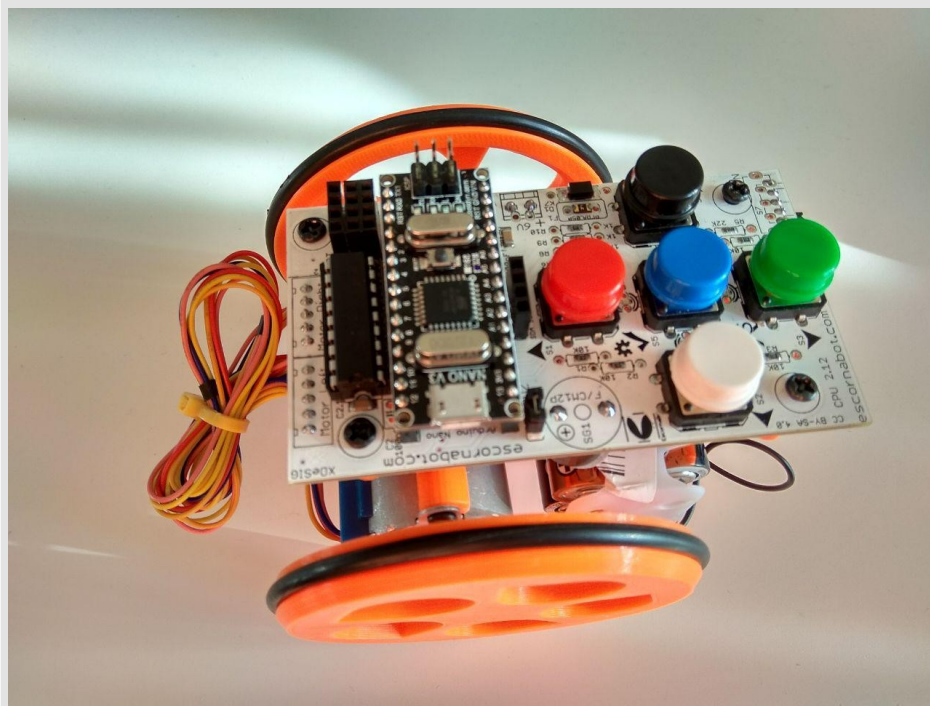


Taller Escornabot DIY

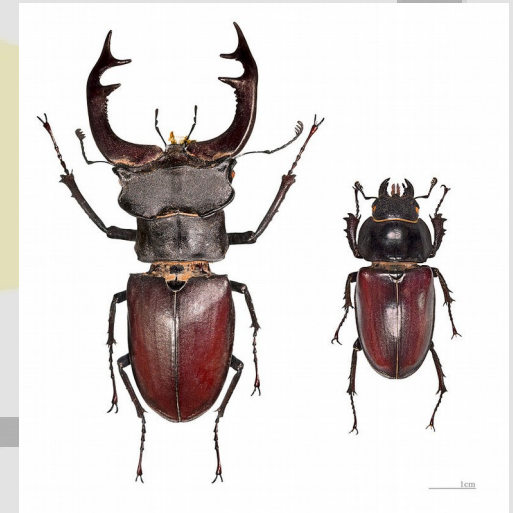
Club Robótica Granada



¿Escorna que ...?



ESCORNABOI
(escarabajo
lucanus cervus en
galego) + BOT
(robot)



Objetivo

ROBÓTICA Y PROGRAMACIÓN



Sustituye a Robots privados



Características

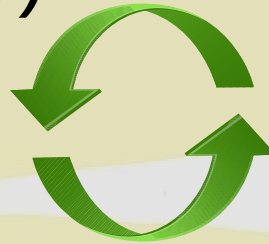
- Lo haces tú
- Hardware abierto y Software libre
- Asequible
- Bien documentado

¿Quién?

Equipo de desarrollo (Github

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

Grupo de google)



Comunidad (Grupo de google /
Telegram)

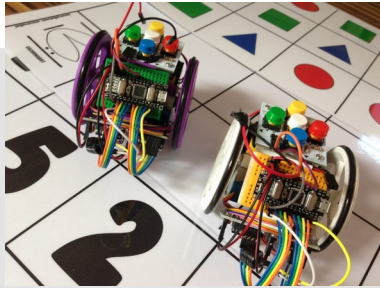
iii Escornafan !!!

Pablo Rubio

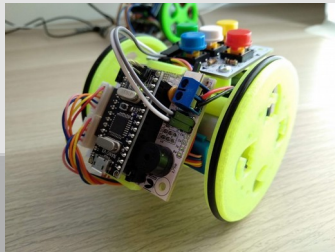
(<https://pablorubma.cc/>)



Versiones



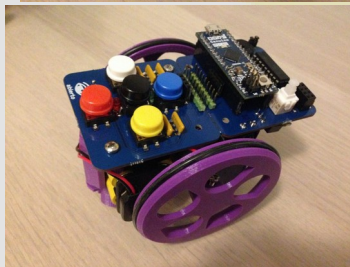
Do It Yourself (DIY)



