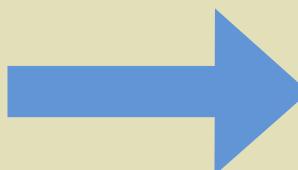
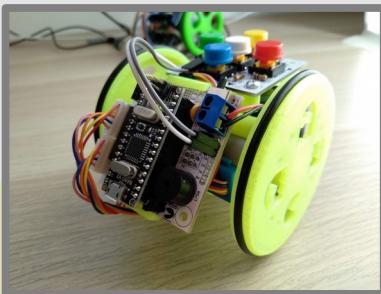
Alimentación
4 pilas



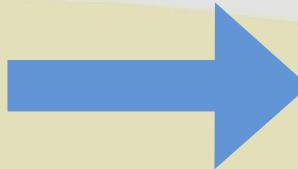
Versiones



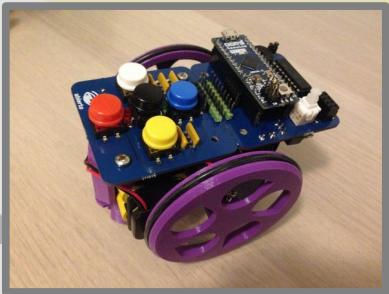
DO IT YOURSELF



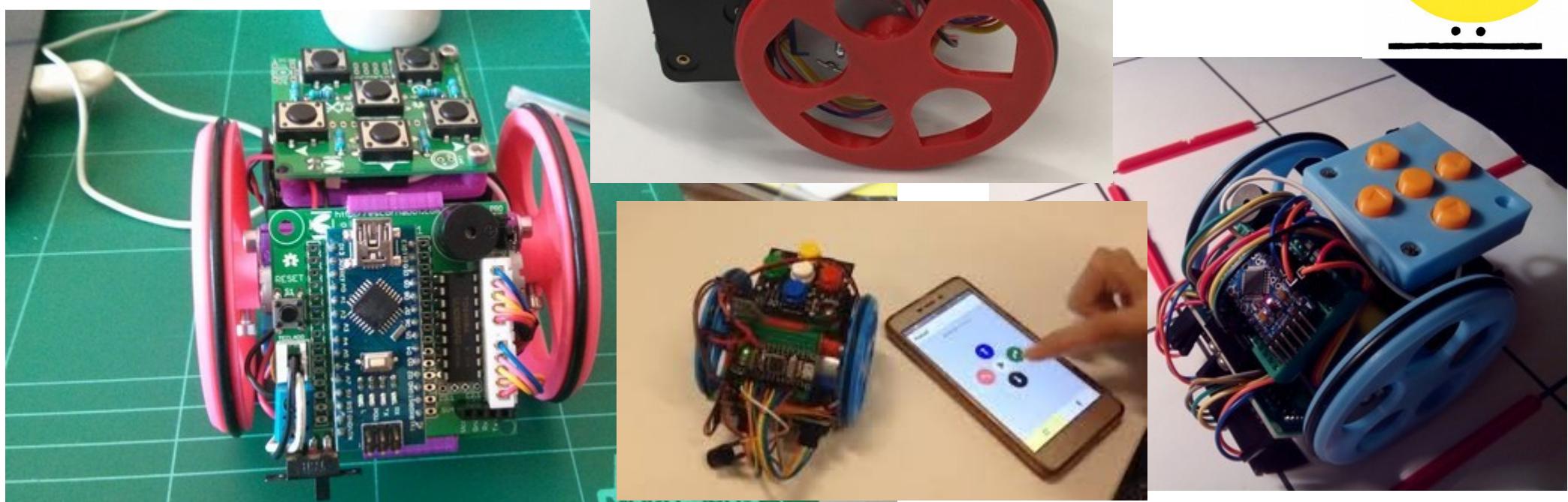
COMPACTUS



PLACA 2.12

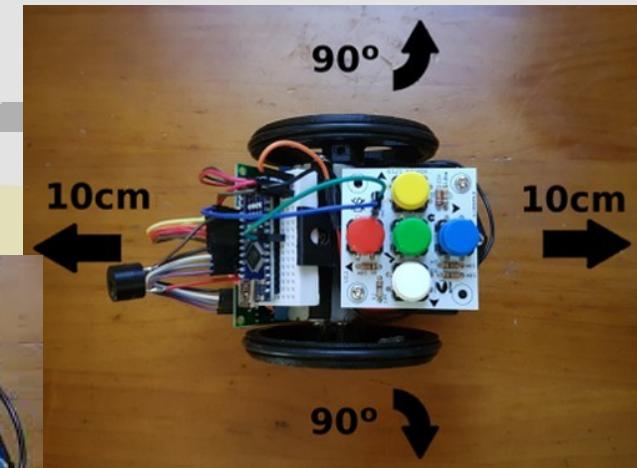
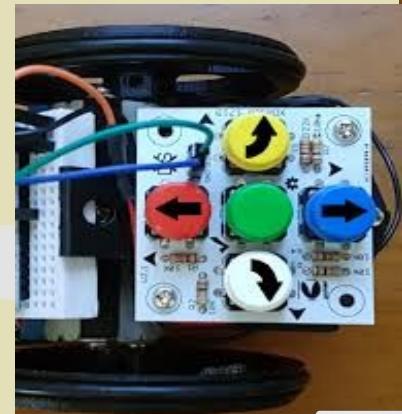


OKAGI

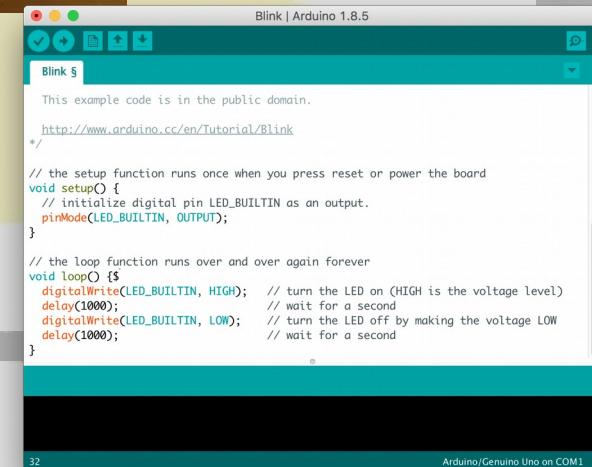


Funcionamiento y programación

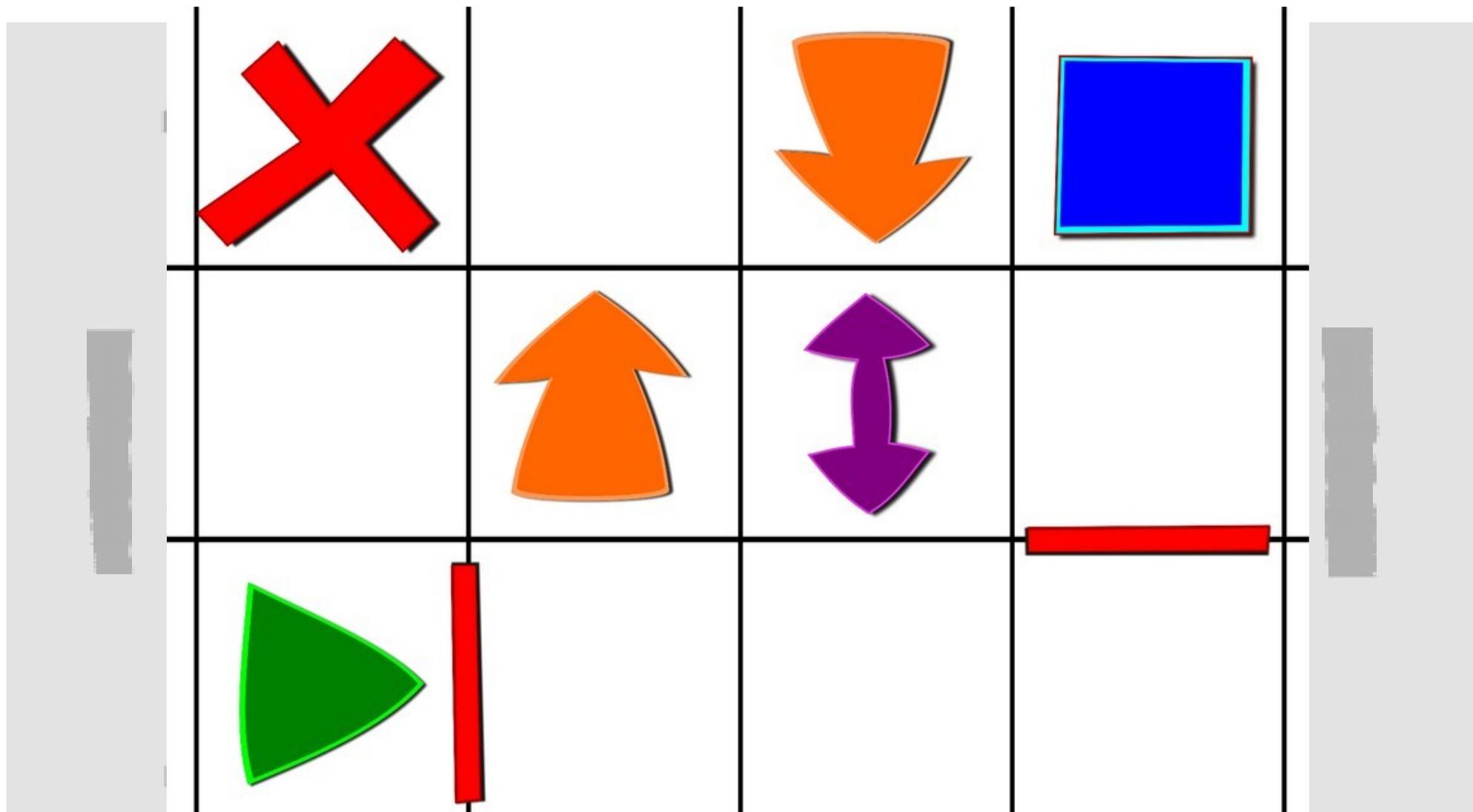
- Introduce firmware preparado y se maneja con botonera (modo clásico)

