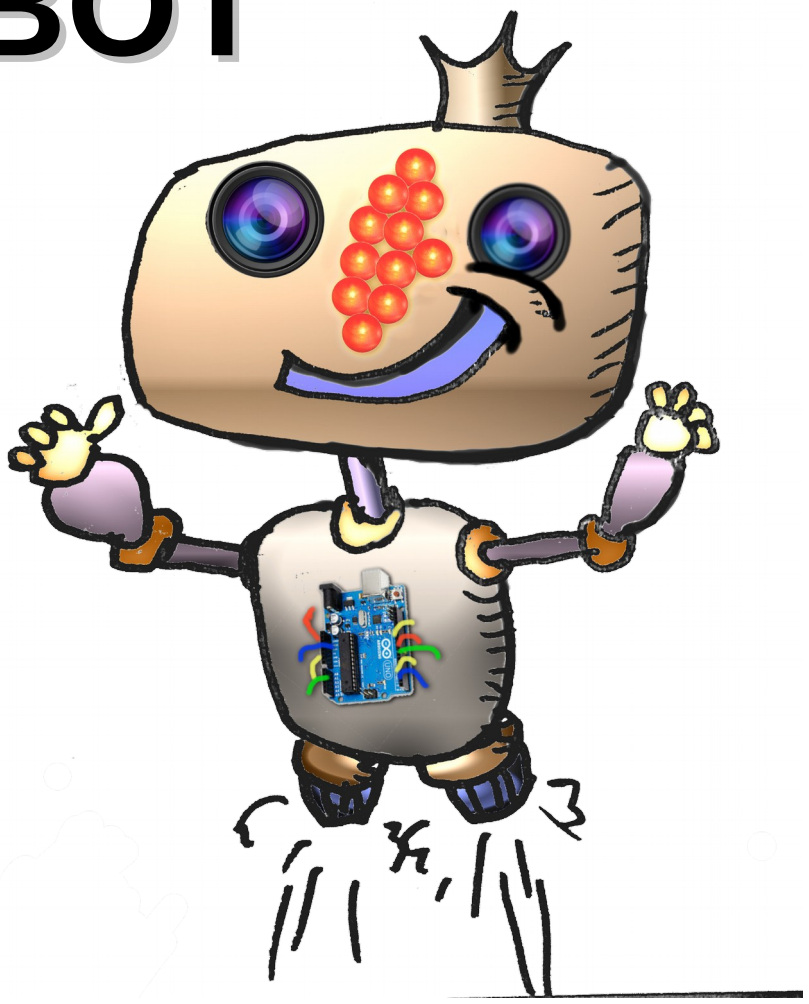
TALLER DE ESCORNABOT

Club de Robótica de Granada



https://clubroboticagranada.github.io/

RED ANDALUZA DE ROBÓTICA Y TECNOLOGÍA EDUCATIVA

APTA

PARQUE de las CIENCIAS
ANDALUCÍA - GRANADA

@clubroboticagra

Pedro Ruíz @pedroruizf

Ángela Pozo @Ankgiel

¿Escorna... qué?

ESCORNABOI:

escarabajo

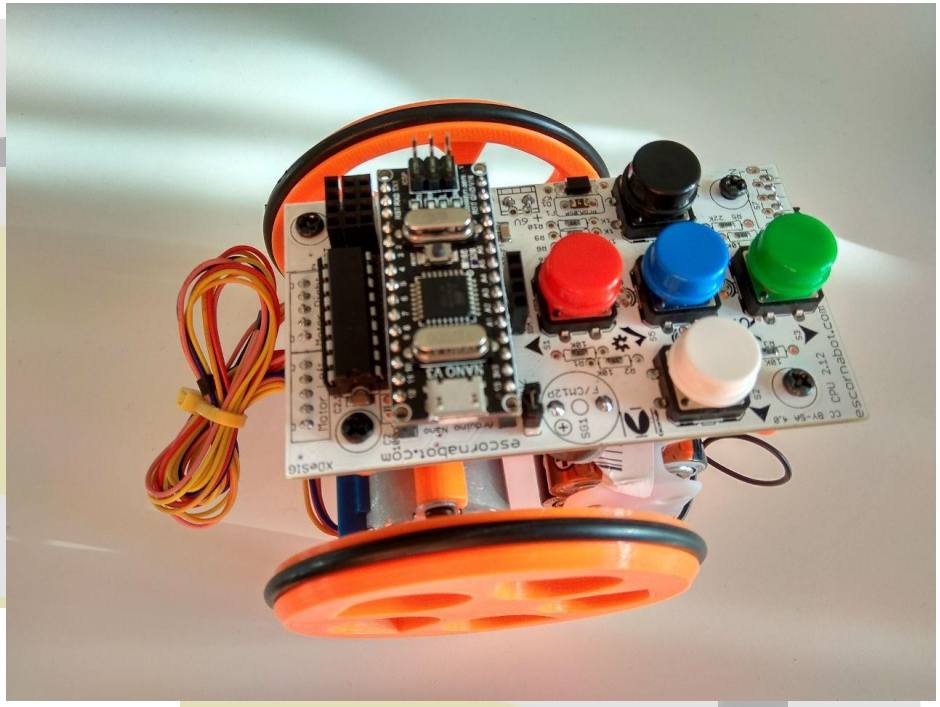
lucanus cervus

en gallego

+

BOT:

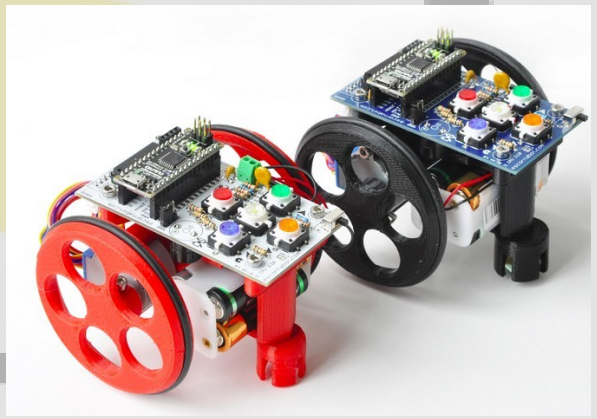
robot



Objetivo

ROBÓTICA Y PROGRAMACIÓN

Sustituye a robots privativos



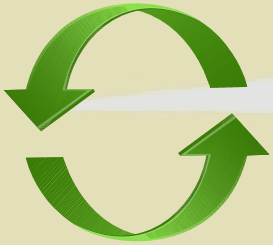
Características

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

¿Quién?

Proyecto liderado y soportado por la comunidad:

Profesores, diseñadores, desarrolladores, traductores...



Comunidad: Grupo de Google / Telegram

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

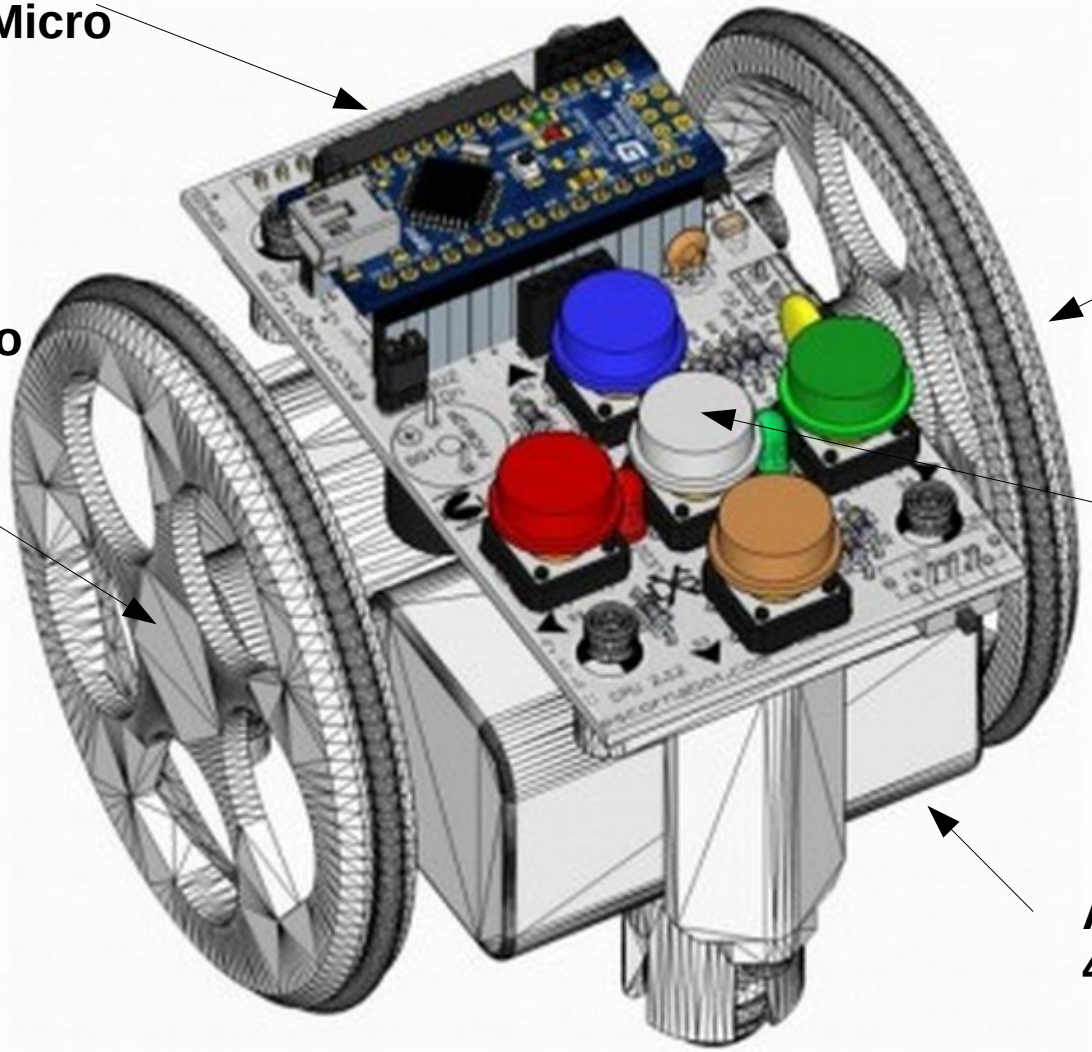
¡¡¡ Escornafan !!!