Compactus



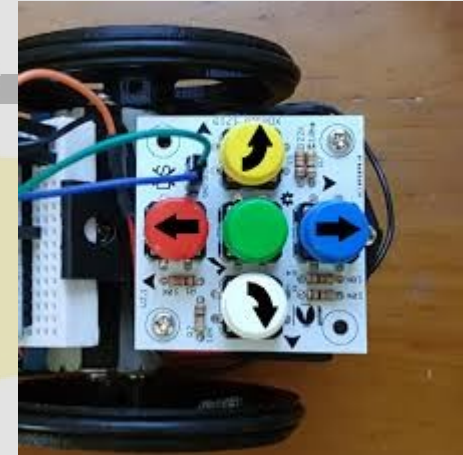
Placa 2.12



Okagi

Funcionamiento y programación

- Introduce Firmware y se maneja con botonera (modo clásico)
- Se puede programar con librería para arduino e incluso poner sensores extras



```
Blink | Arduino 1.8.5

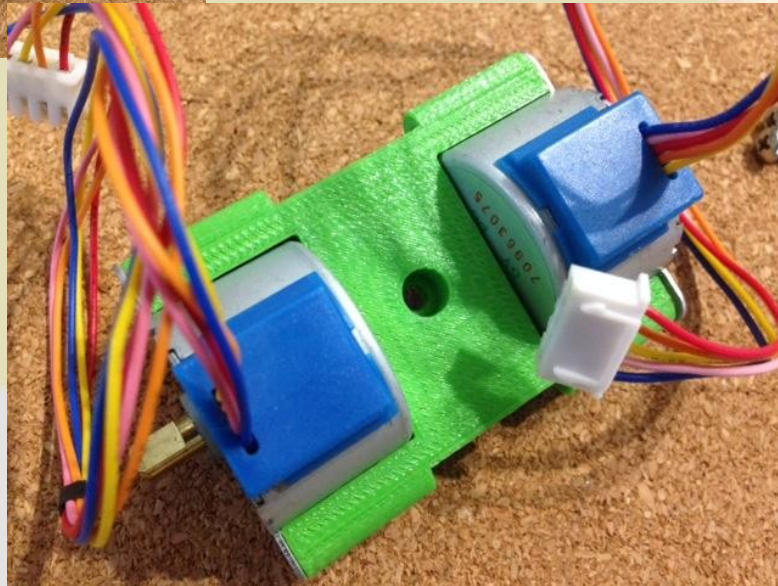
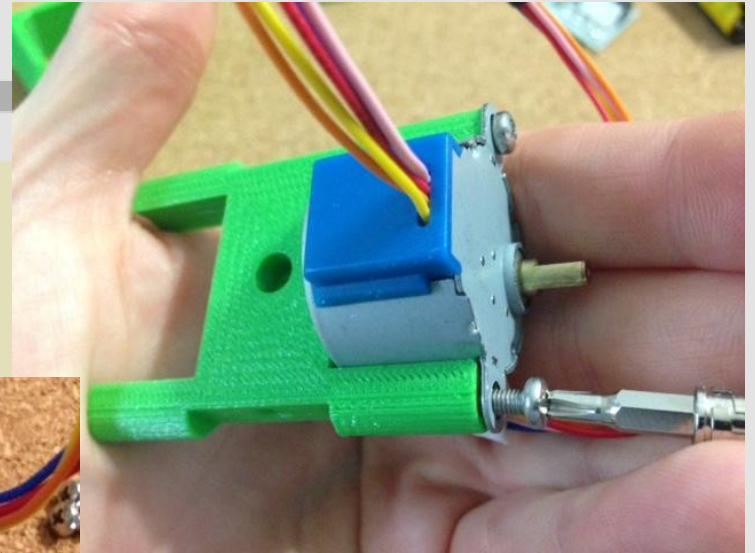