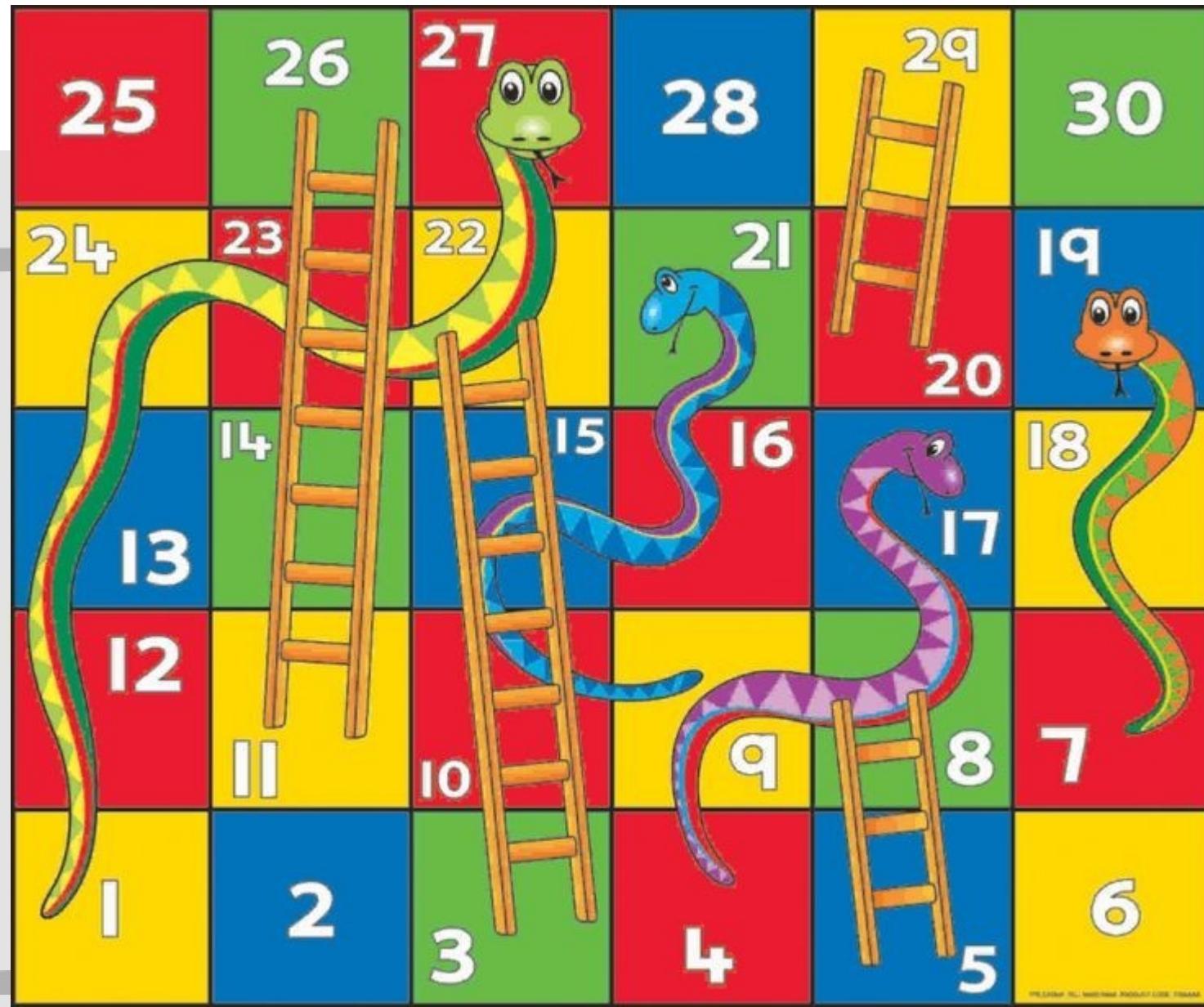


- También se puede programar con librería para Arduino e incluso poner sensores extras