Pablo Rubio

https://pablorubma.cc/

Escornabot

Sistema de Control
(Programación)
Arduino Nano o Micro



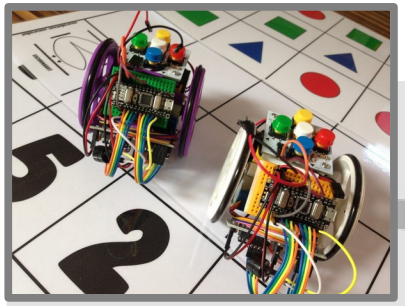
Actuador 1
Motor Paso a Paso

Actuador 2
Motor Paso a Paso

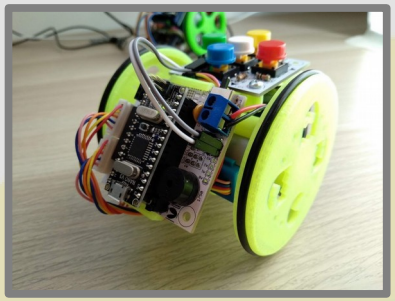
Sensores
5 botones

Alimentación
4 pilas

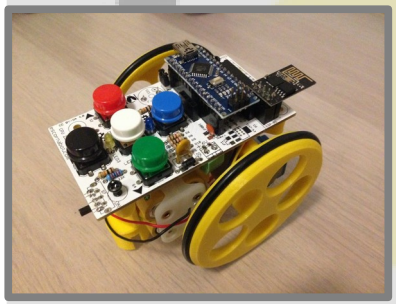
Versiones



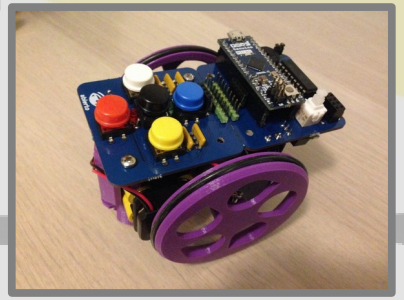
DO IT YOURSELF



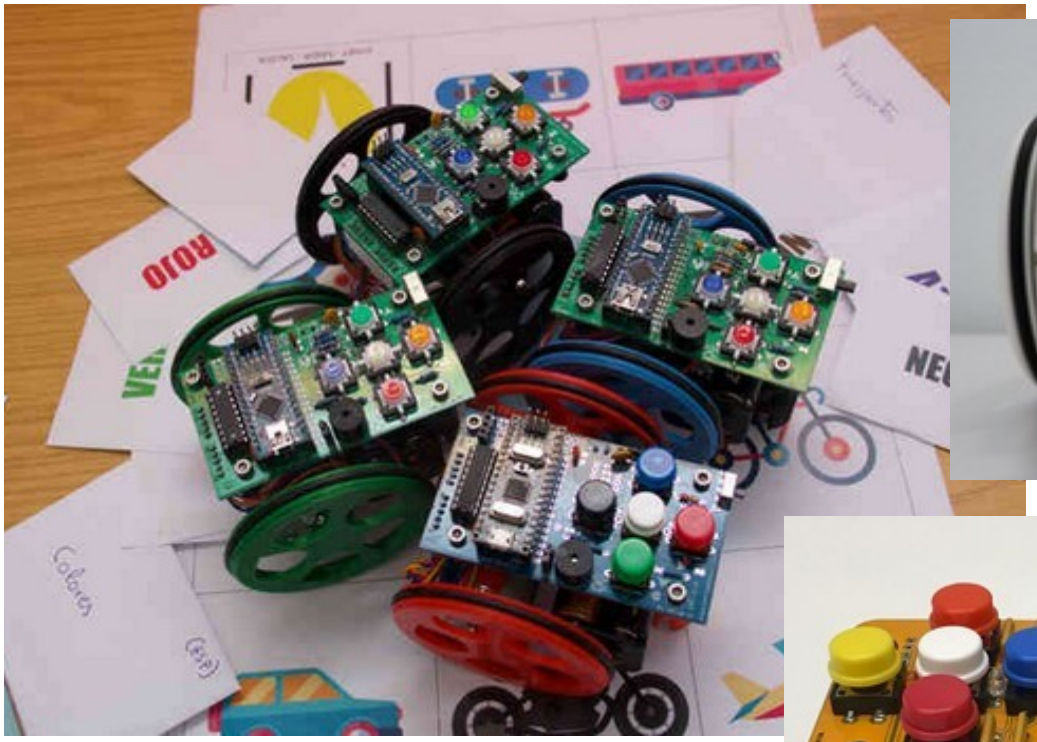
COMPACTUS