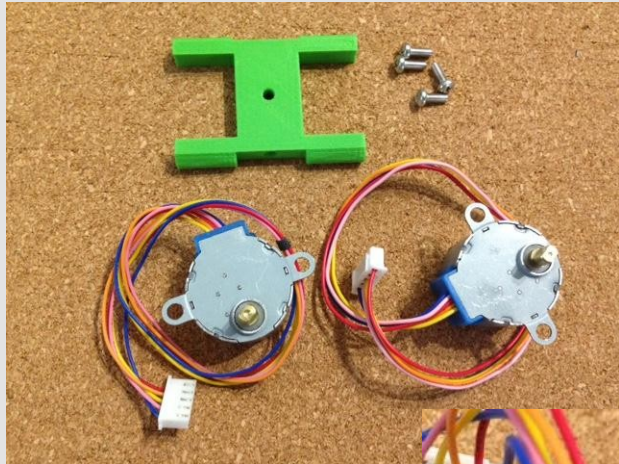
This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink

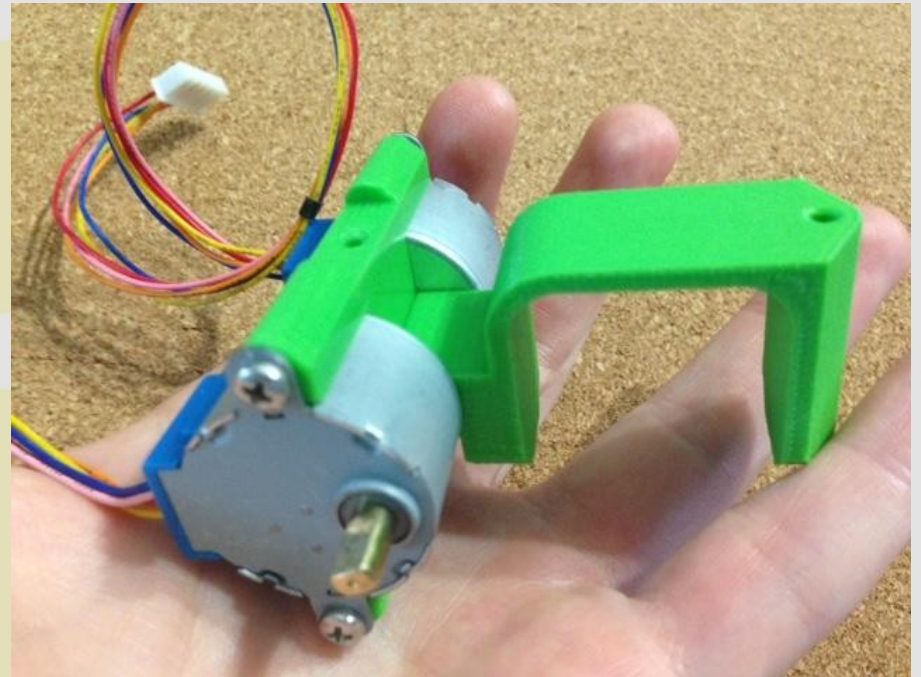
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

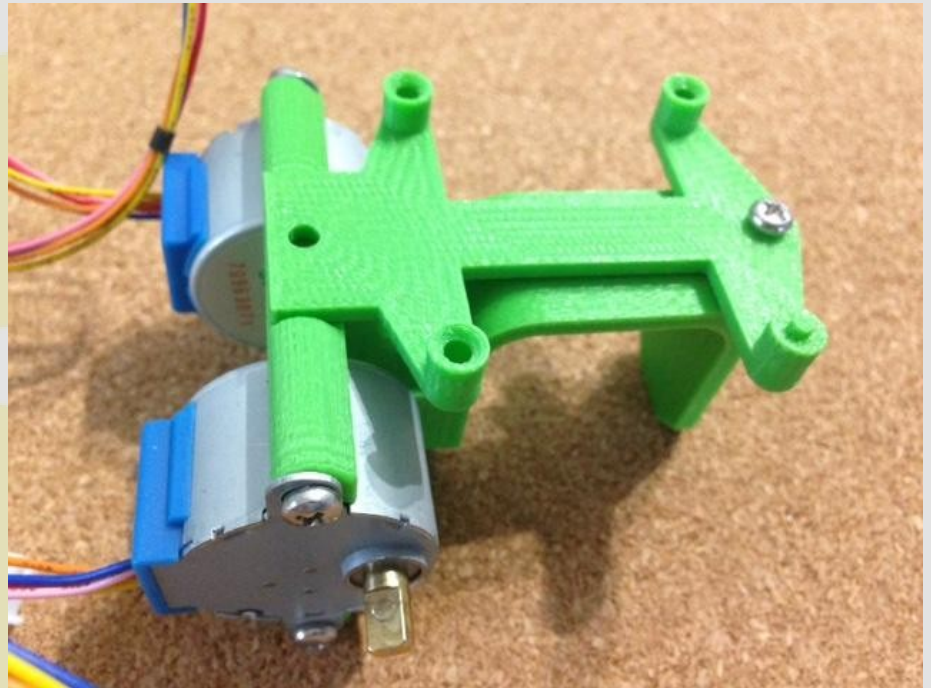
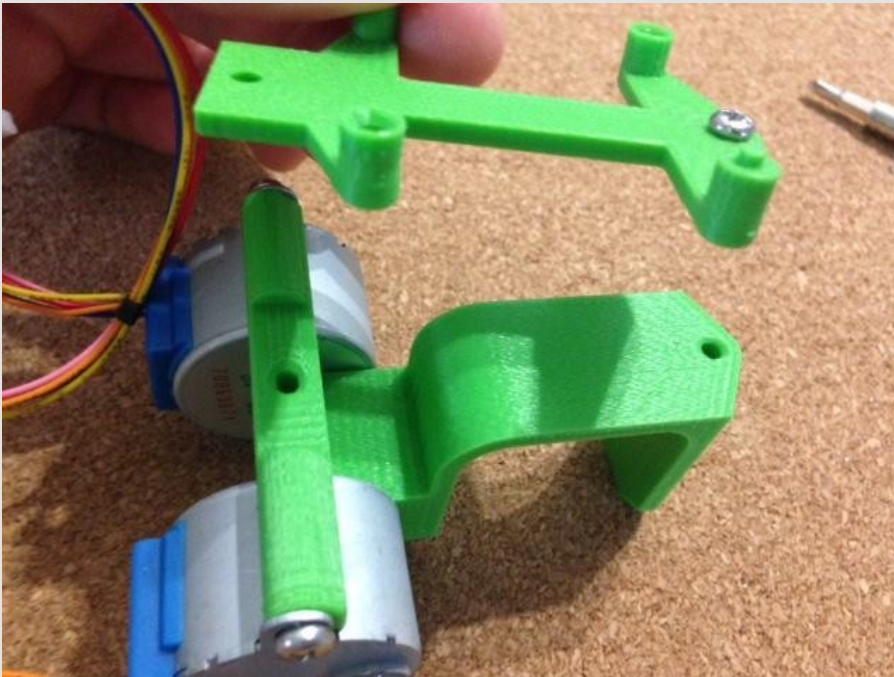

Motores



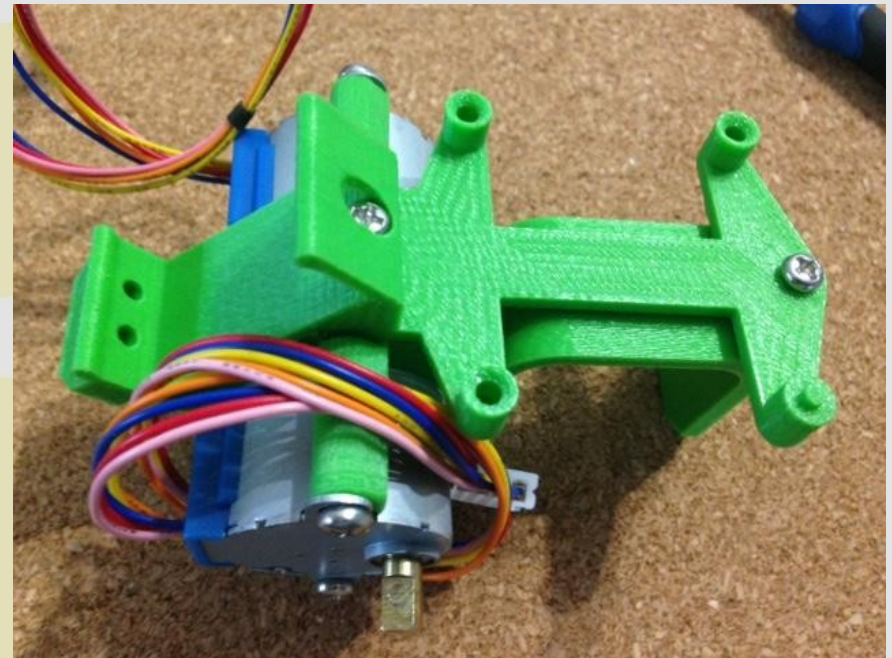
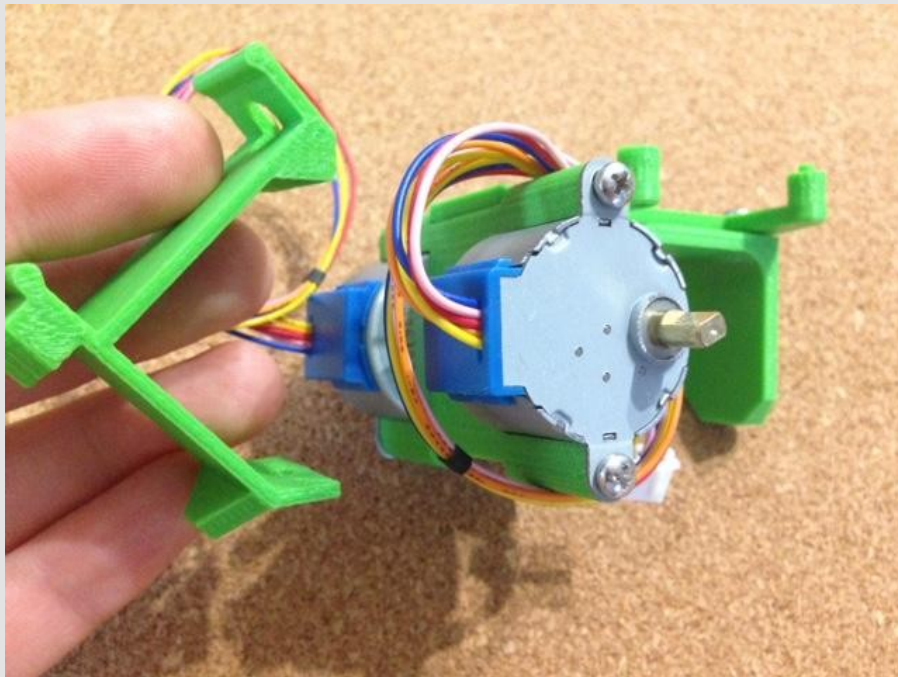