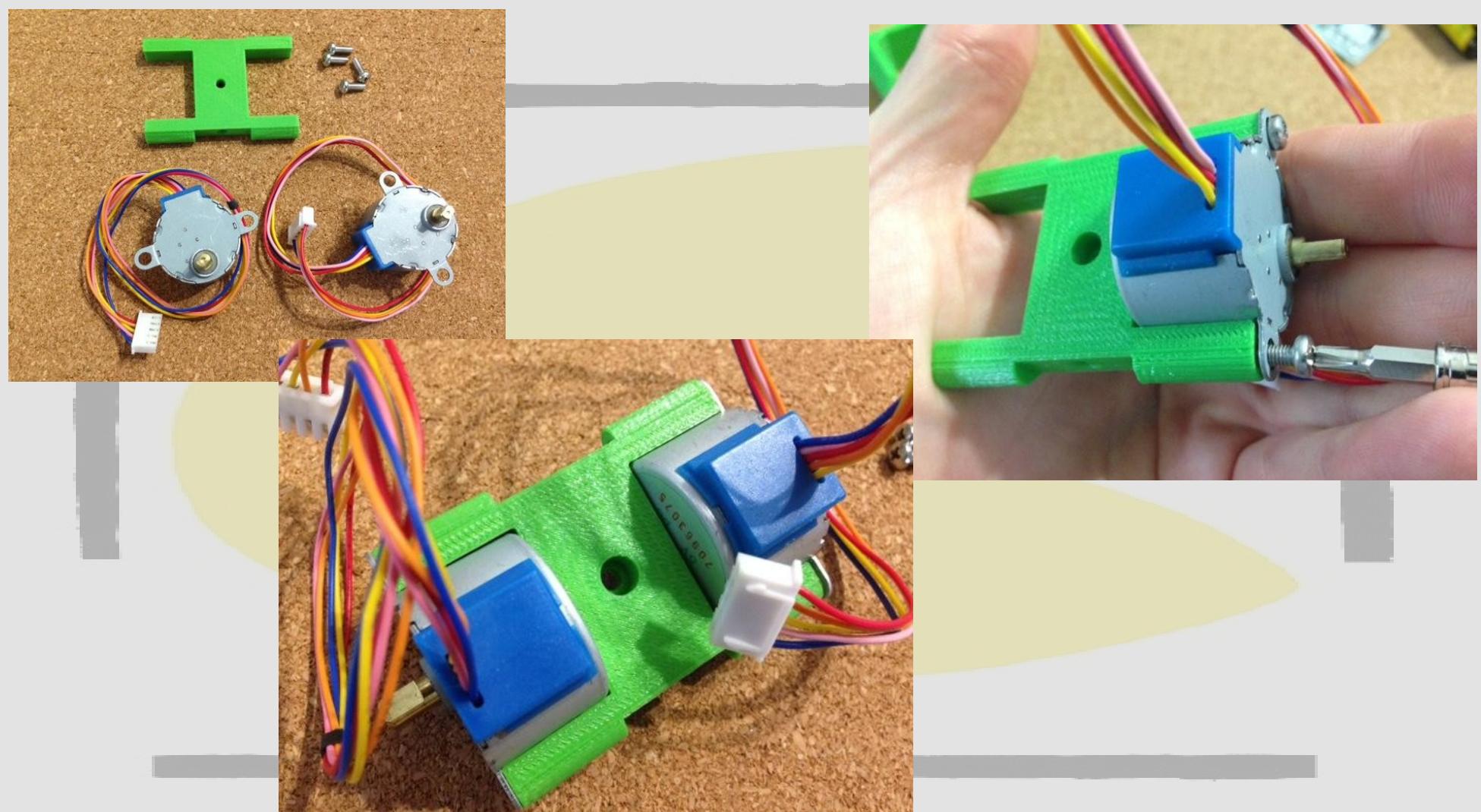


Juegos con Escornabot

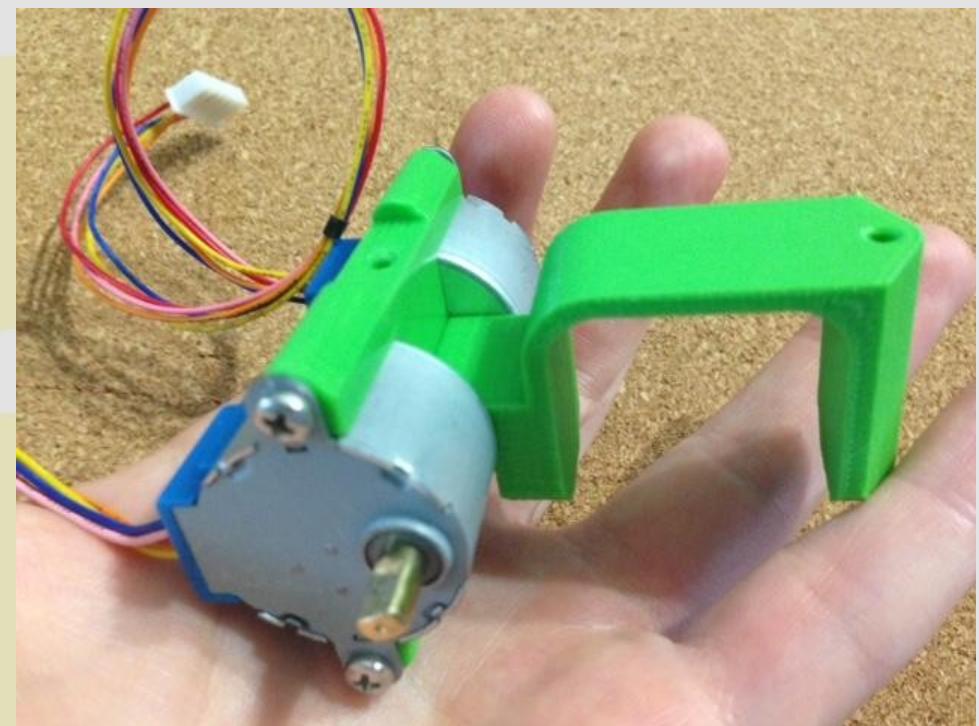




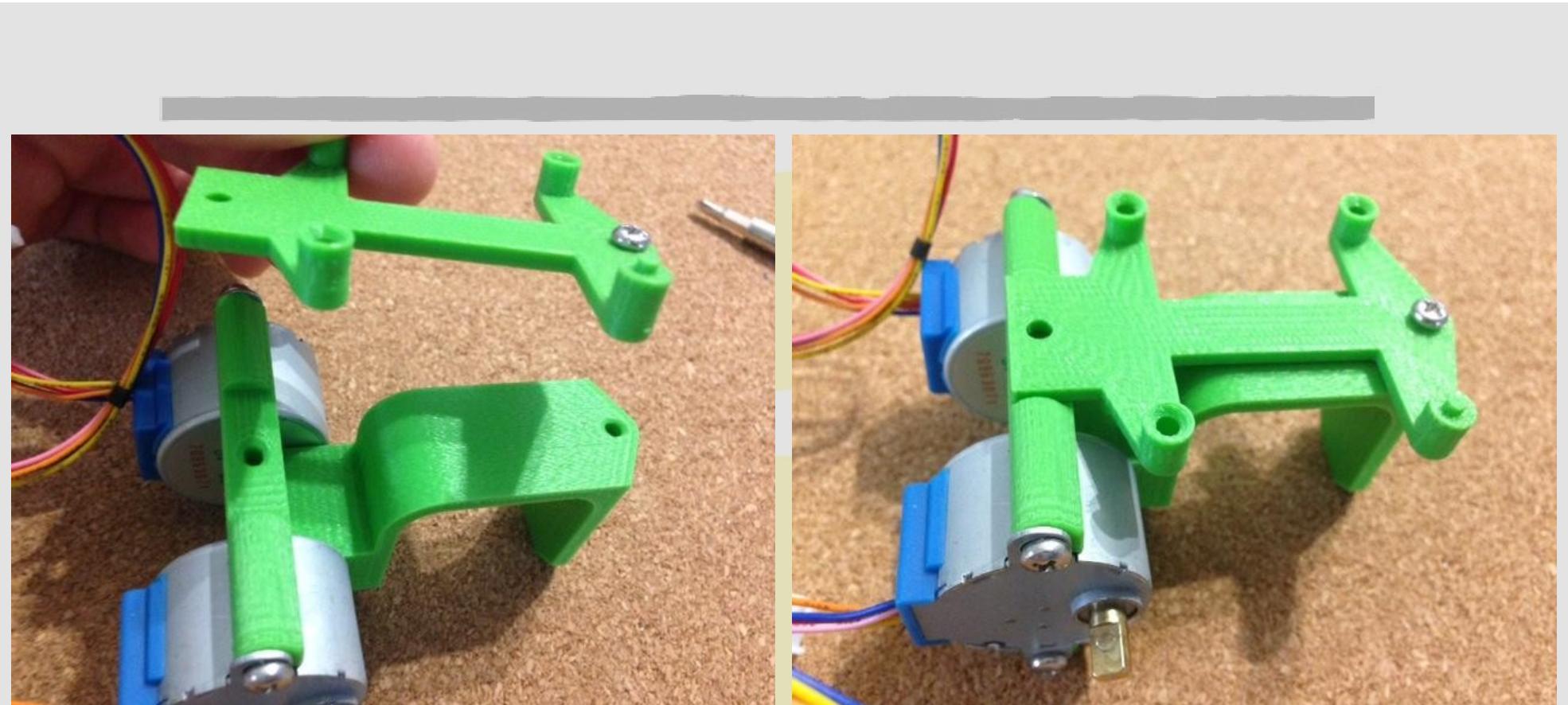
Motores



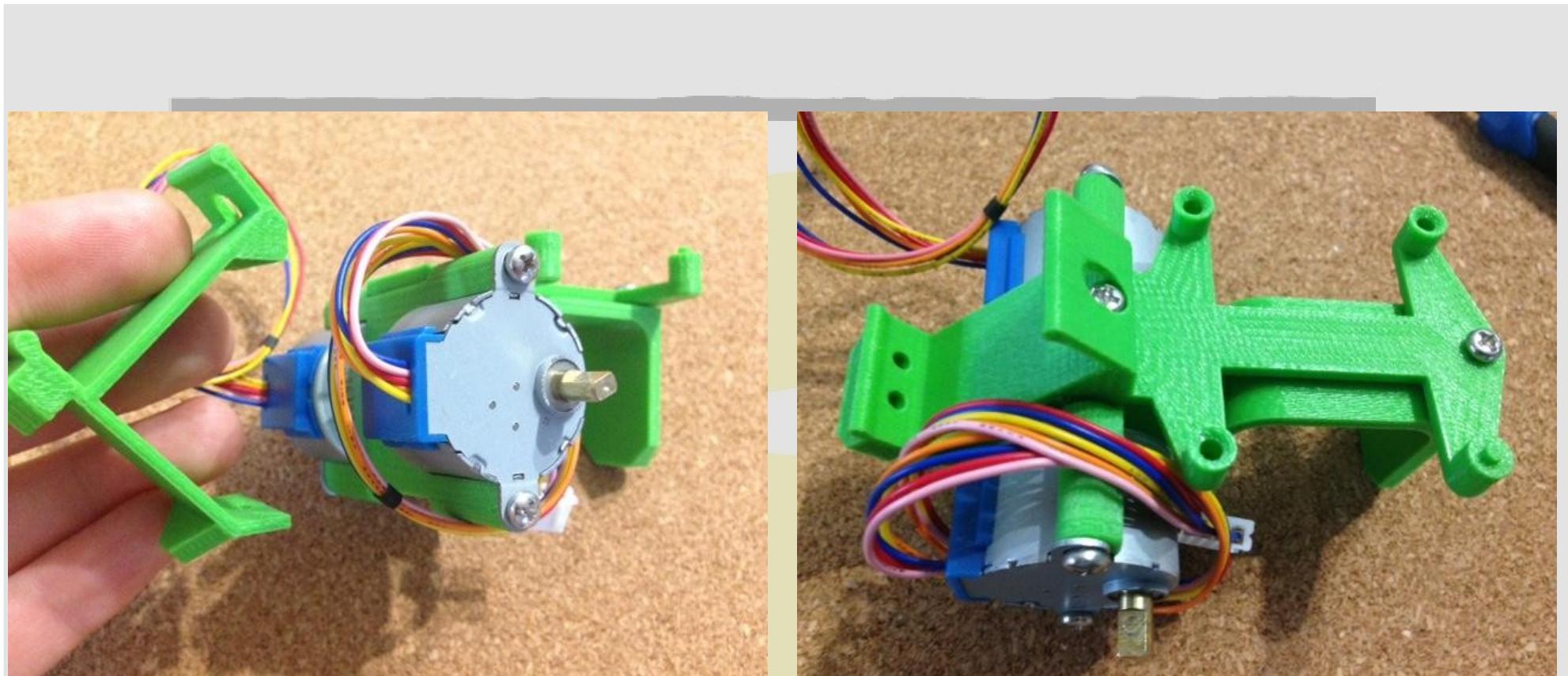
Soporte portapilas



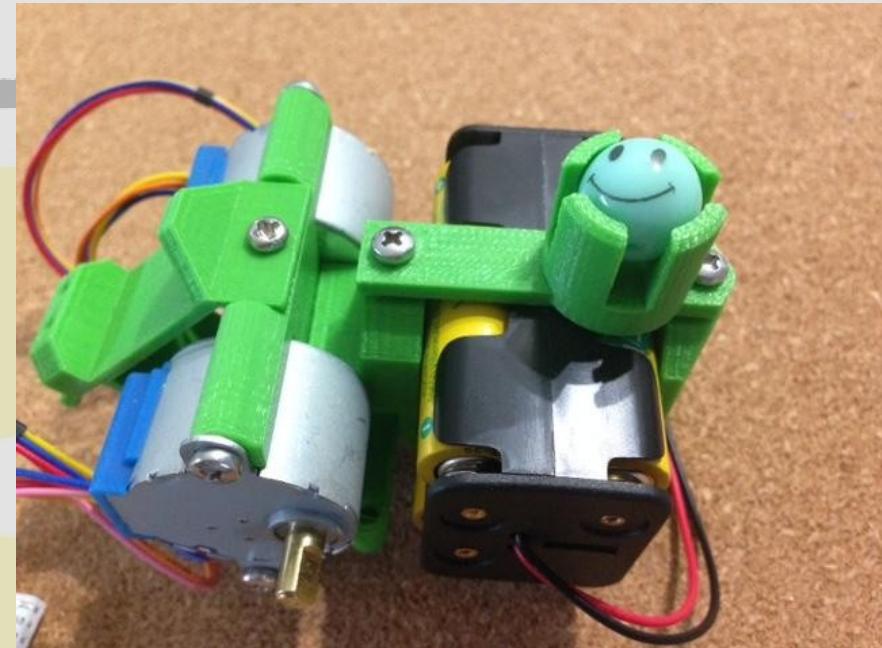