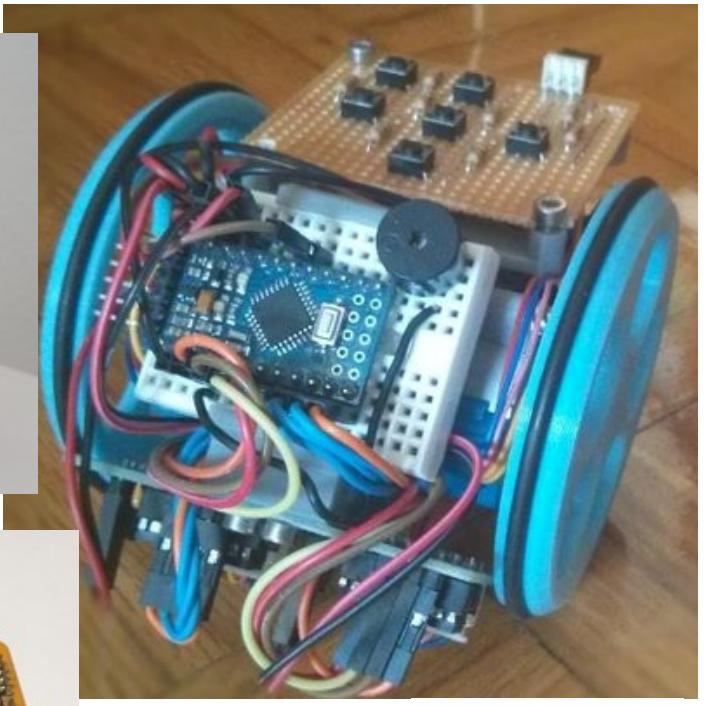


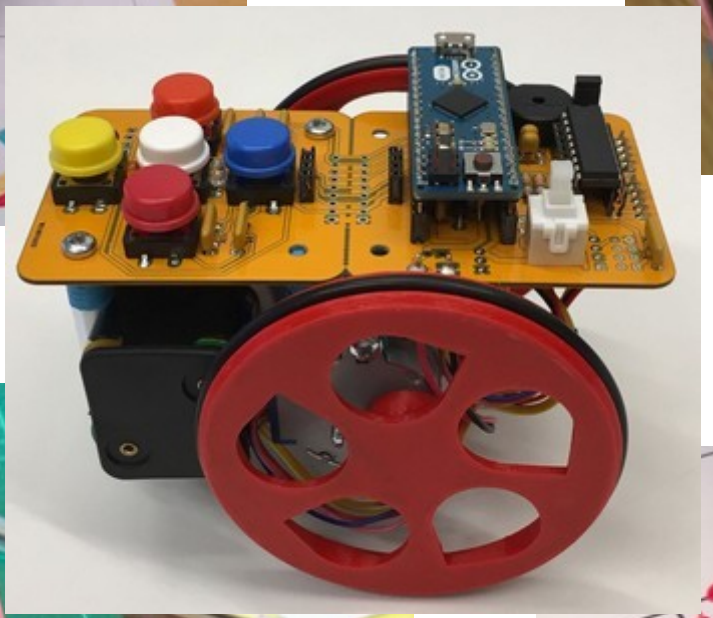
PLACA 2.12

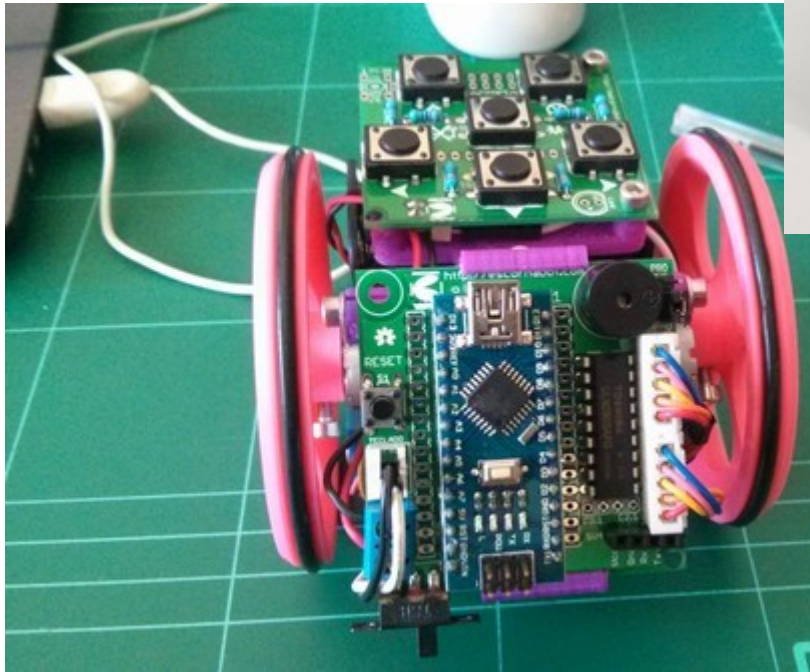


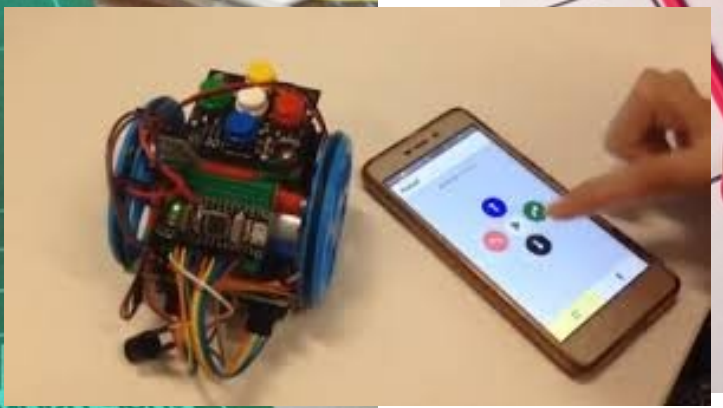
OKAGI

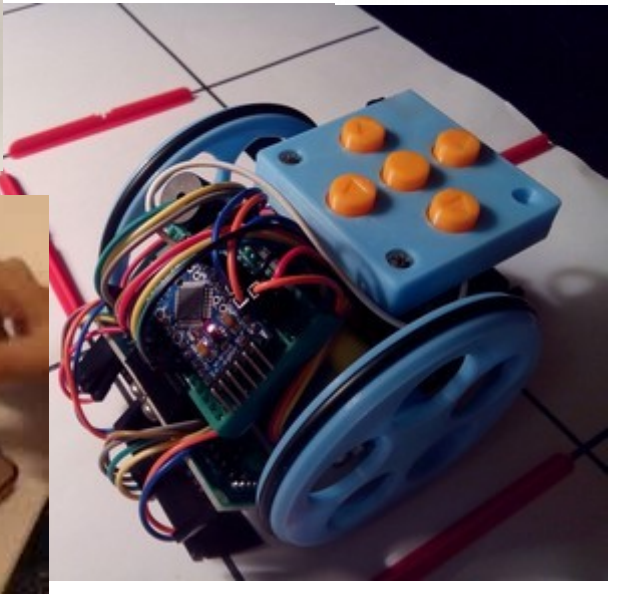






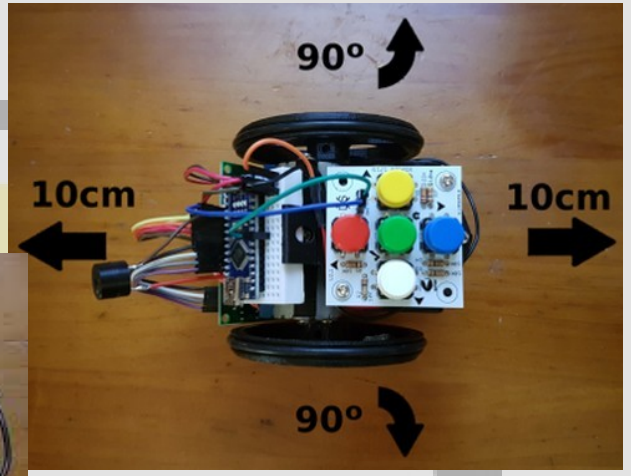




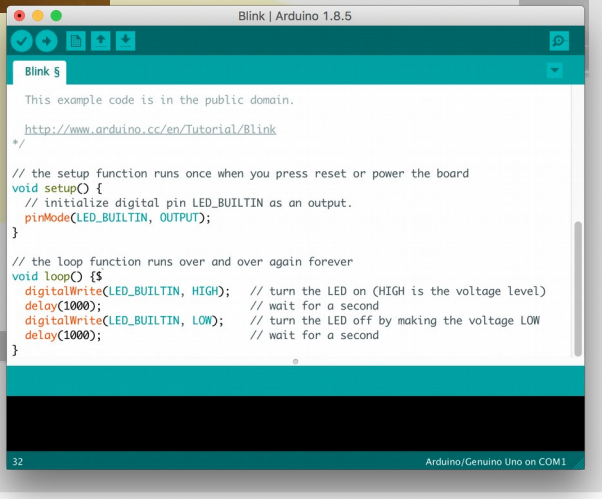


Funcionamiento y programación

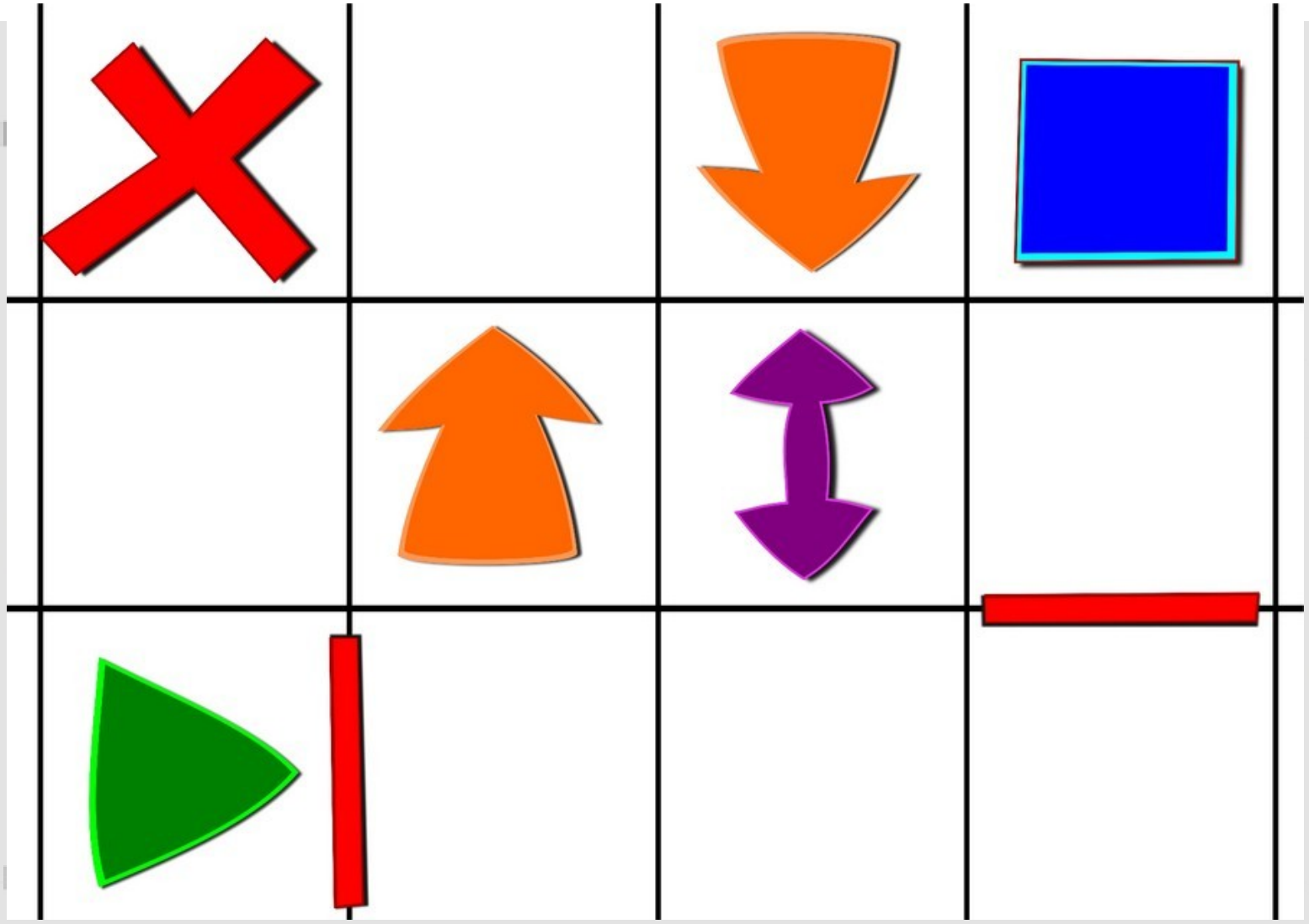
- Introduce firmware preparado y se maneja con botonera (modo clásico)