Soporte portapilas



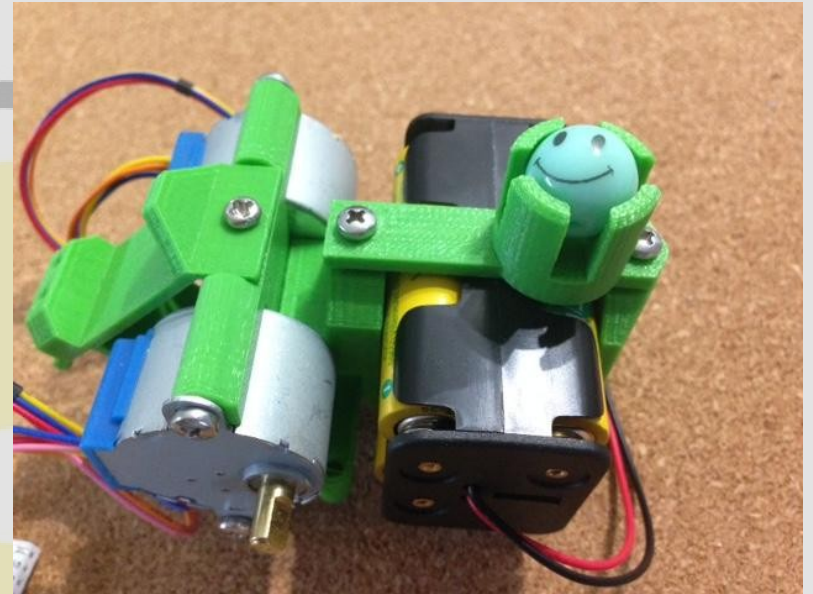
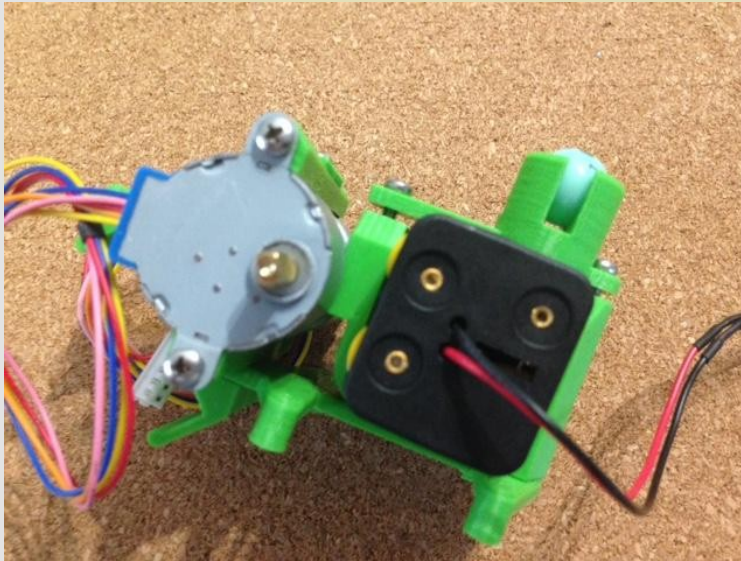
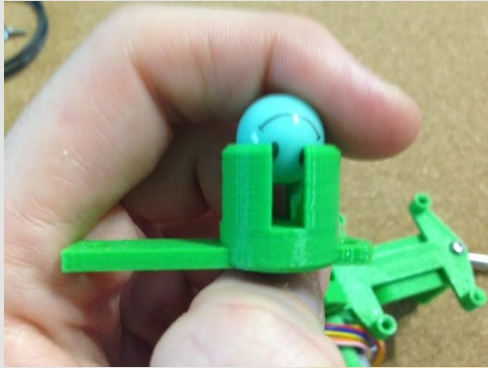
Soporte botonera



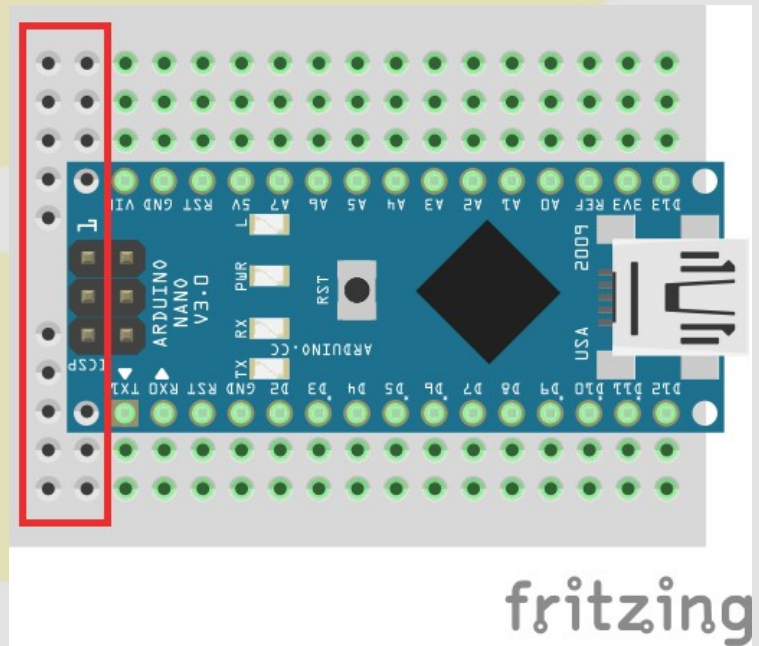
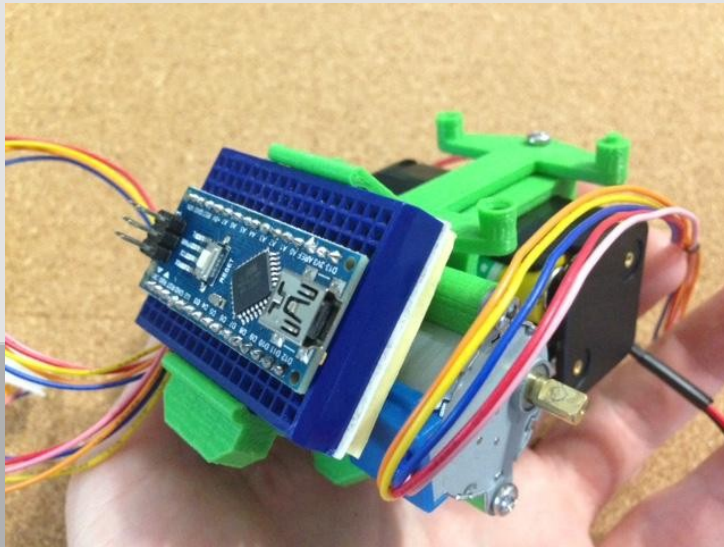
Soporte board



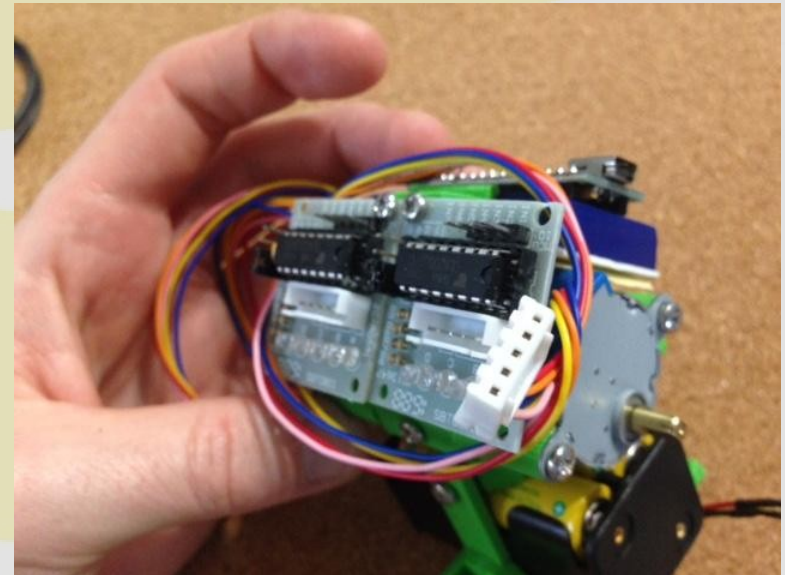
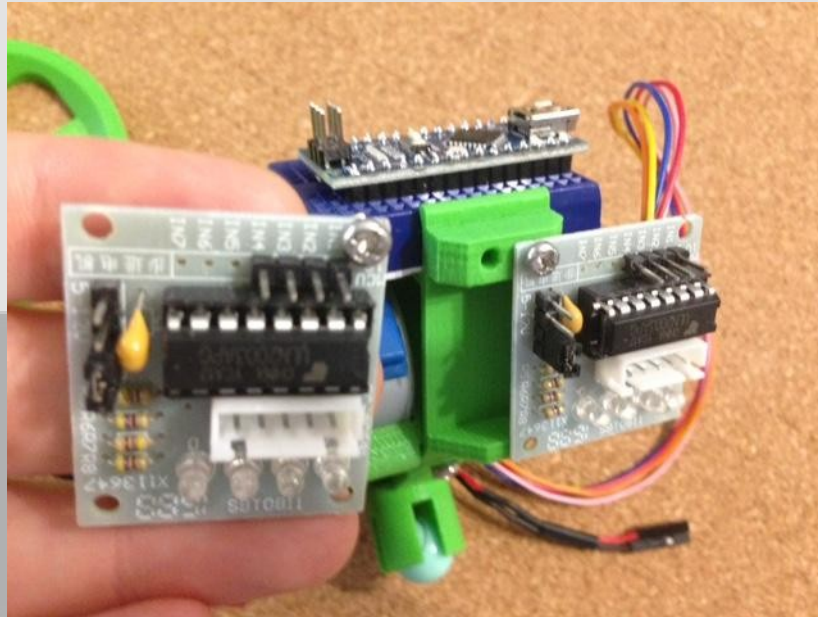