Soporte botonera



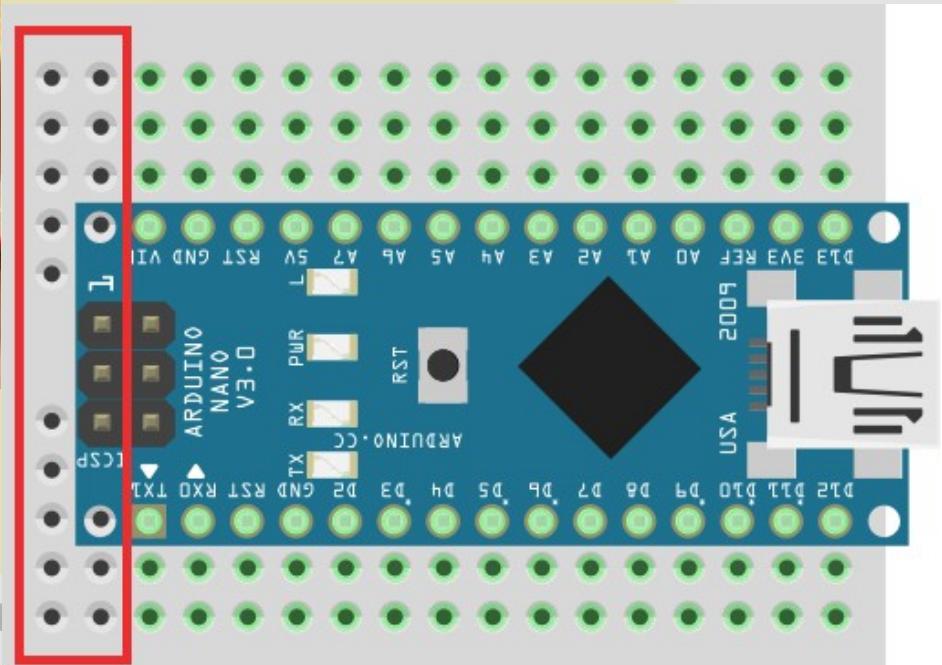
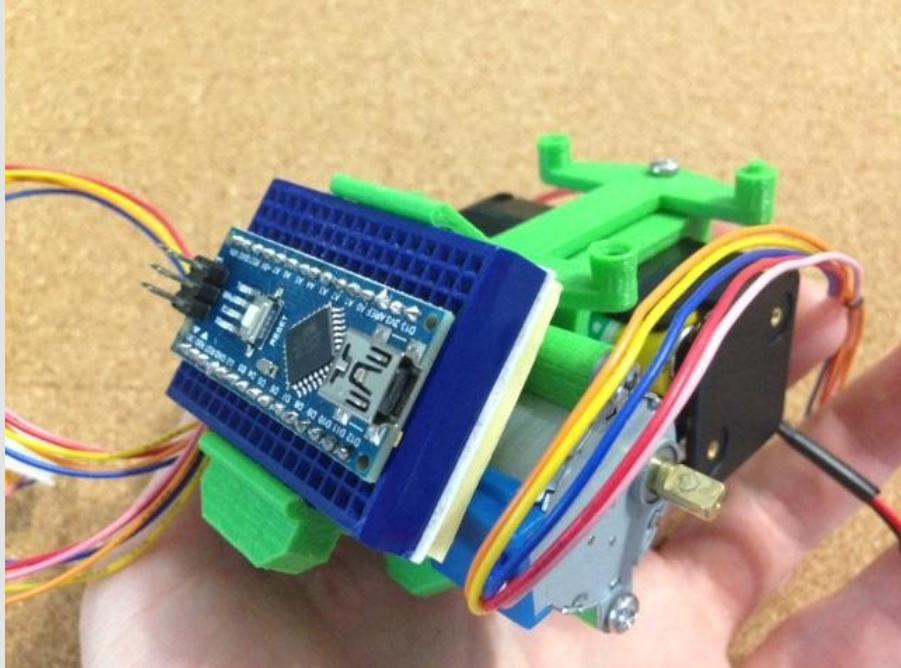
Soporte board



Portapilas y bola



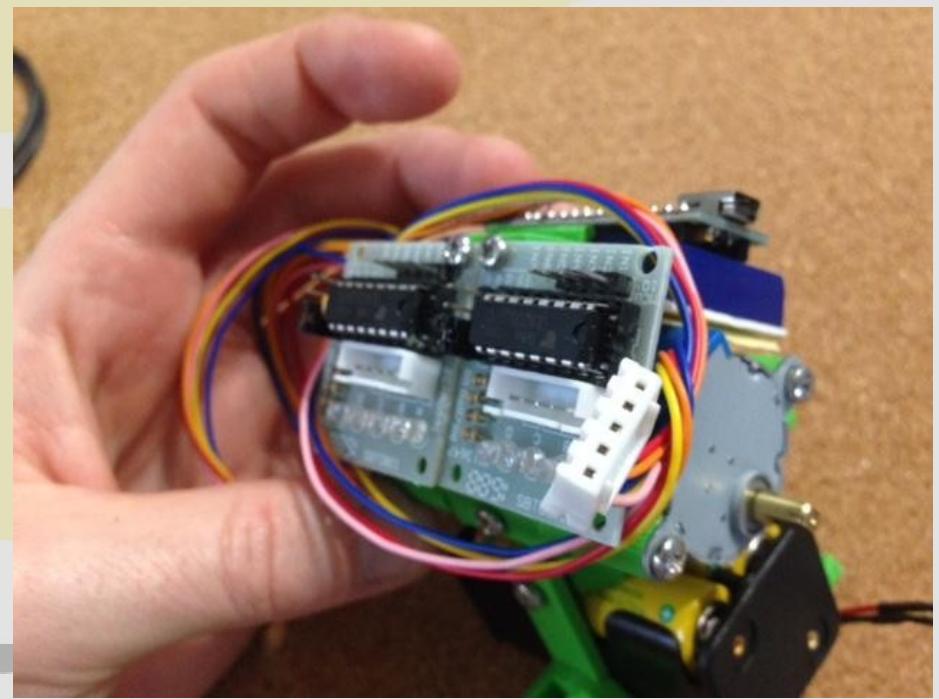
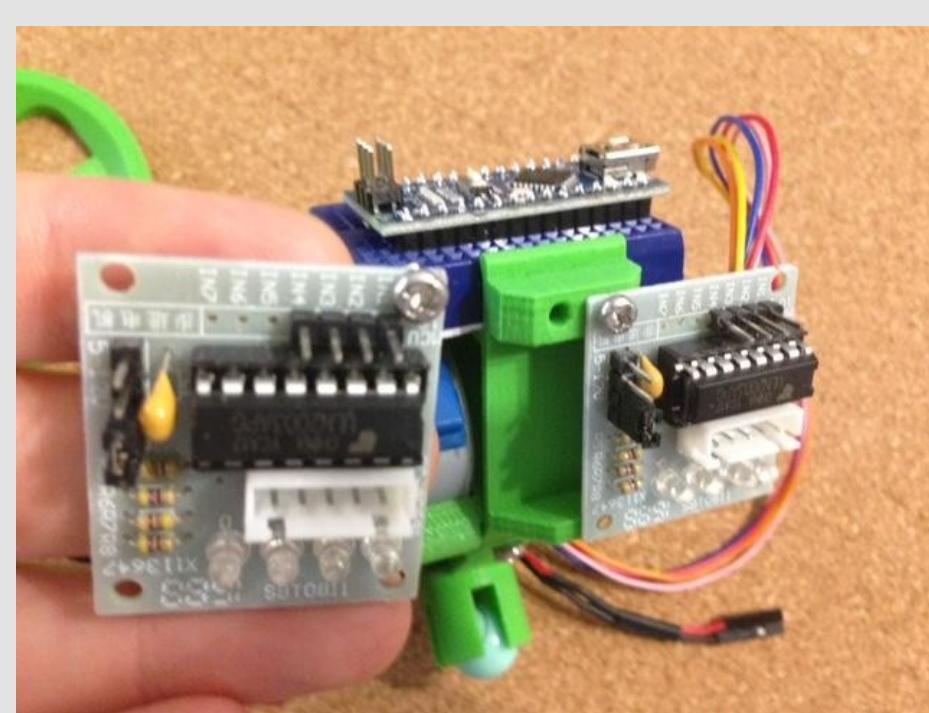
Board