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

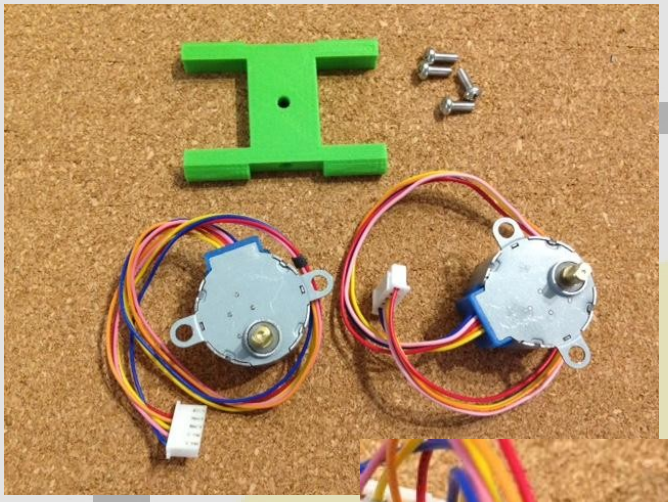


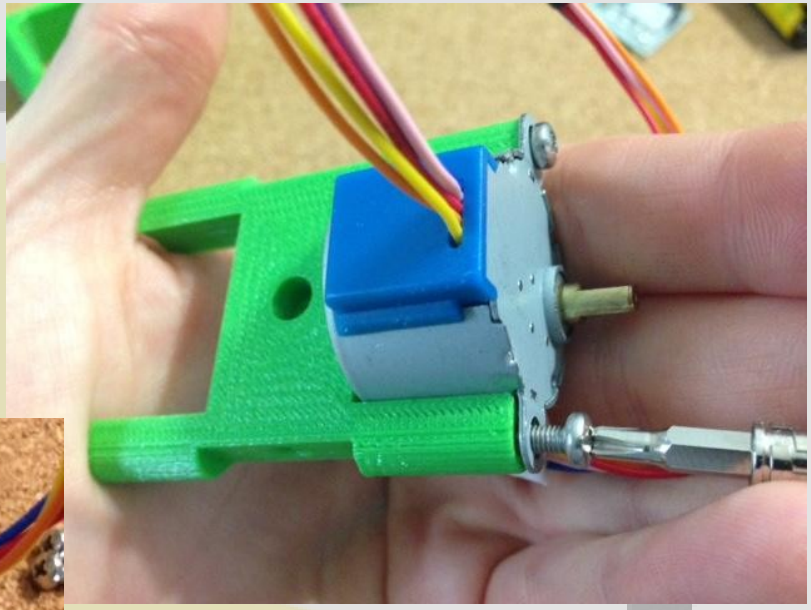
Juegos con Escornabot

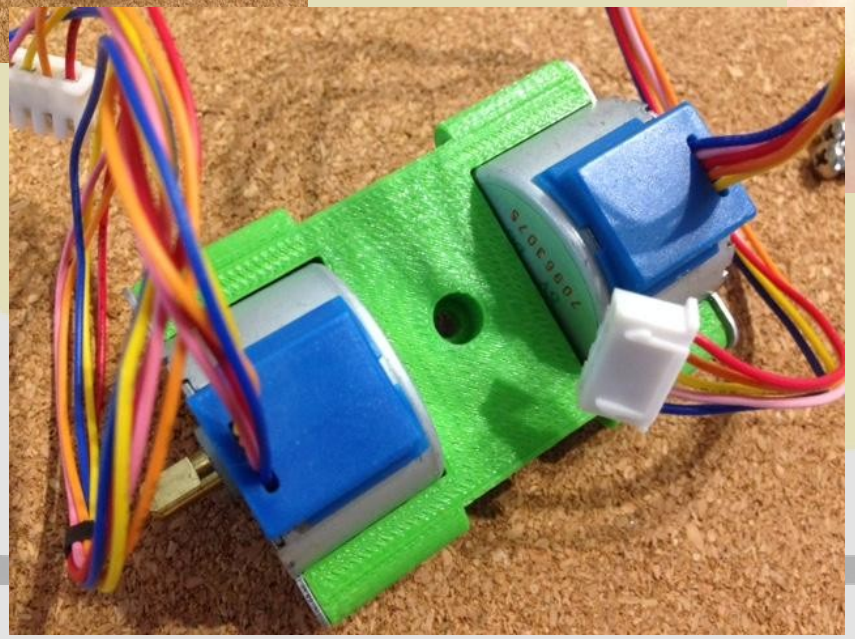



25
26
27
28
29
30
24
23
22
21
20
19
14
15
16
17
18
13
12
11
10
9
8
7
1
2
3
4
5
6

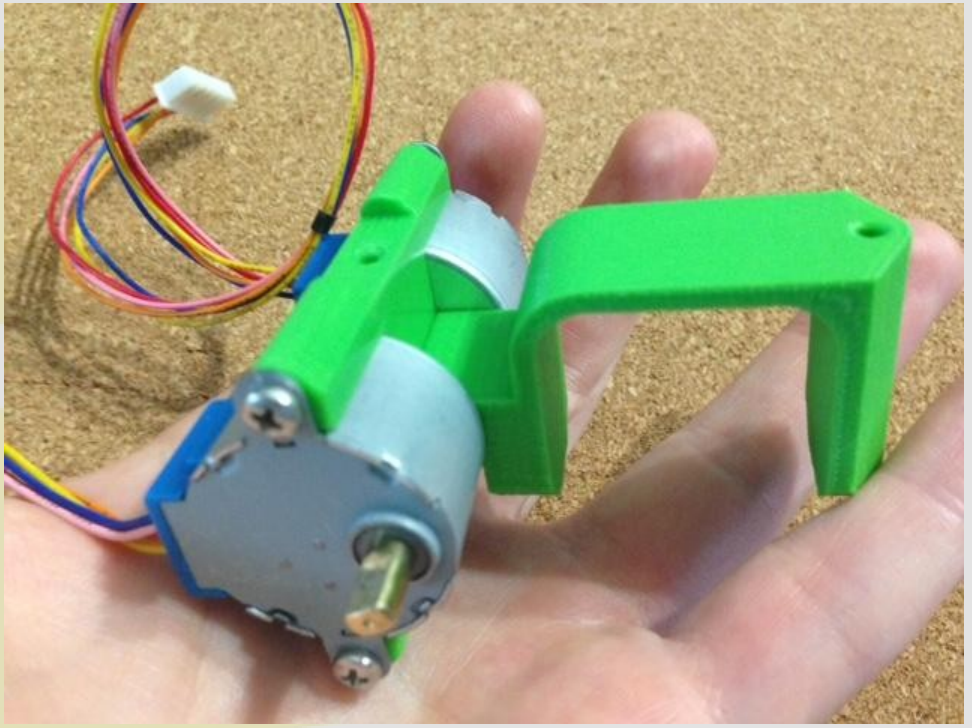
Motores



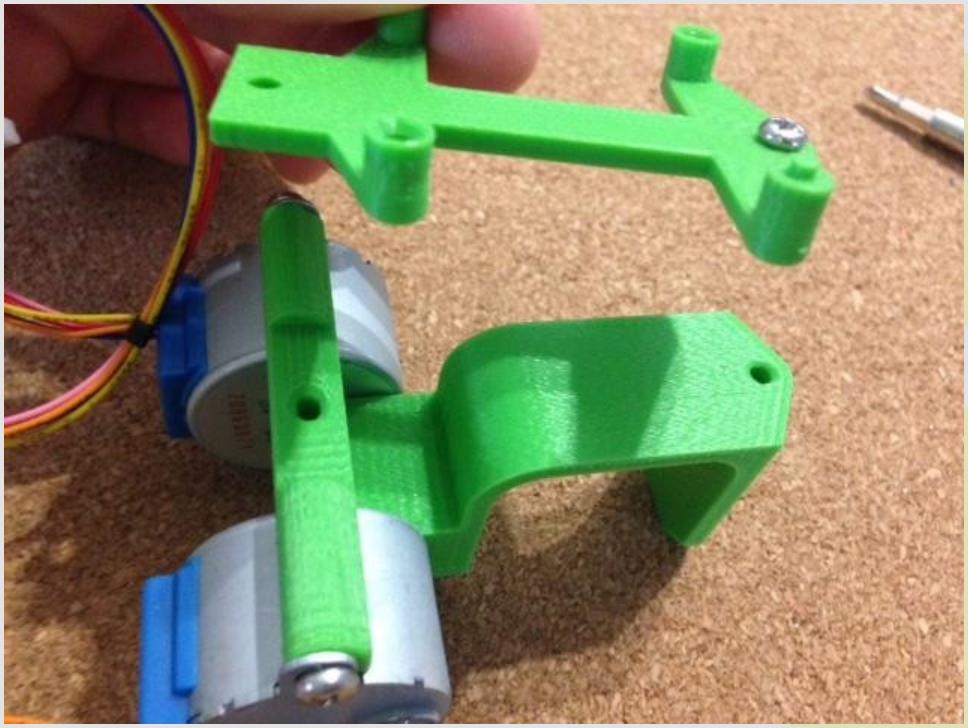