Portapilas y bola



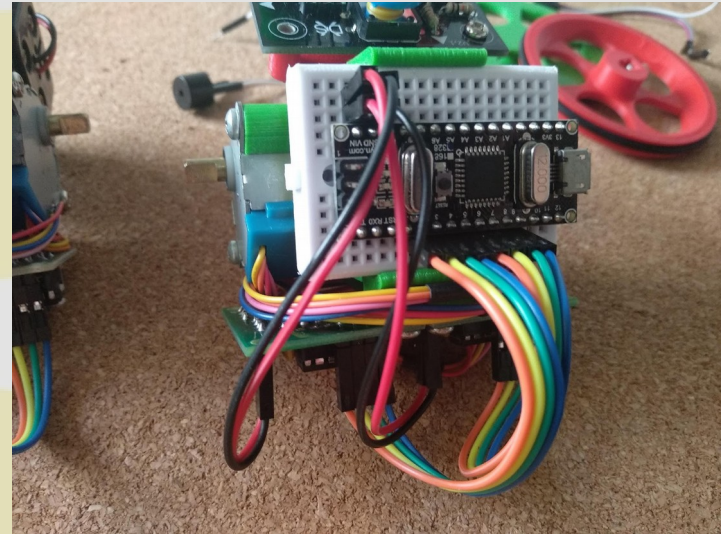
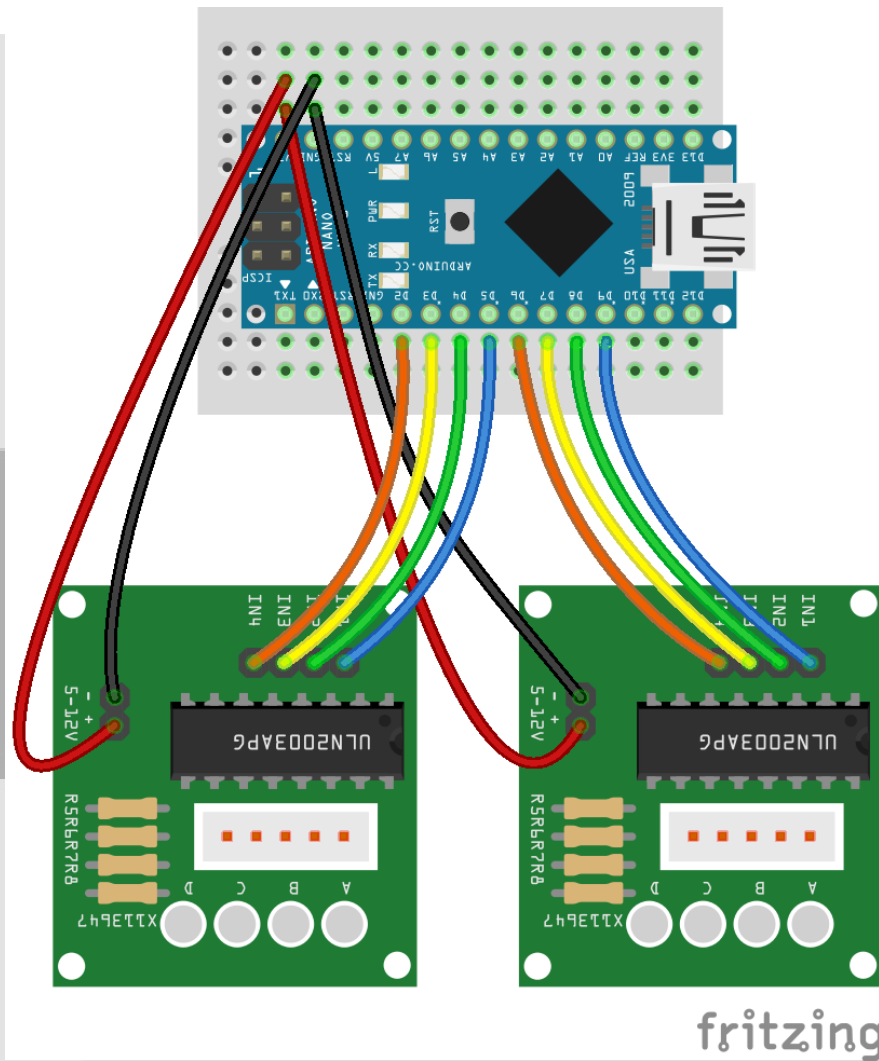
Board



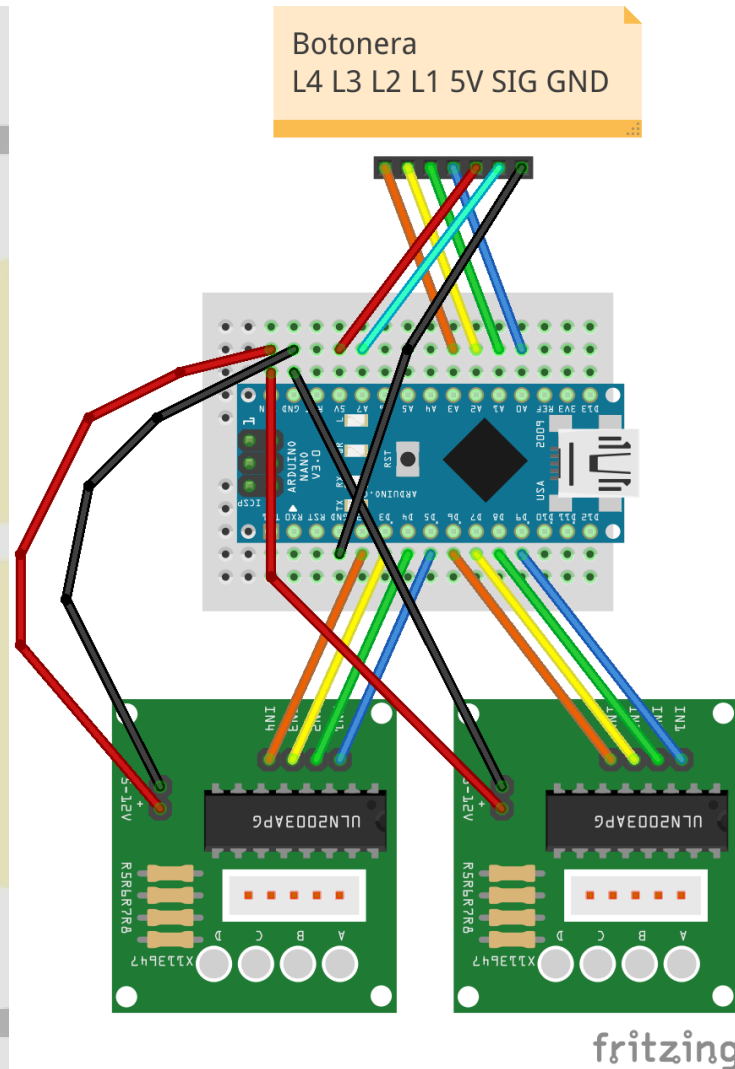
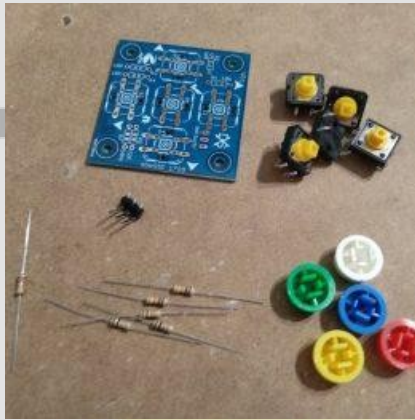
Drivers de motores



Conexionado drivers



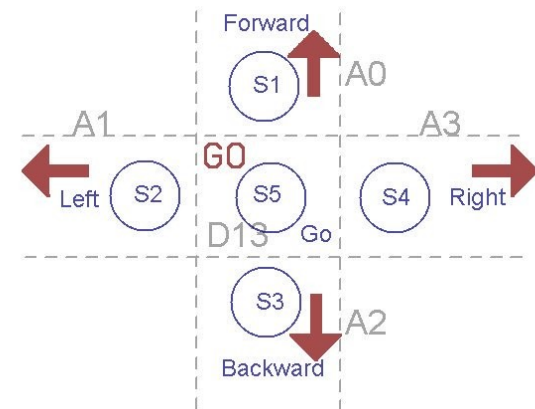
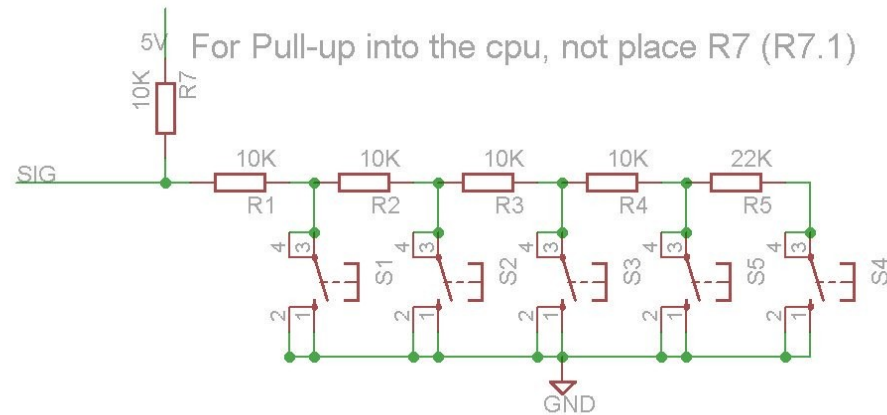
Conexionado Botonera



Conexionado botonera