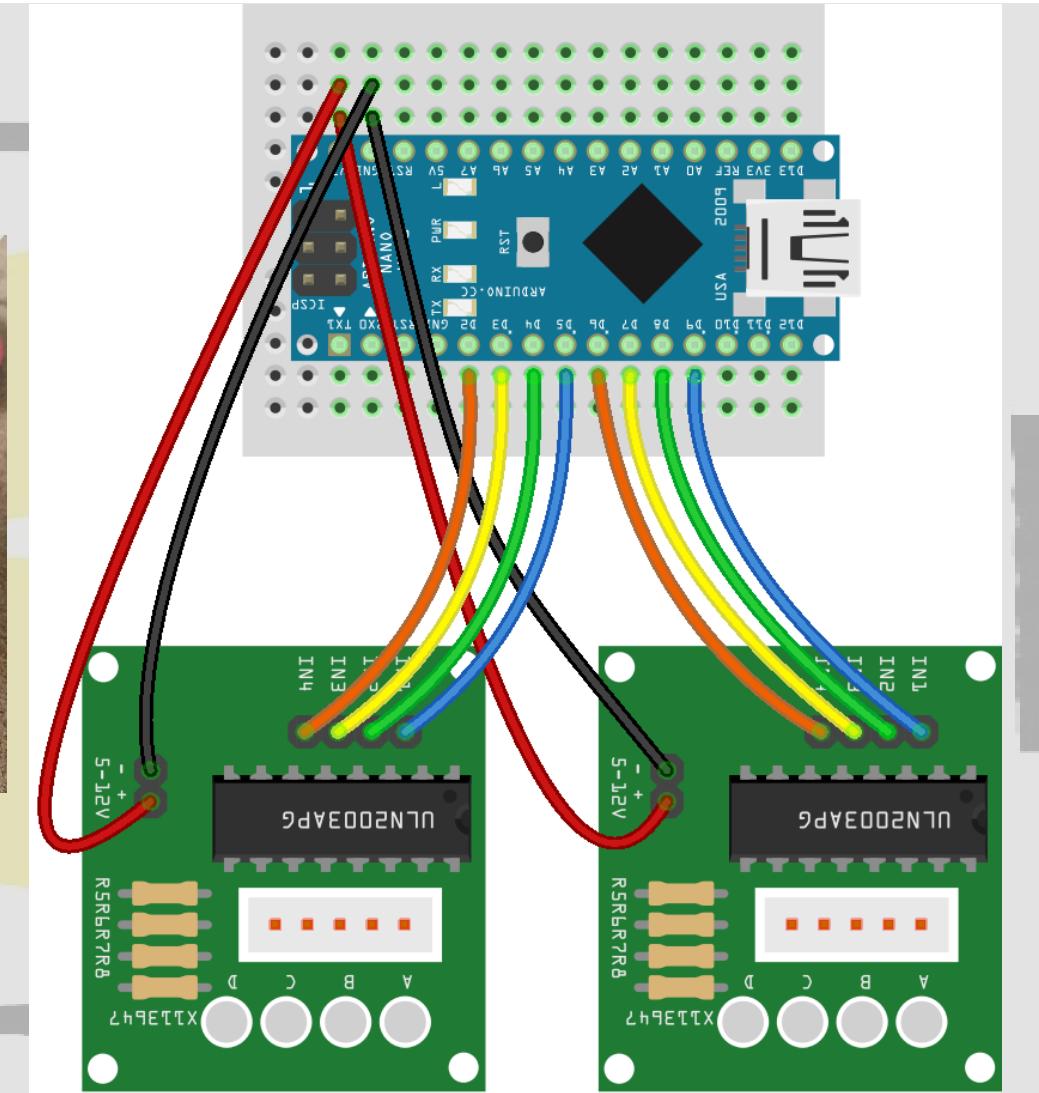
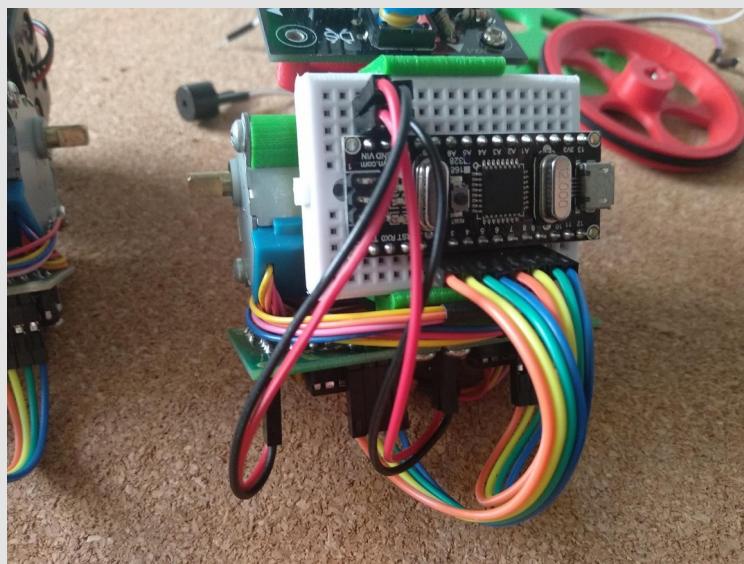
lunes, 11 de noviembre de 2019

fritzing 18

Drivers de motores



Conexionado drivers

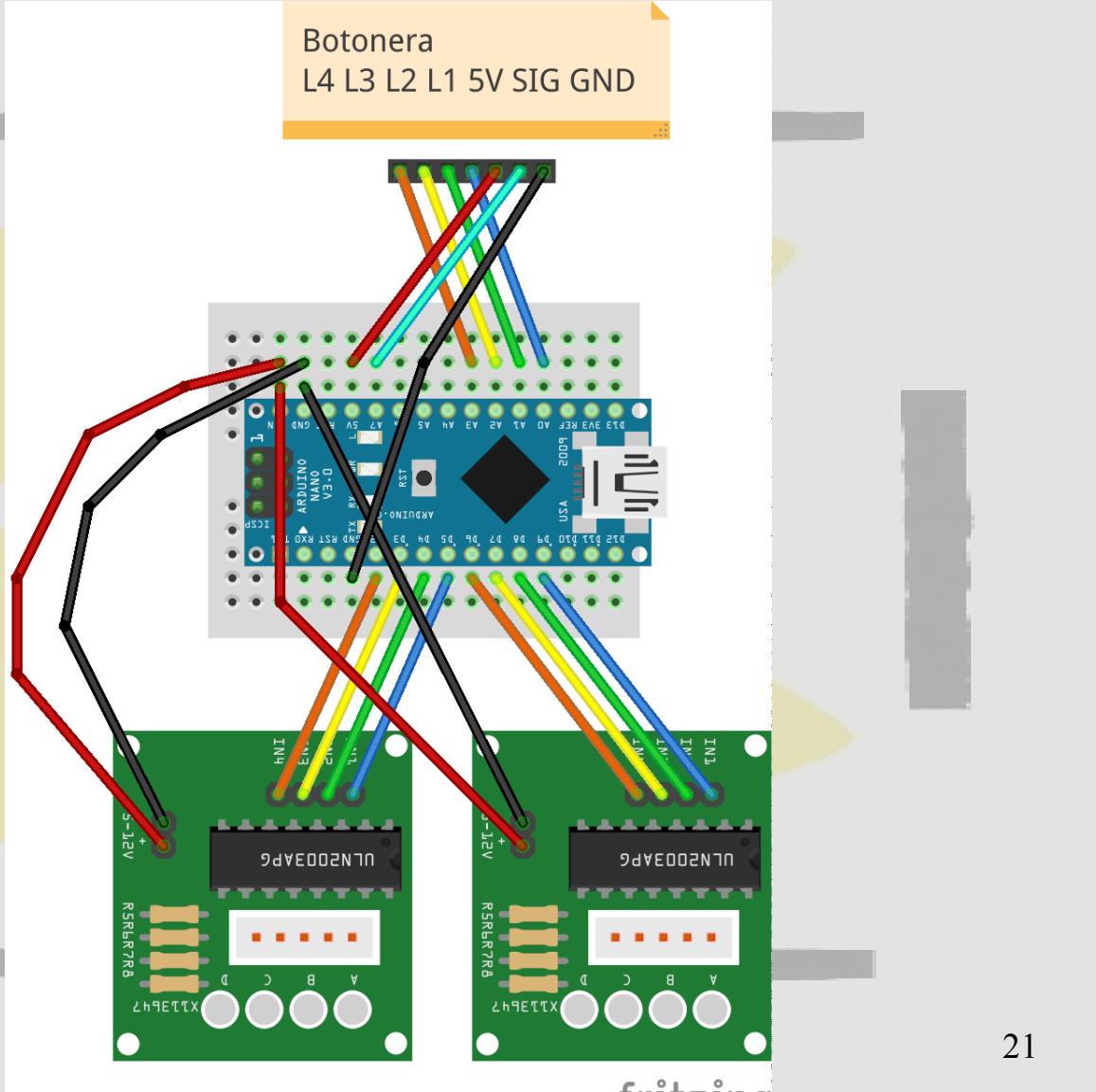
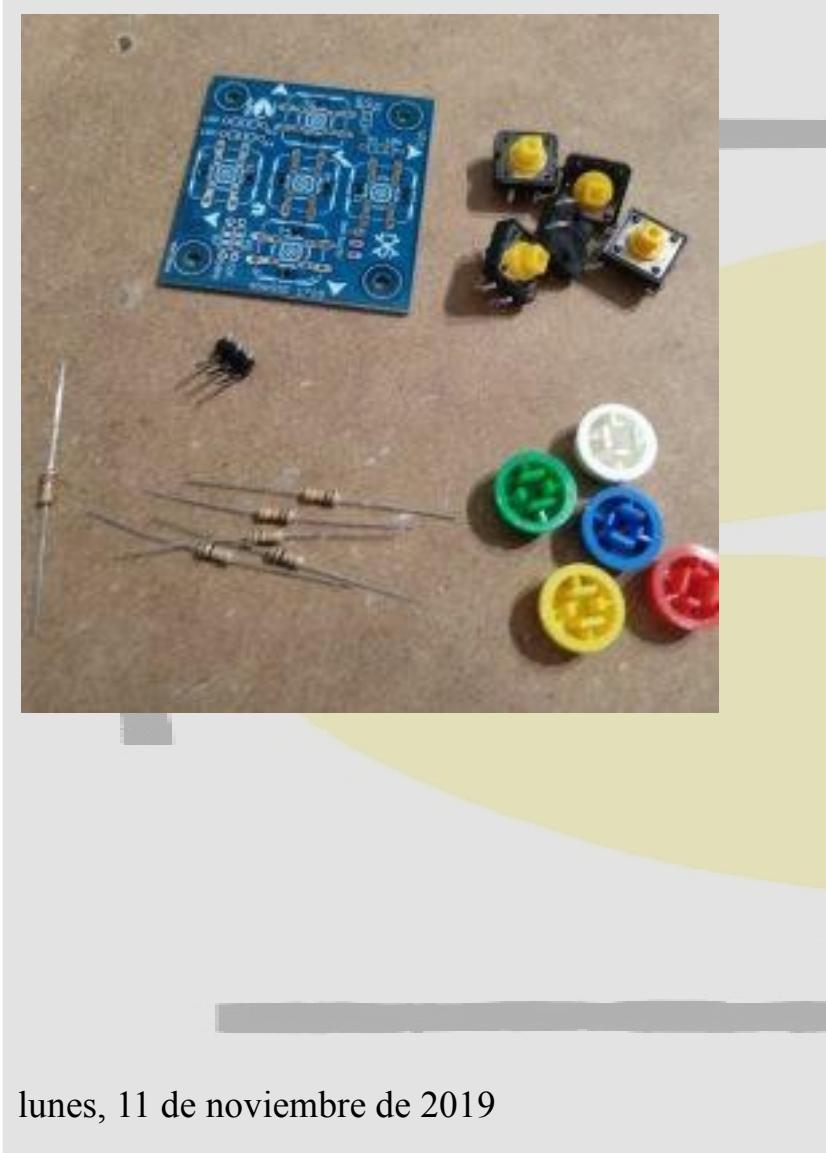