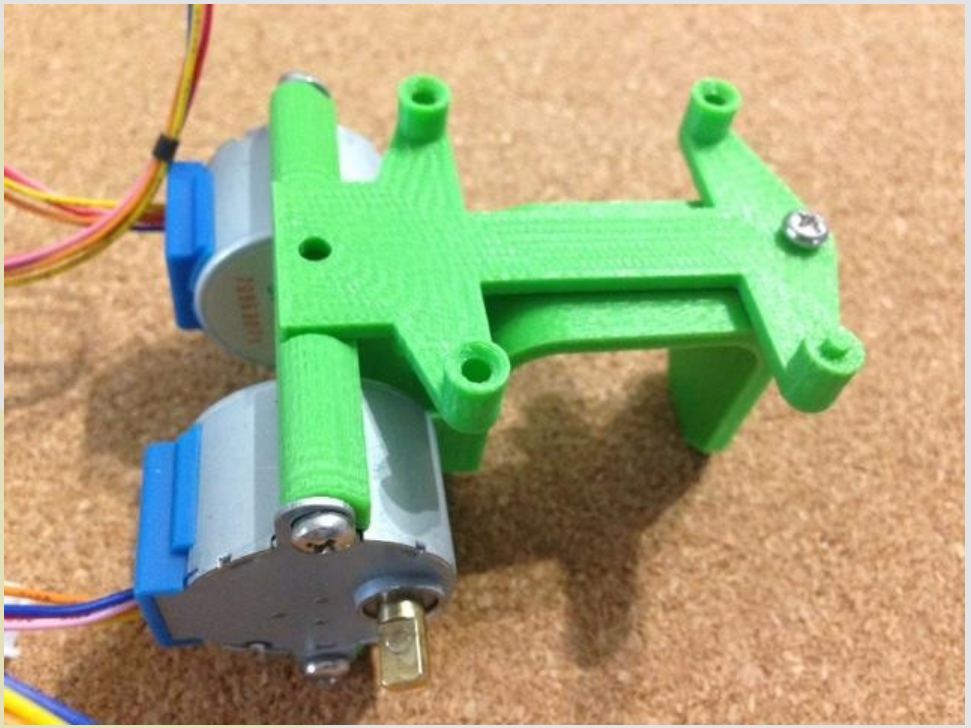


Soporte portapilas

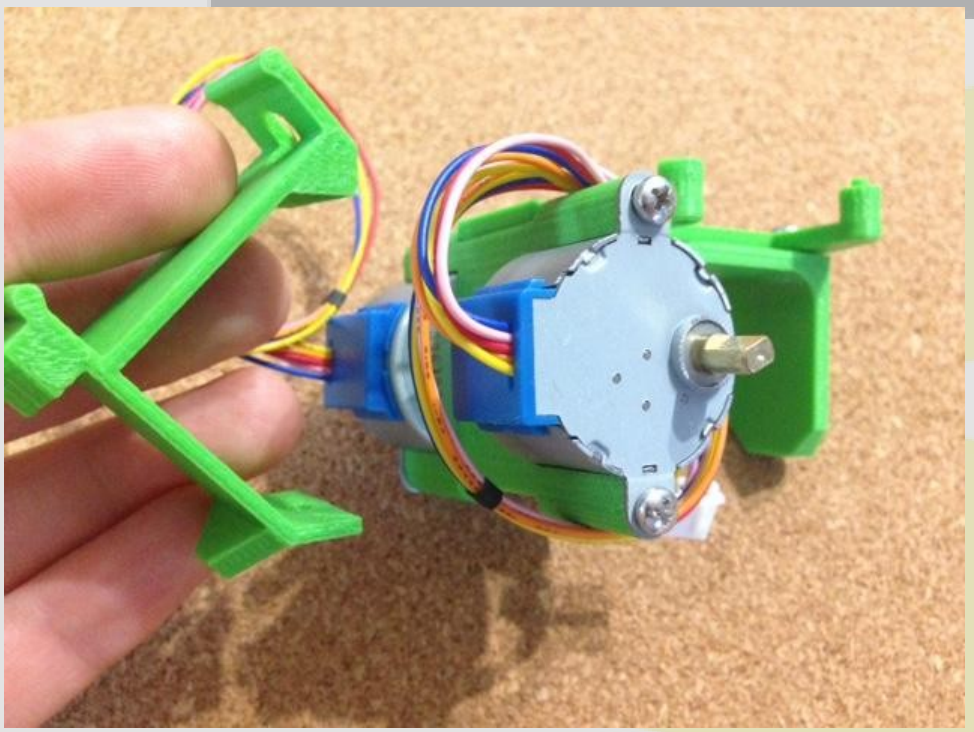


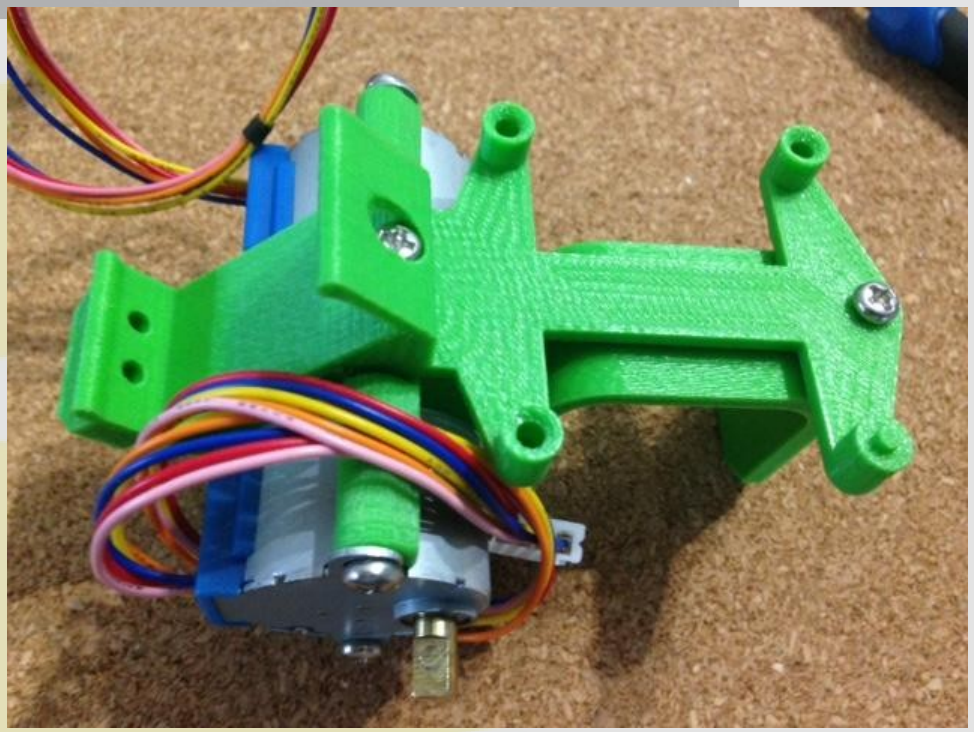
Soporte botonera



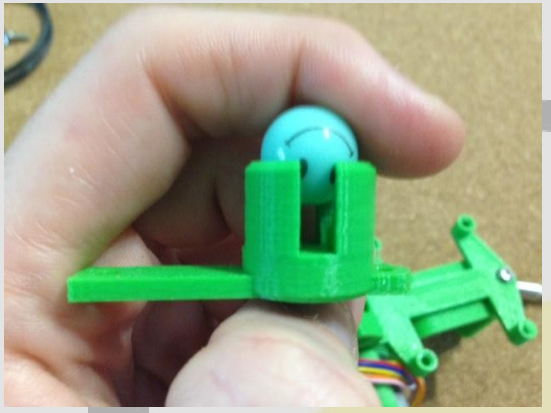


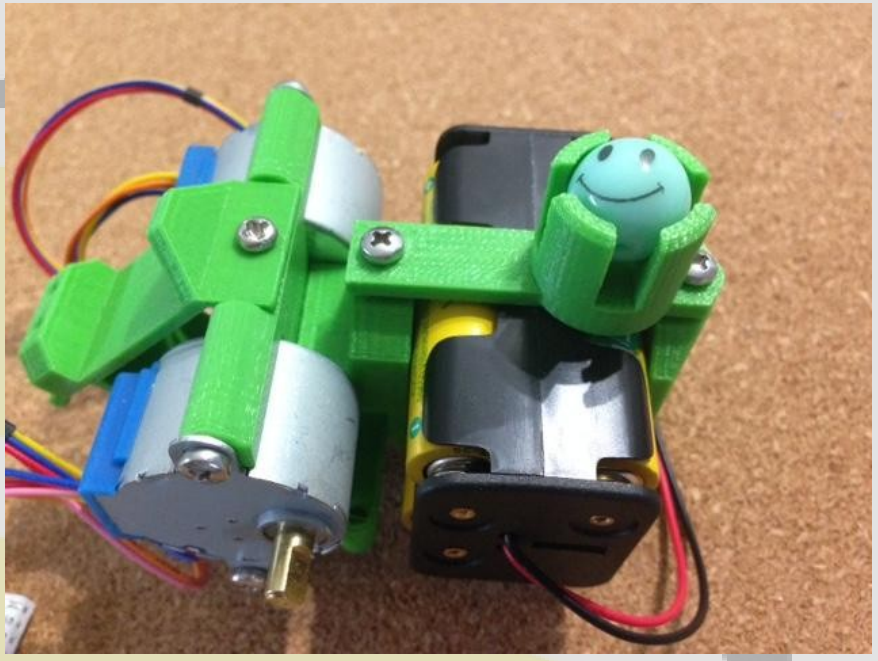
Soporte board

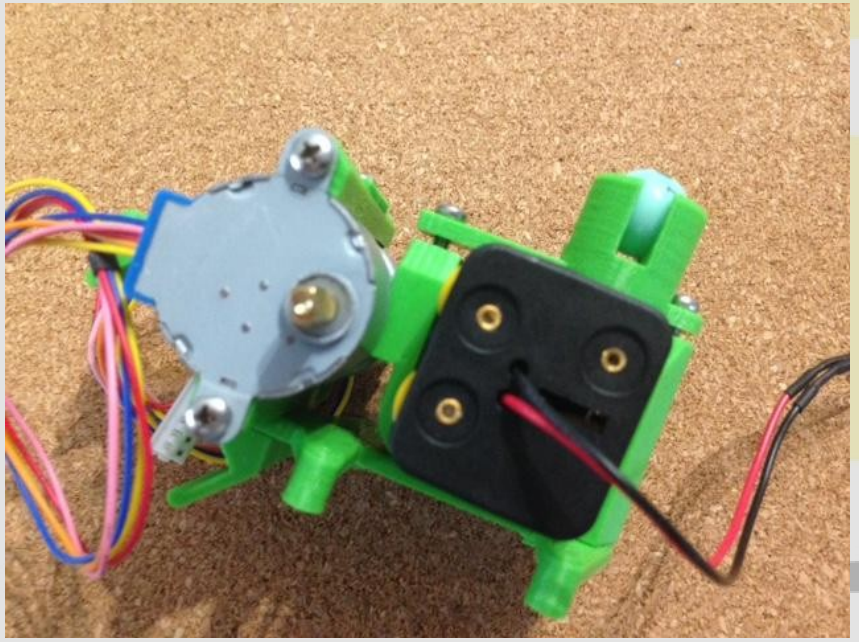




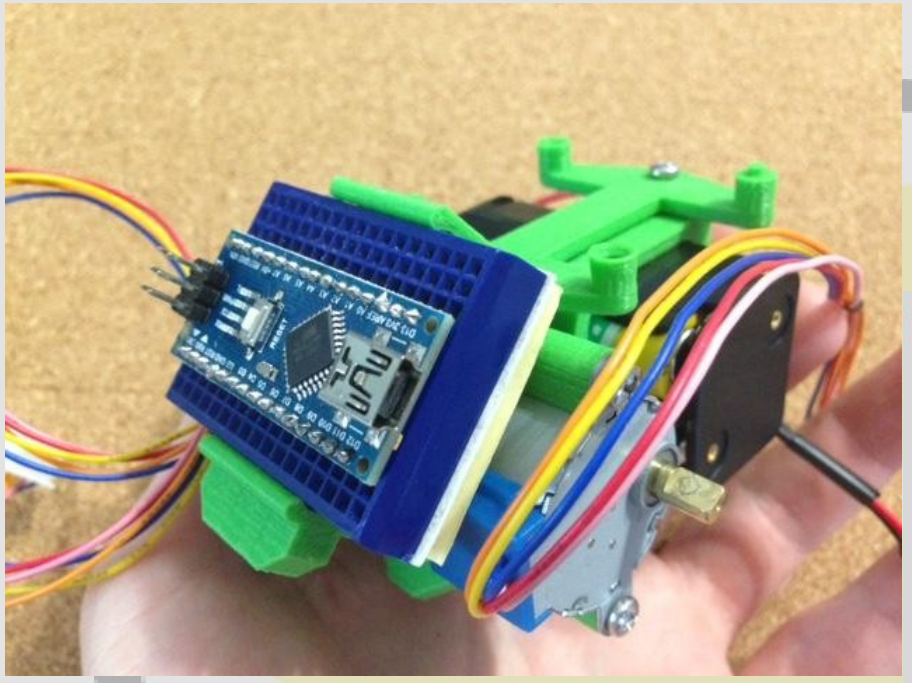
Portapilas y bola