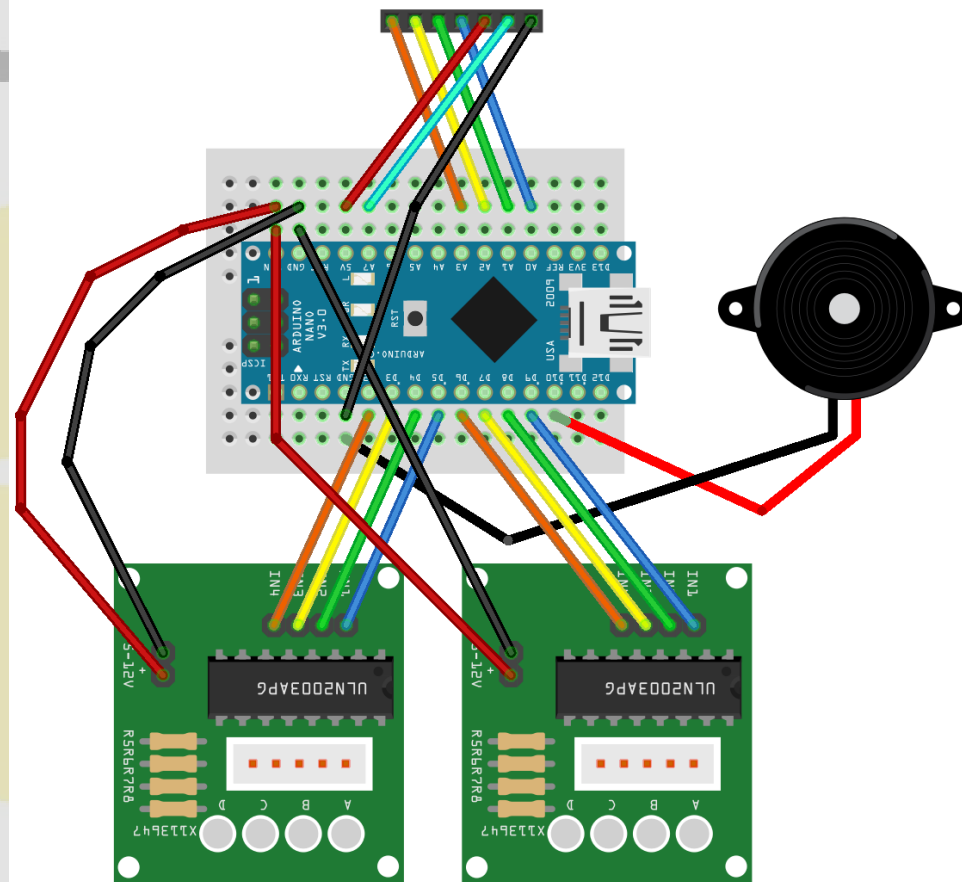
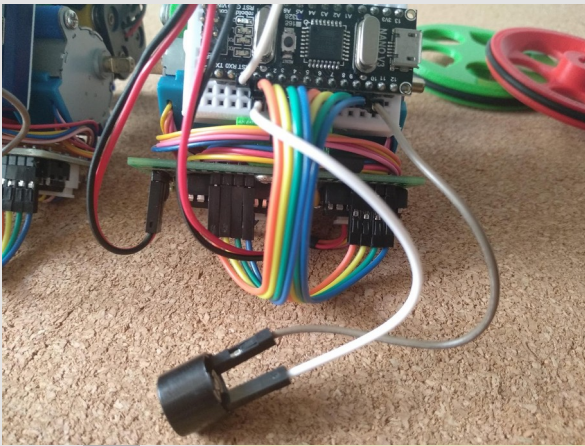
- Pin gnd: gnd de abajo (al lado D2)
- Pin 5V: 5V de arriba
- Pin Signal: A7 (arriba)
- Pin L1: A0
- Pin L2: A1
- Pin L3: A2
- Pin L4: A3

Botonera



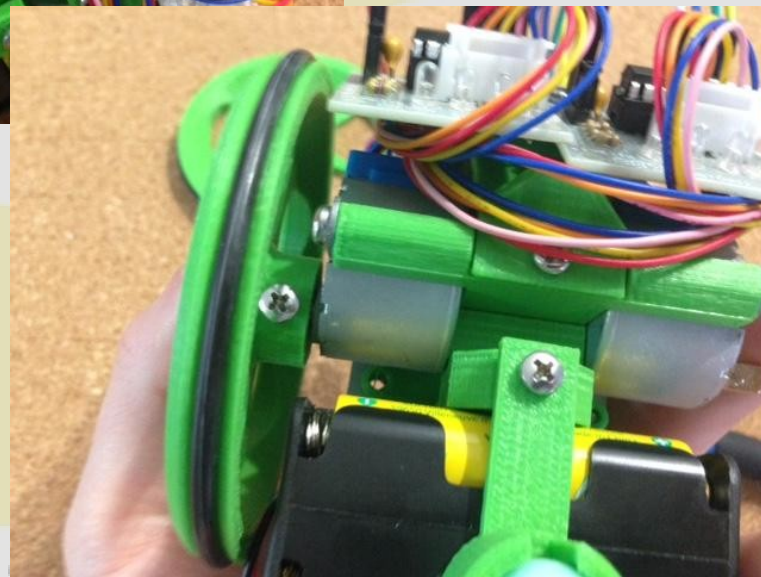
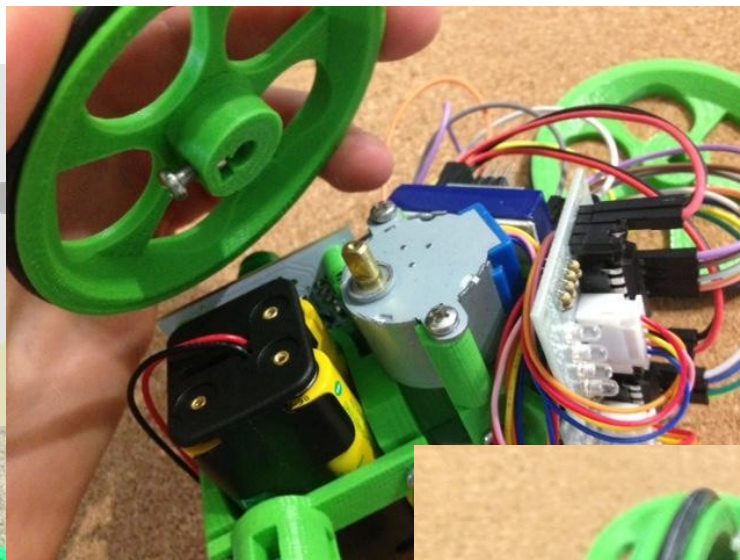
Zumbador y conjunto

Botonera
L4 L3 L2 L1 5V SIG GND