lunes, 11 de noviembre de 2019

fritzing

20

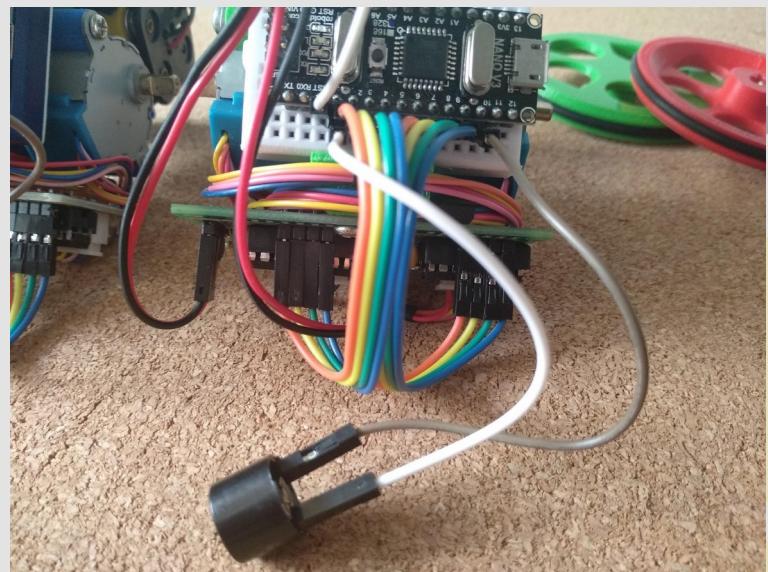
Conexionado Botonera



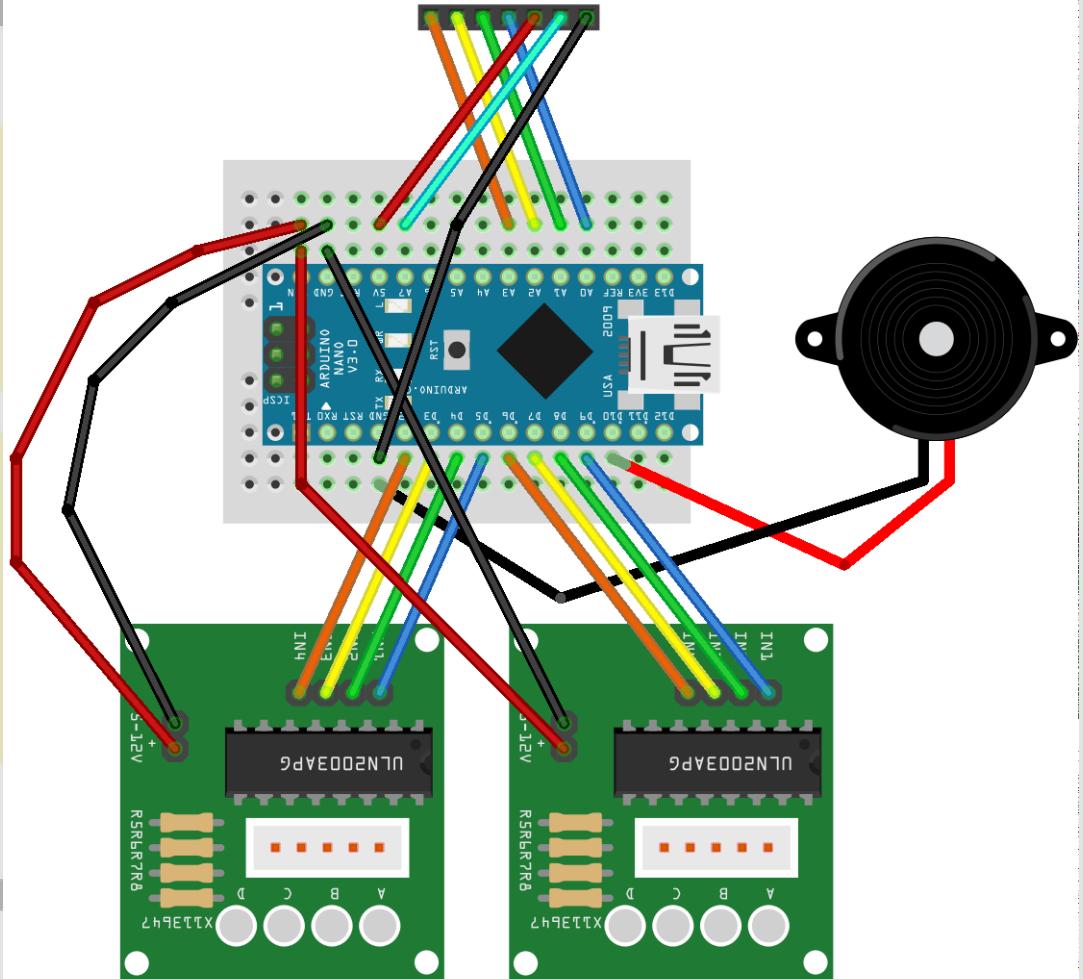
Conexionado botonera

- Pin gnd: gnd de abajo (al lado D2)
- Pin 5V: 5V de arriba
- Pin Signal: A7 (arriba)
- Pin L1: A0
- Pin L2: A1
- Pin L3: A2
- Pin L4: A3