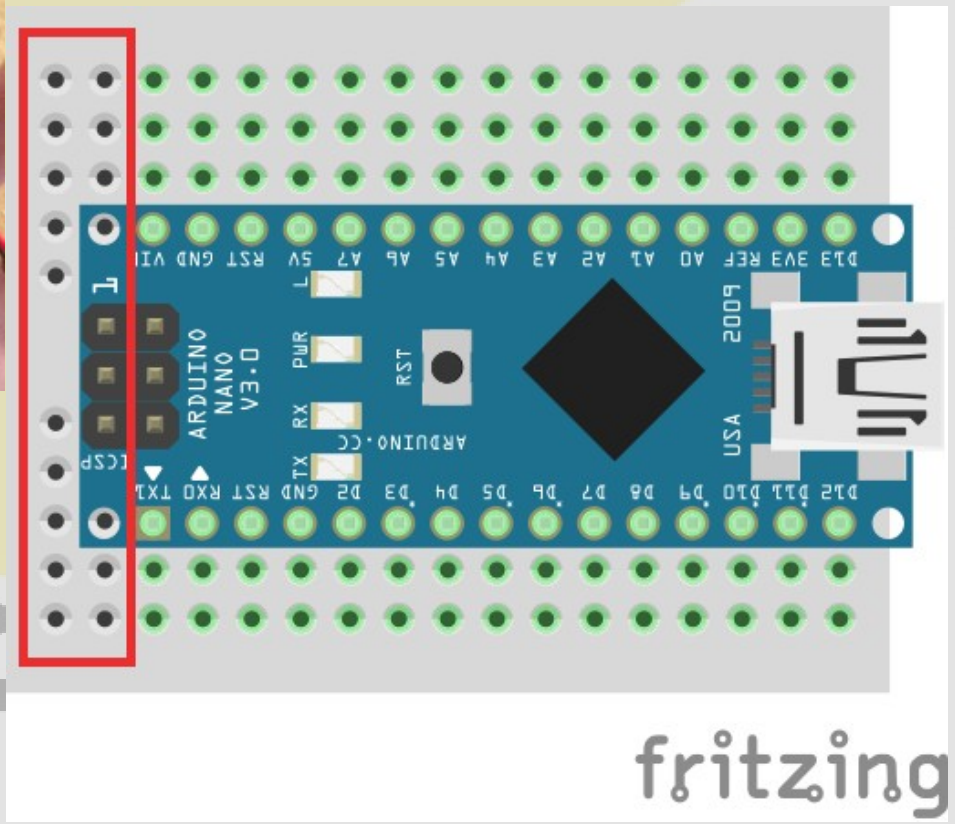




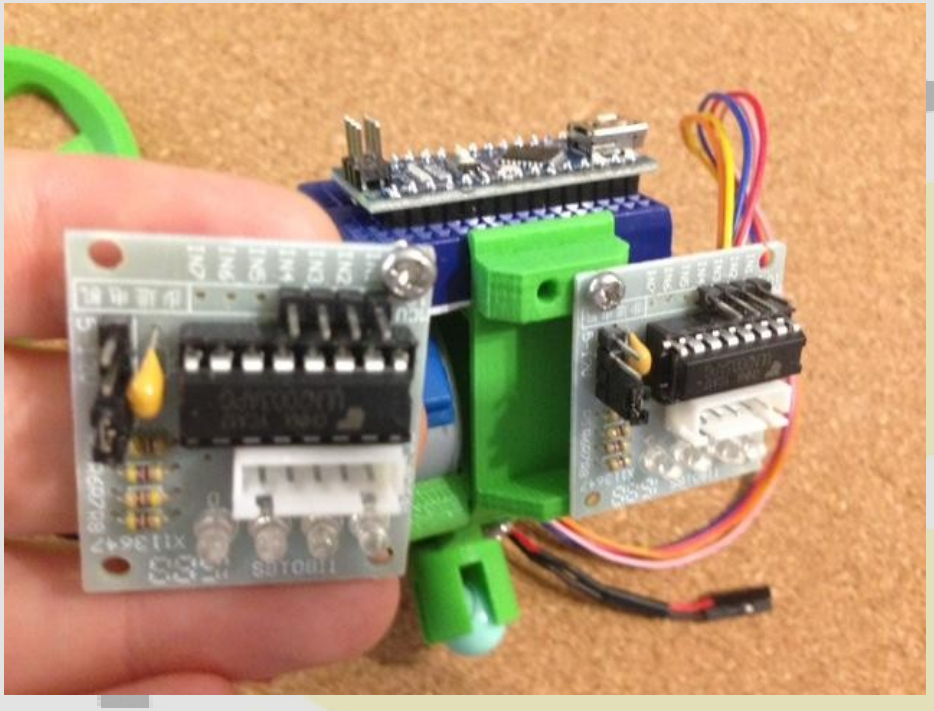


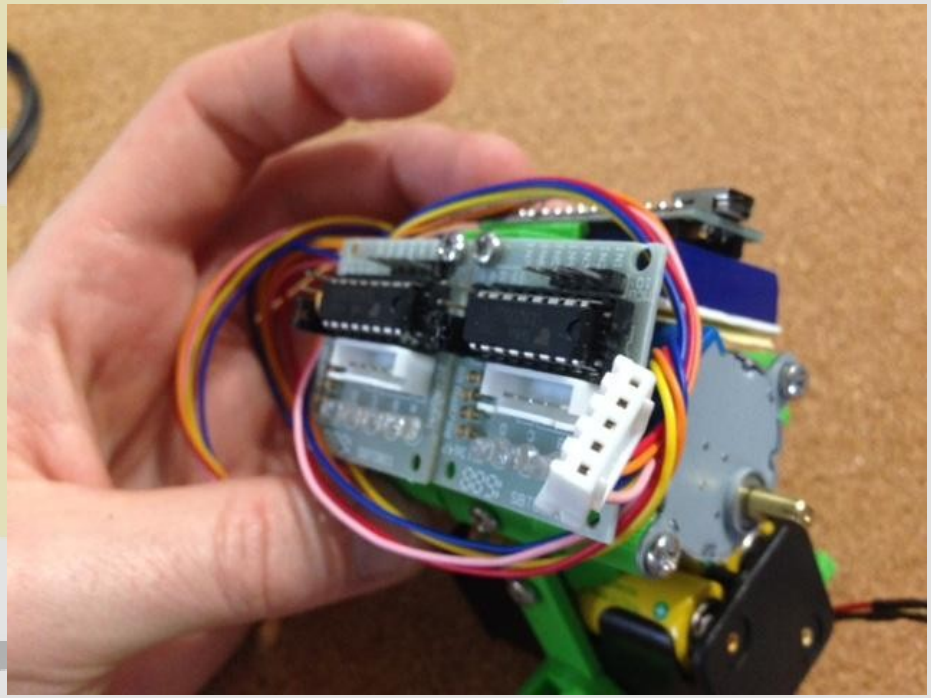
Board



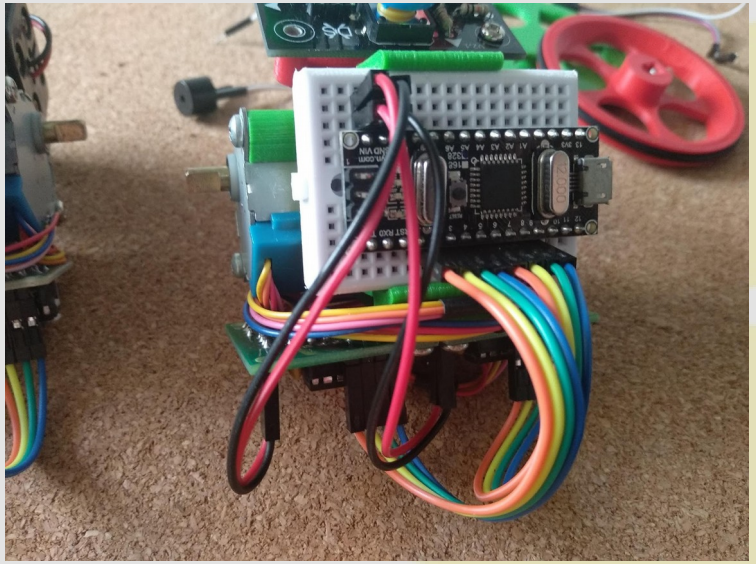

ARDUINO
NANO
V3.0
ARDUINO.CC
USA
2009
PWR
RX
TX
RST
ICSP
fritzing

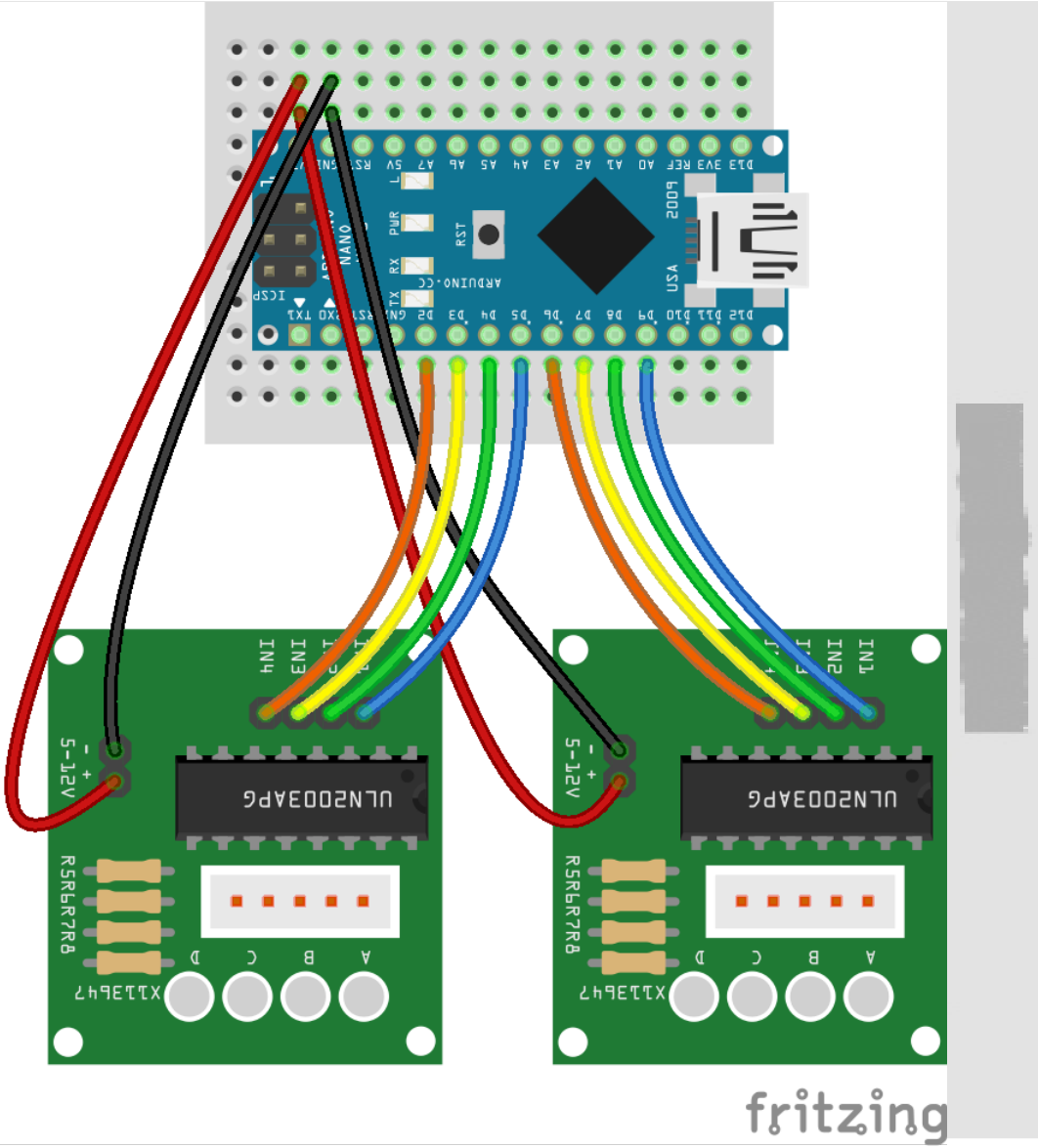
Drivers de motores



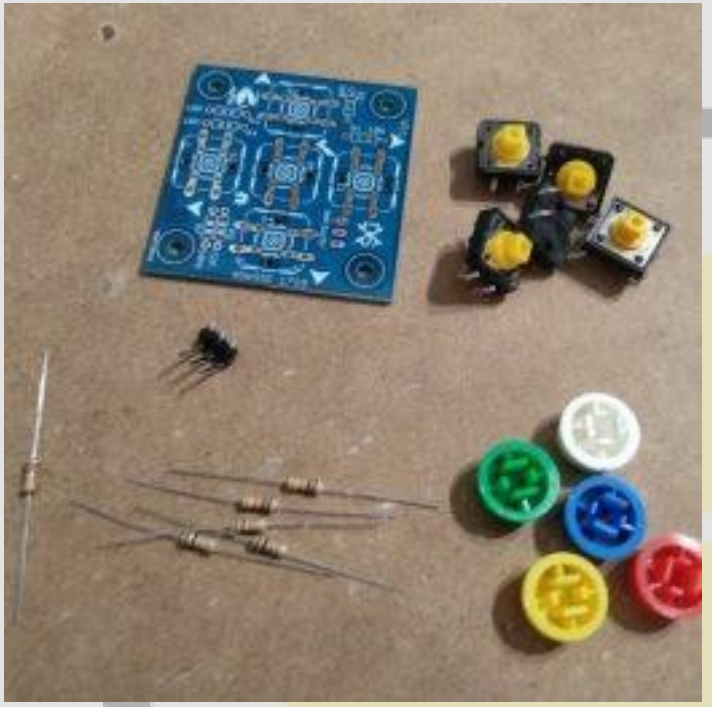


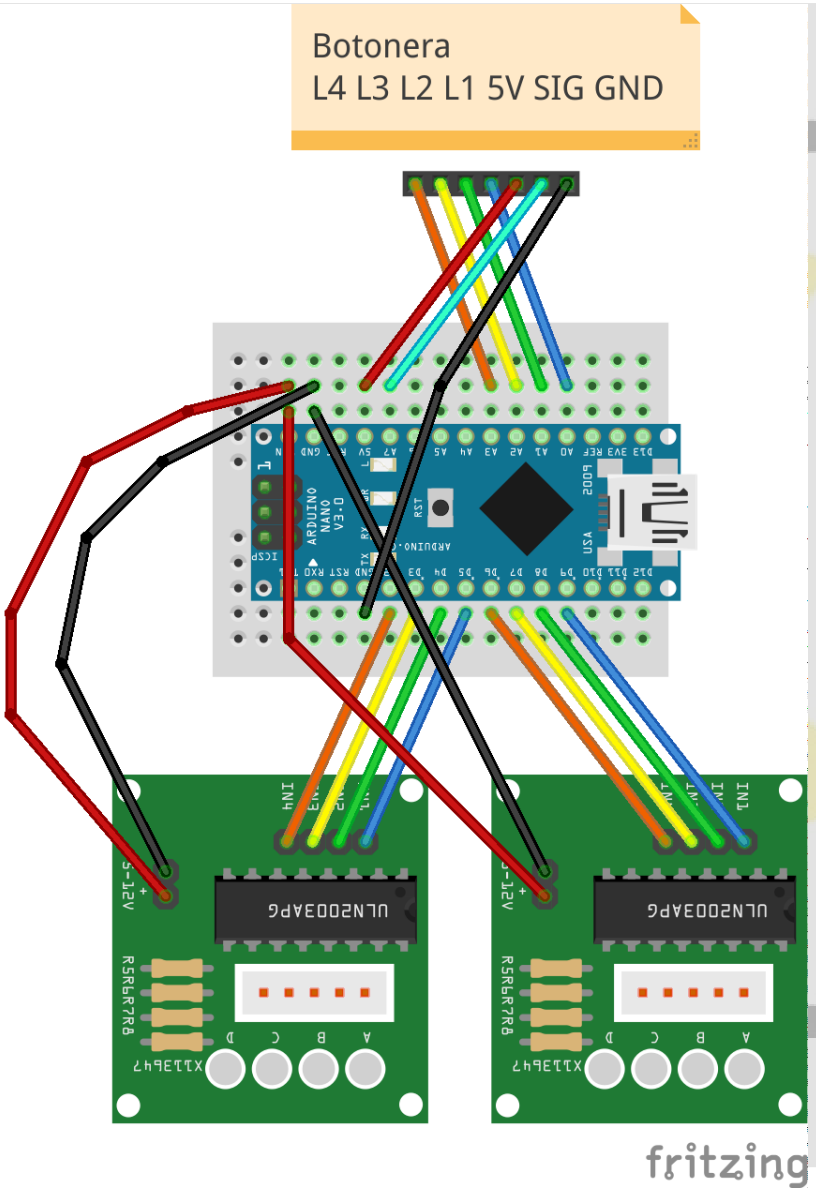