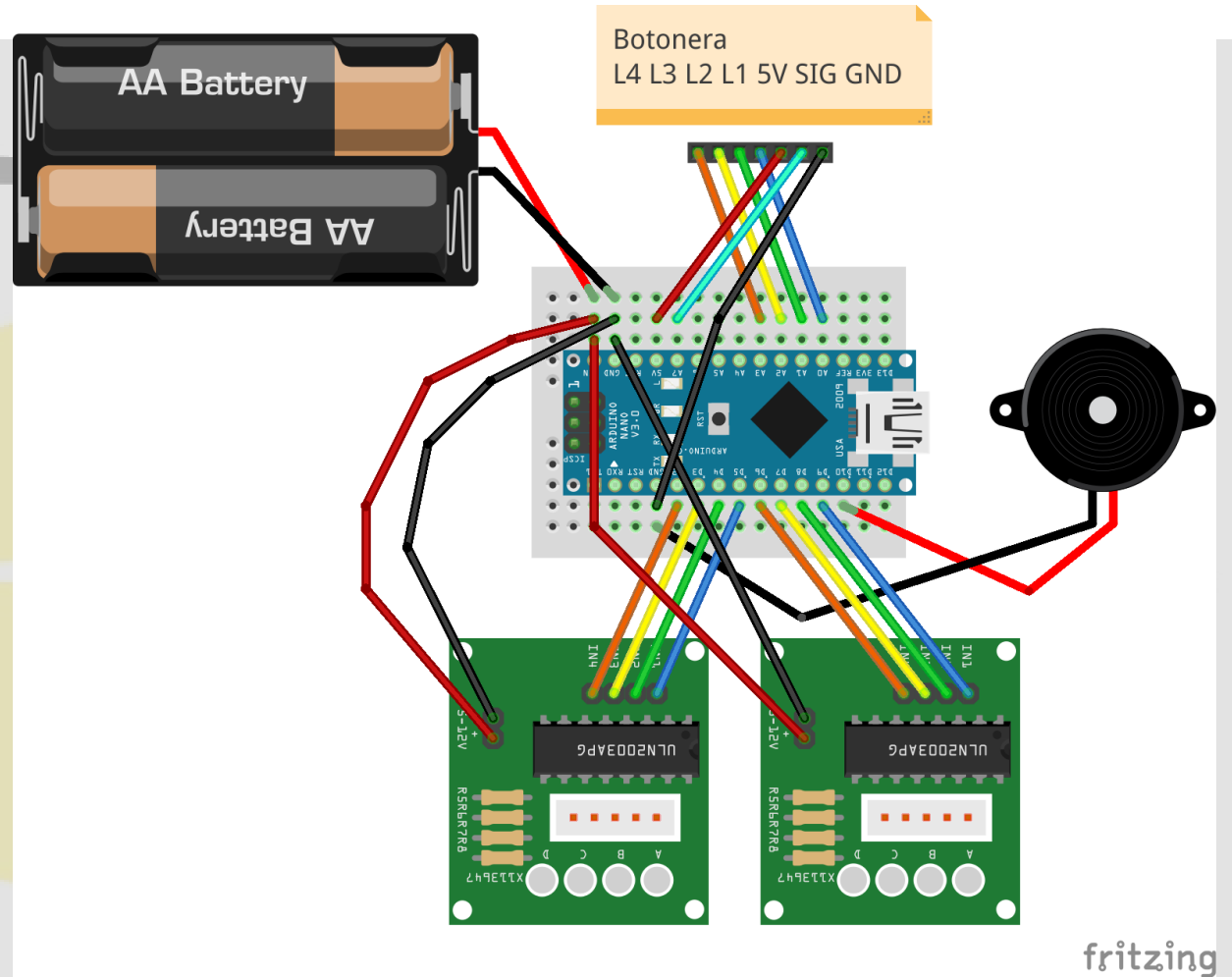
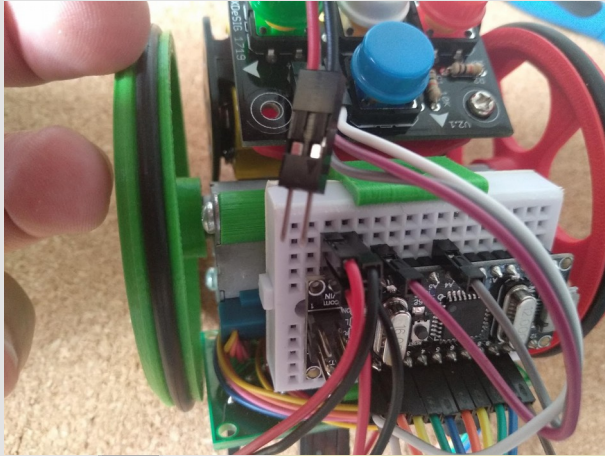


fritzing

Ruedas



Conexionado Portapilas



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.

Cambios en firmware

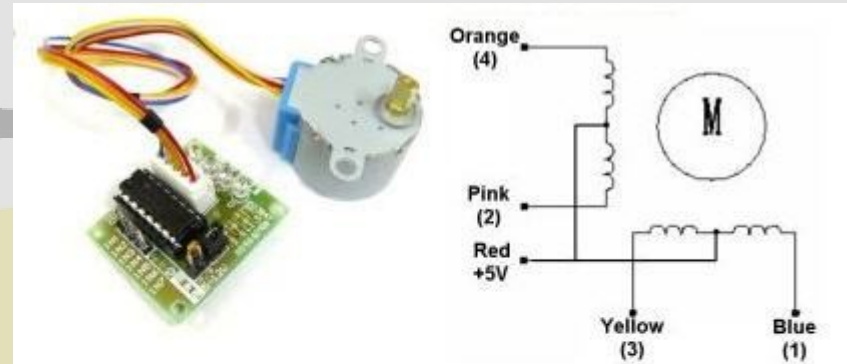
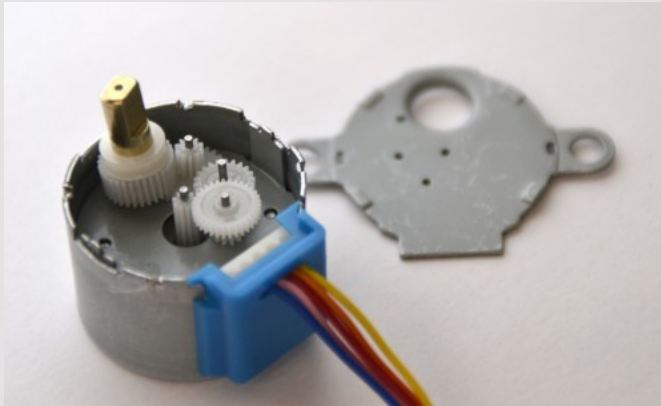
- Abrimos Escornabot.ino, pestaña Configuration.h
 - #define STEPPERS_STEPS_PER_SECOND 1000: numero de pasos por segundo, el tope anda sobre 2300
 - #define STEPPERS_LINE_STEPS 1738: da un avance de 10cm
 - #define STEPPERS_TURN_STEPS 1024: establece un giro de 90°, una vuelta completa 4096 pasos

```
#ifndef 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
```

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