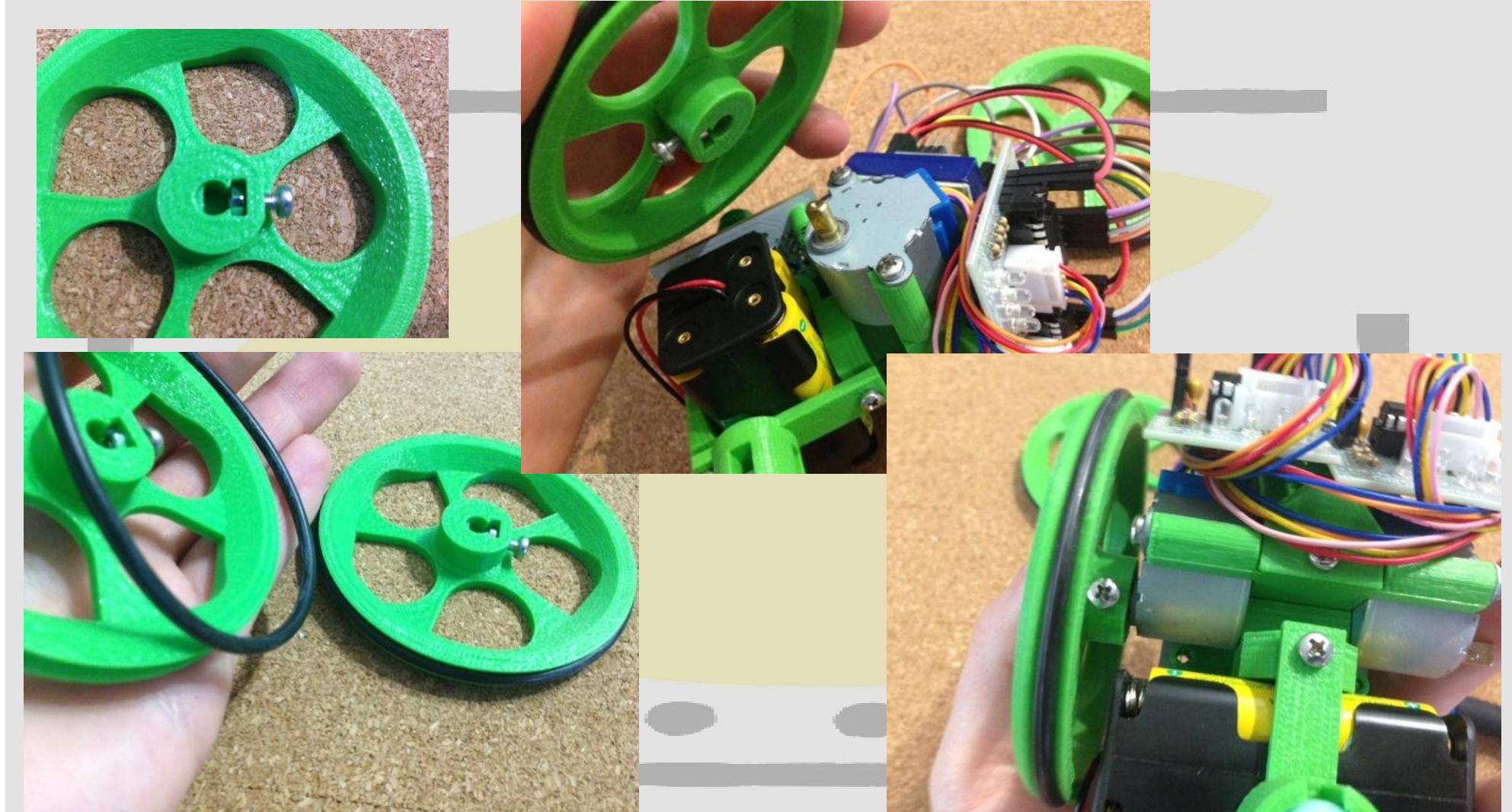
Zumbador y conjunto



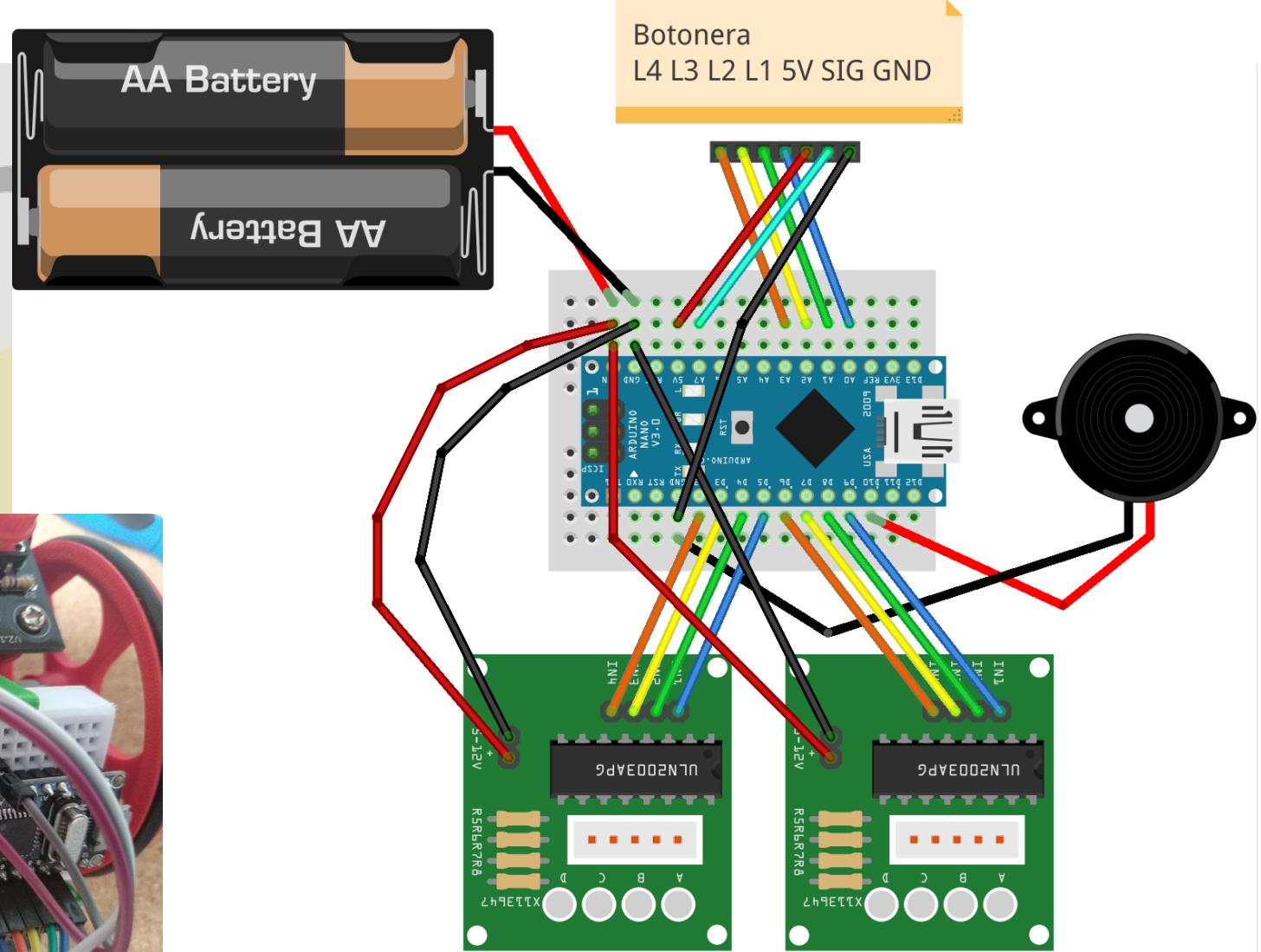
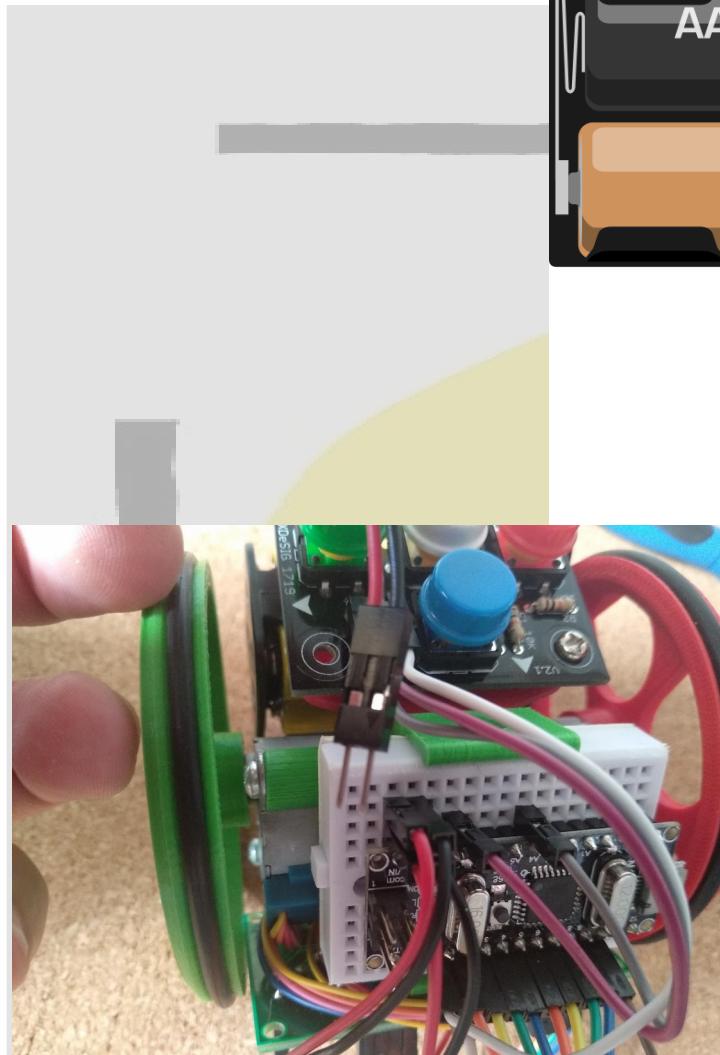
Botonera
L4 L3 L2 L1 5V SIG GND