Conexionado drivers





Conexionado Botonera

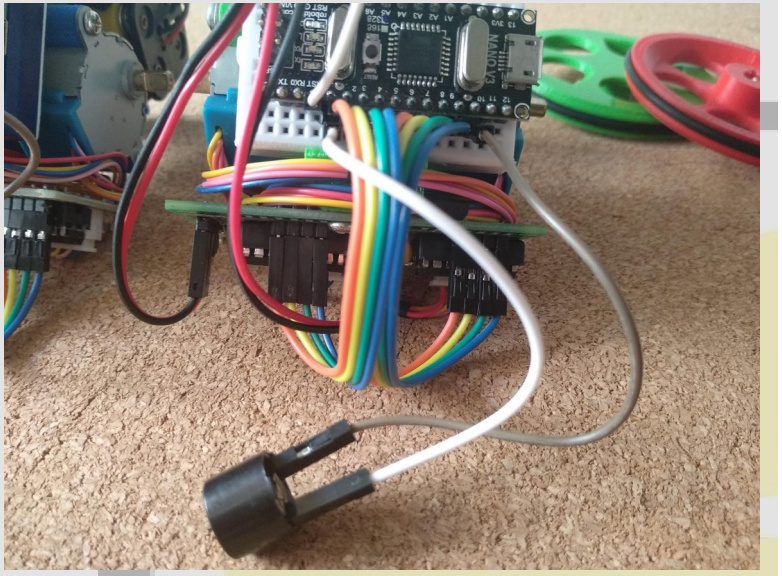


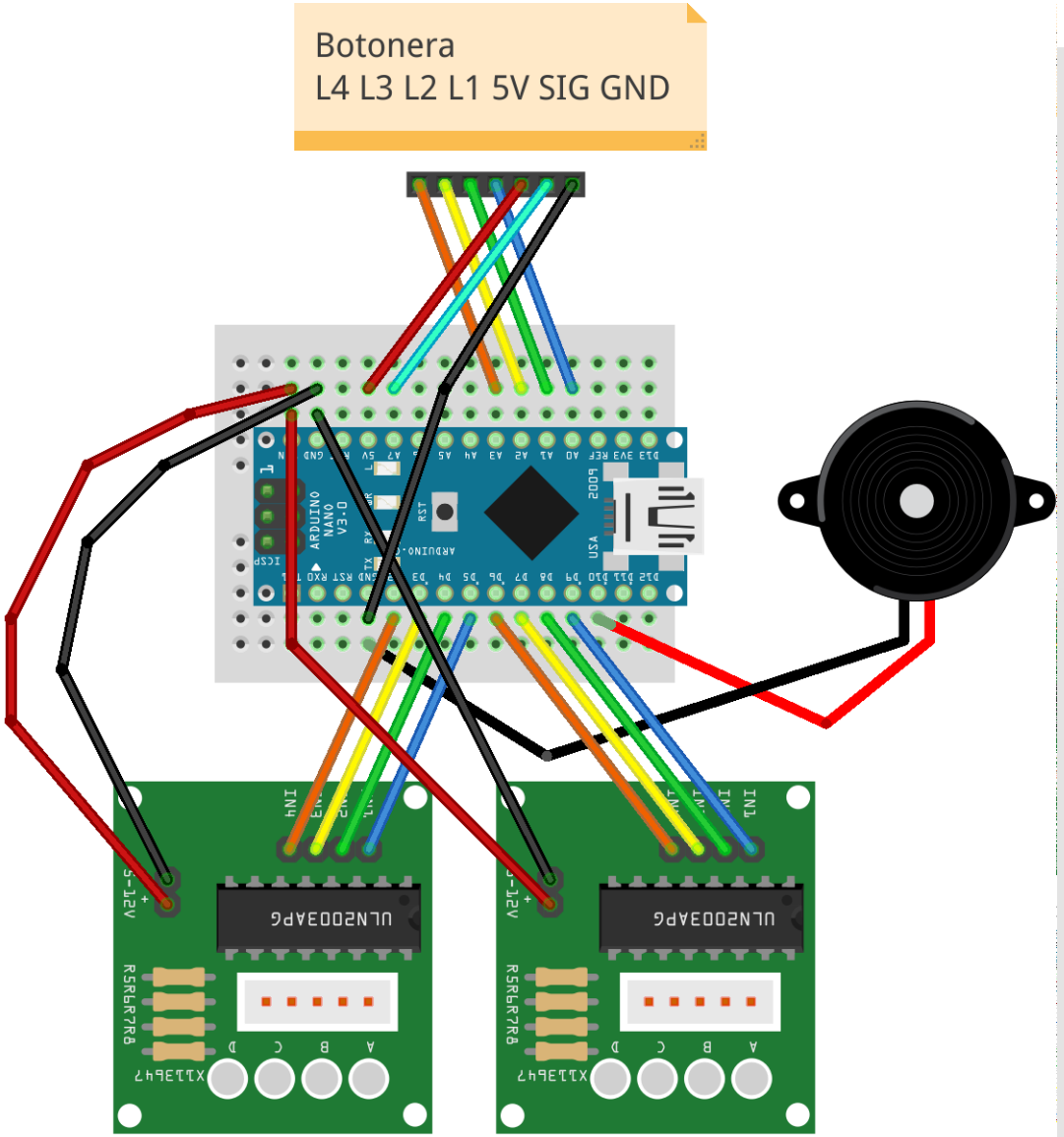


Conexionado botonera

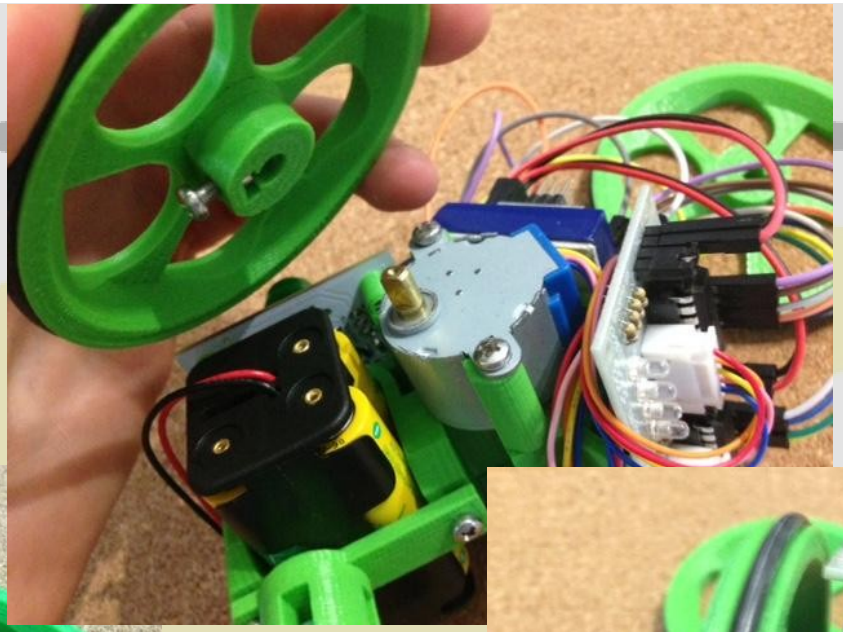
- **Pin gnd: gnd de abajo (al lado D2)**
- **Pin 5V: 5V de arriba**
- **Pin Signal: A7 (arriba)**
- **Pin L1: A0**
- **Pin L2: A1**
- **Pin L3: A2**
- **Pin L4: A3**

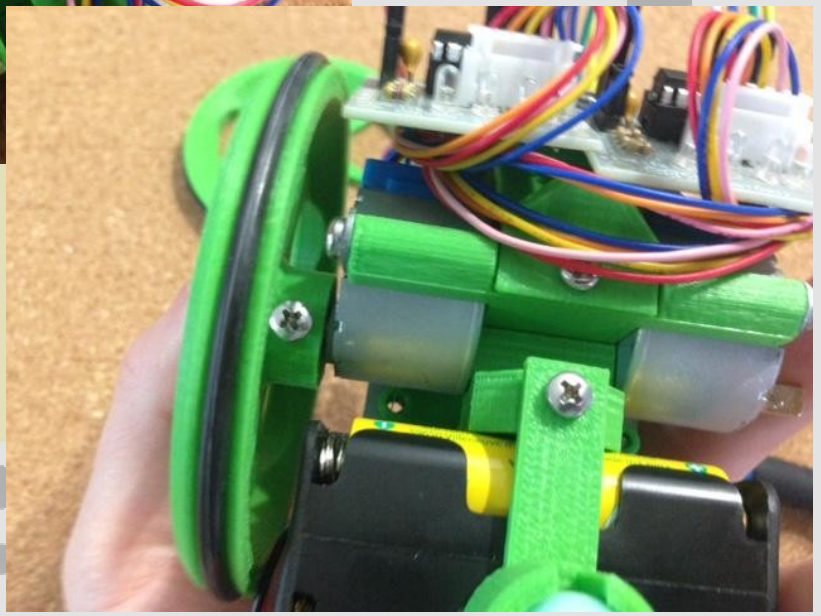
Zumbador y conjunto



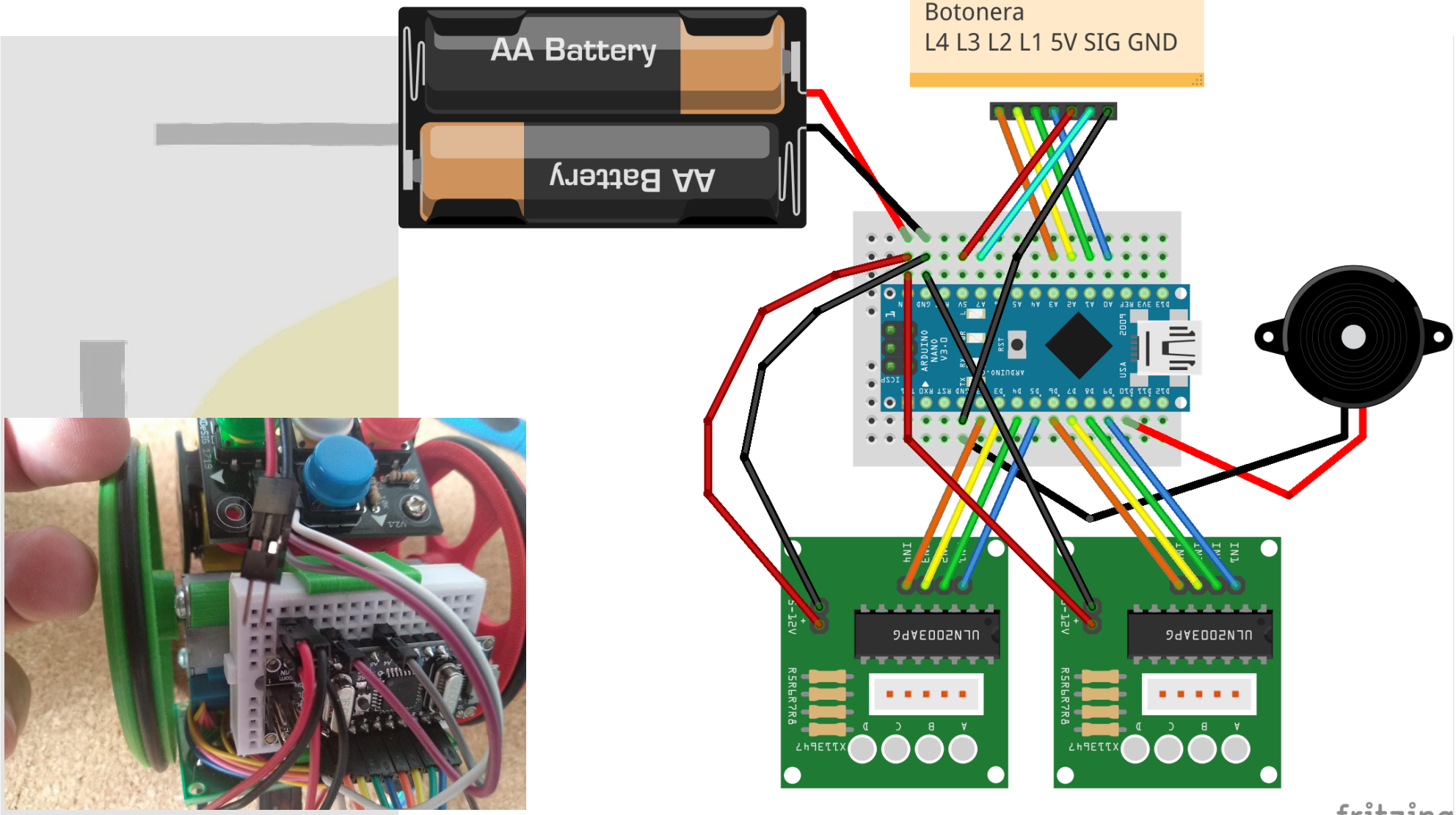


Ruedas

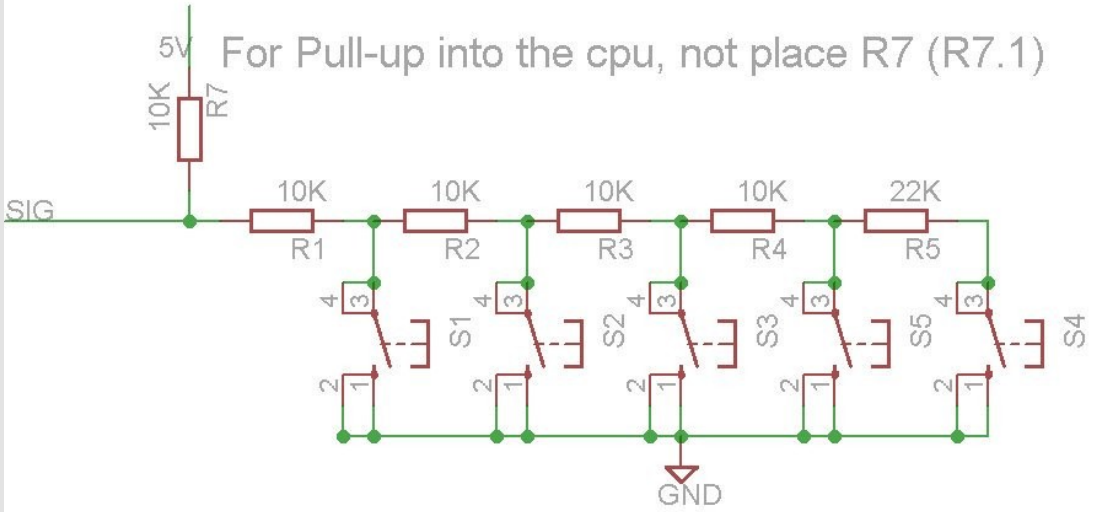


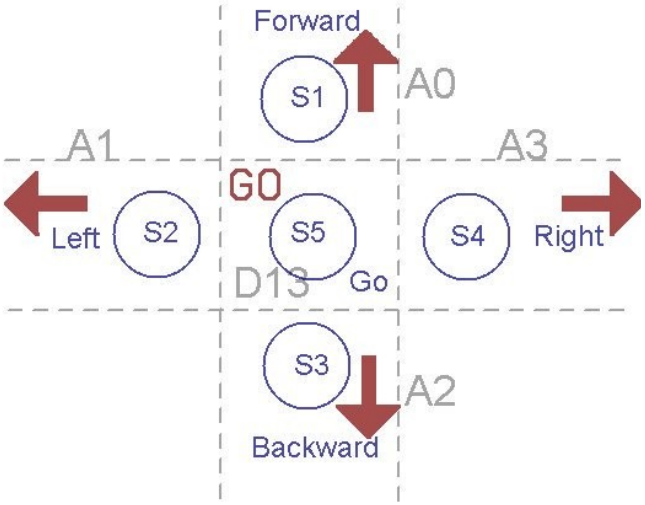


Conexionado Portapilas



Botonera





S1 ▲	S2 ◄	S3 ▼	S5 GO	S4 ►
(10/10*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.

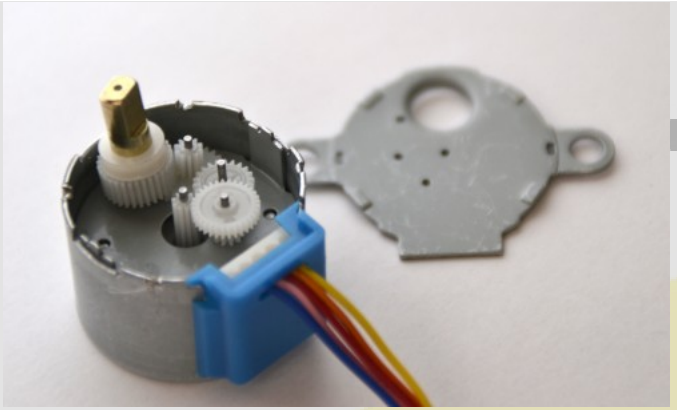
1º: Se carga a arduino nano código de programa de botonera, para detectar valor de pulsación de cada botón y se anotan.
2º: Se carga a arduino el firmware cambiando previamente en la pestaña “Configuration.h” los valores de pulsación de los diferentes botones.

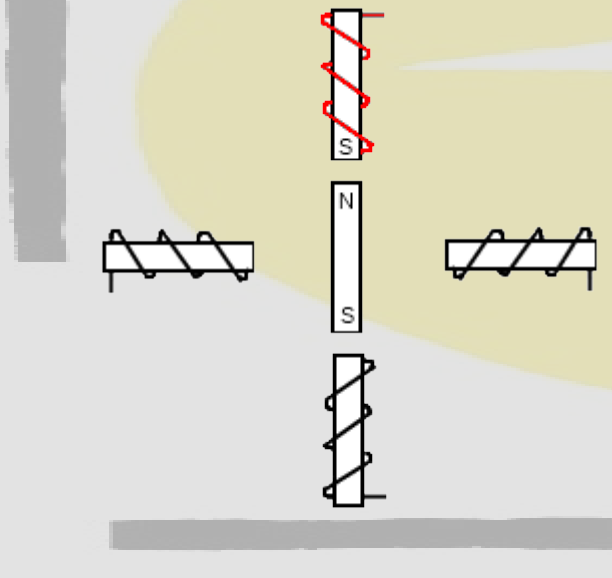
```
////////////////////////////////////////////////////////////////////////////
///// Button set analog
////////////////////////////////////////////////////////////////////////////

#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
```

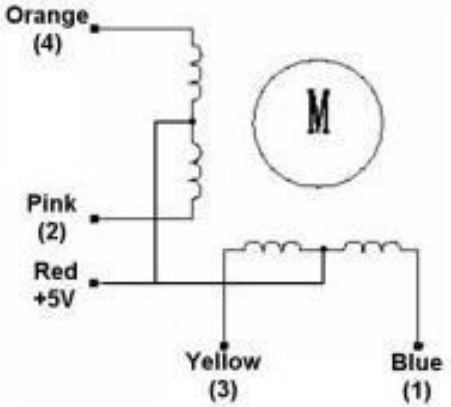
Modos firmware 1.6.2

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

Motor paso a paso







Half-Step Switching Sequence

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 Orange	-	-						-
3 Yellow		-	-	-				
2 Pink				-	-	-		
1 Blue						-	-	-

64 pasos/vuelta x 64 reductora = 4096 pasos para una vuelta precisión 360º/4096=0,088º por paso

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= 2xΠxR
  - **#define STEPPERS_TURN_STEPS 1024**
    Establece un giro de 90º,
    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
```