Ruedas

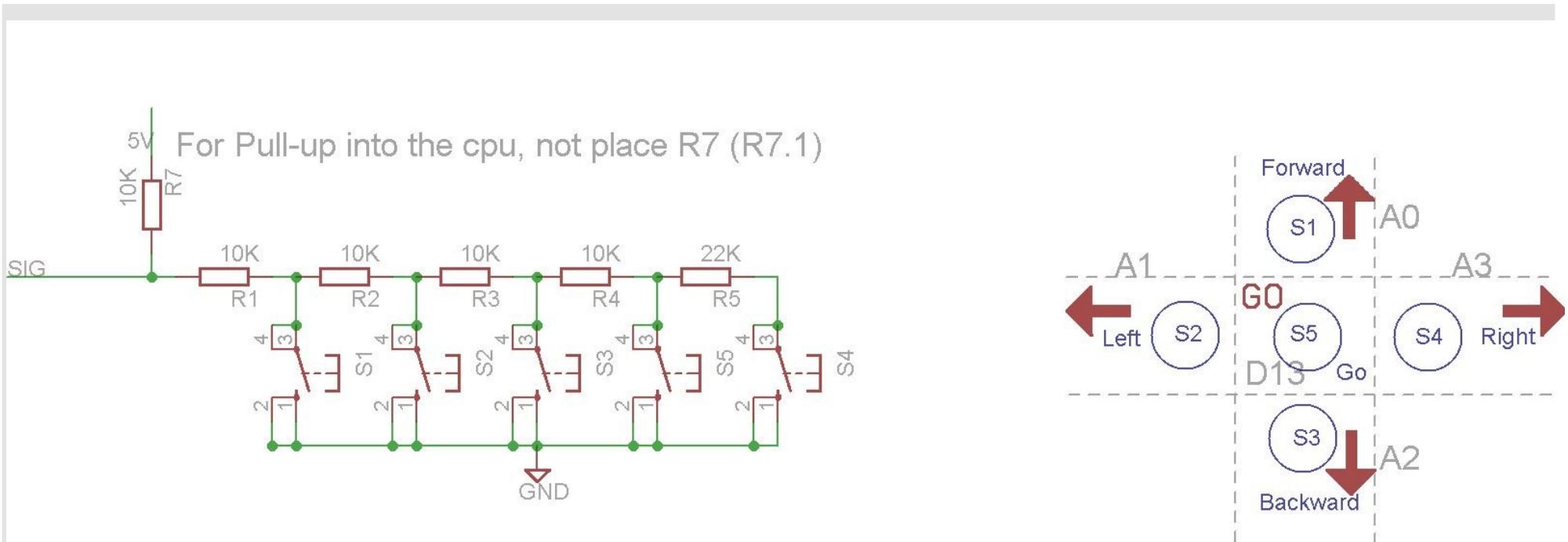


Conexionado Portapilas



lunes, 11 de noviembre de 2019

Botonera



S1 ▲	S2 ◀	S3 ▼	S5 GO	S4 ►
$(10/20^* 1024)= 512$	$(20/30^* 1024)= 683$	$(30/40^* 1024)= 768$	$(40/50^* 1024)= 819$	$(66/76^* 1024)= 889$

Cargar firmware

Firmware: es el código de arduino que hacer funcionar a escornabot.

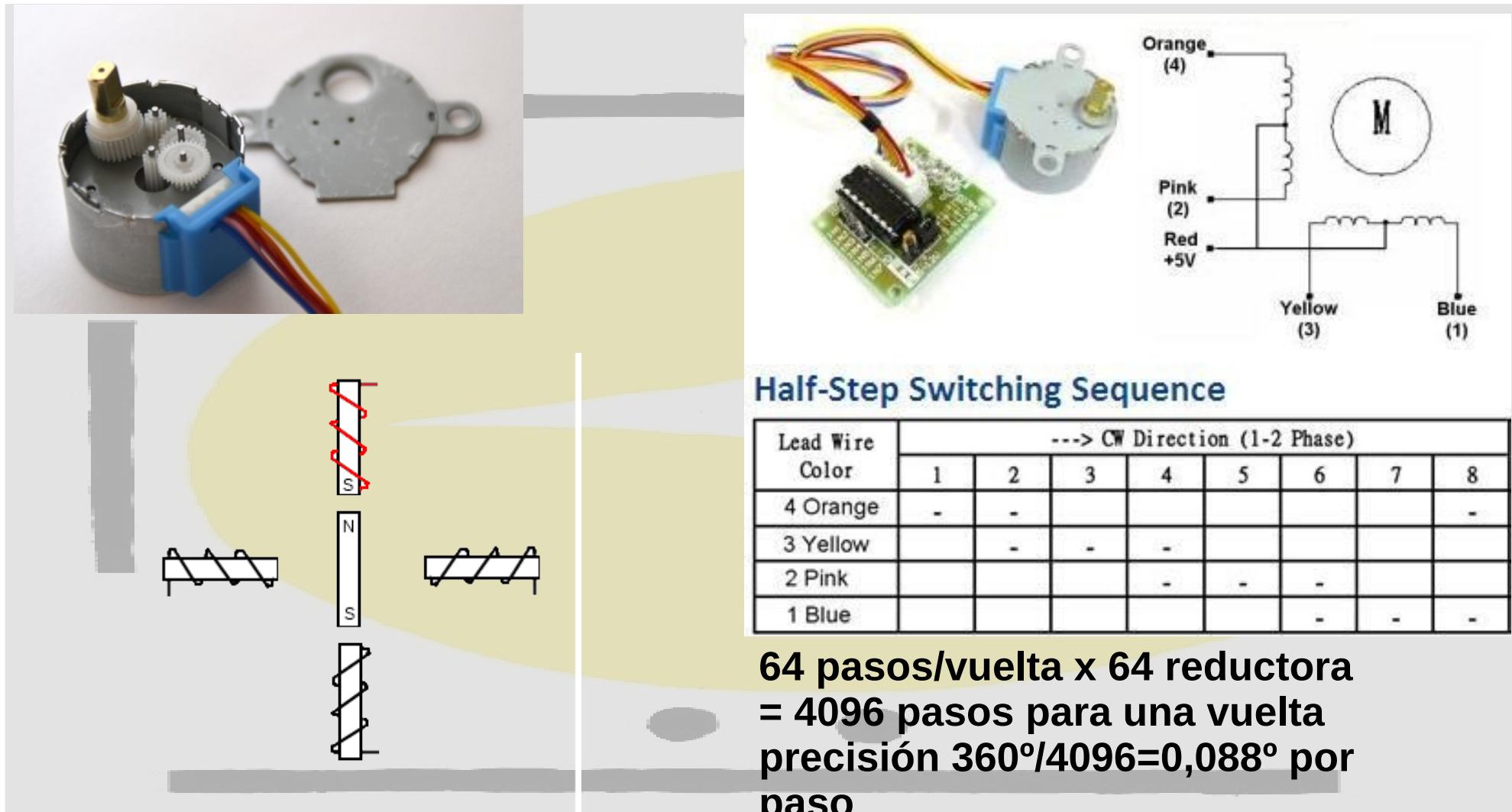
1. Se carga a arduino nano código de programa de botonera, para detectar valor de pulsación de cada botón y se anotan.
2. Se carga a arduino el firmware cambiando previamente en la pestaña “Configuration.h” los valores de pulsación de los diferentes botones.

```
//////////  
//// Button set analog  
/////////  
  
#ifdef BUTTONS_ANALOG  
  
#define BS_ANALOG_WIRES 2  
//#define BS_ANALOG_WIRES 3  
  
// keypad pin setup (analog input)  
#define BS_ANALOG_PIN A7  
  
// input values for each key pressed (0 if key doesn't exist)  
#define BS_ANALOG_VALUE_UP 512  
#define BS_ANALOG_VALUE_RIGHT 860  
#define BS_ANALOG_VALUE_DOWN 769  
#define BS_ANALOG_VALUE_LEFT 683  
#define BS_ANALOG_VALUE_GO 810  
#define BS_ANALOG_VALUE_RESET 0  
  
#endif // BUTTONS_ANALOG
```

Modos firmware 1.6.2

- **Modo normal**
 - Pulsación corta: giros 90°
 - Pulsación larga: giros 45°
- **Modo 60° (tecla GO pulsación larga)**
 - Pulsación corta: giros 60°
 - Pulsación larga: giros 120°
- **Pausa:** pulsación larga tecla atrás.

Motor paso a paso



lunes, 11 de noviembre de 2019

Cambios en firmware

- Abrimos **Escornabot.ino**, pestaña **Configuration.h**
 - **#define STEPPERS_STEPS_PER_SECOND 1000**
Número de pasos por segundo, el tope anda sobre 2300
 - **#define STEPPERS_LINE_STEPS 1738**
Da un avance de 10cm, AVANCE 1 vuelta= $2\pi \times R$
 - **#define STEPPERS_TURN_STEPS 1024**
Establece un giro de 90°,
Una vuelta completa 4096 pasos

```
#ifdef ENGINE_TYPE_STEPPERS

    // stepper pin setup (digital outputs)
#define STEPPERS_MOTOR_RIGHT_IN1 5
#define STEPPERS_MOTOR_RIGHT_IN2 4
#define STEPPERS_MOTOR_RIGHT_IN3 3
#define STEPPERS_MOTOR_RIGHT_IN4 2
#define STEPPERS_MOTOR_LEFT_IN1 9
#define STEPPERS_MOTOR_LEFT_IN2 8
#define STEPPERS_MOTOR_LEFT_IN3 7
#define STEPPERS_MOTOR_LEFT_IN4 6

    // step calibration
#define STEPPERS_STEPS_PER_SECOND 1000
#define STEPPERS_LINE_STEPS 1738
#define STEPPERS_TURN_STEPS 1024

#endif
```