

Programación de un ESC con Arduino (http://robologs.net/2016/02/01/programacion-de-un-esc-con-arduino/#comments)

Posted on February 1, 2016 (<http://robologs.net/2016/02/01/programacion-de-un-esc-con-arduino/>) by Tr4nsduc7or (<http://robologs.net/author/7472616e73647563746f72/>)



(http://robologs.net/wp-content/uploads/2016/02/ESC_portada.png)

Saludos, humano. ¿Alguna vez has intentado calibrar un ESC con Arduino? Si lo has hecho, habrás visto que no es nada fácil: hay que hacer todo un ritual para activar el ESC, y después generar un pulso para controlarlo. Además, cada uno de estos aparatos es un mundo y se activa de forma diferente. Yo mismo me he quebrado mis circuitos neuronales para intentar arrancar uno. Y no es agradable.

Así que para ayudarte a conservar tu estabilidad mental, te explicaré como **programar un ESC con Arduino**. El objetivo será **escribir un código para calibrarlo y después variar la velocidad de un motor brushless** enviando comandos por Serial (un poco a la línea de este tutorial (<http://robologs.net/2015/07/18/tutorial-de-arduino-controlar-dos-servos-por-serial/>), pero más simple).



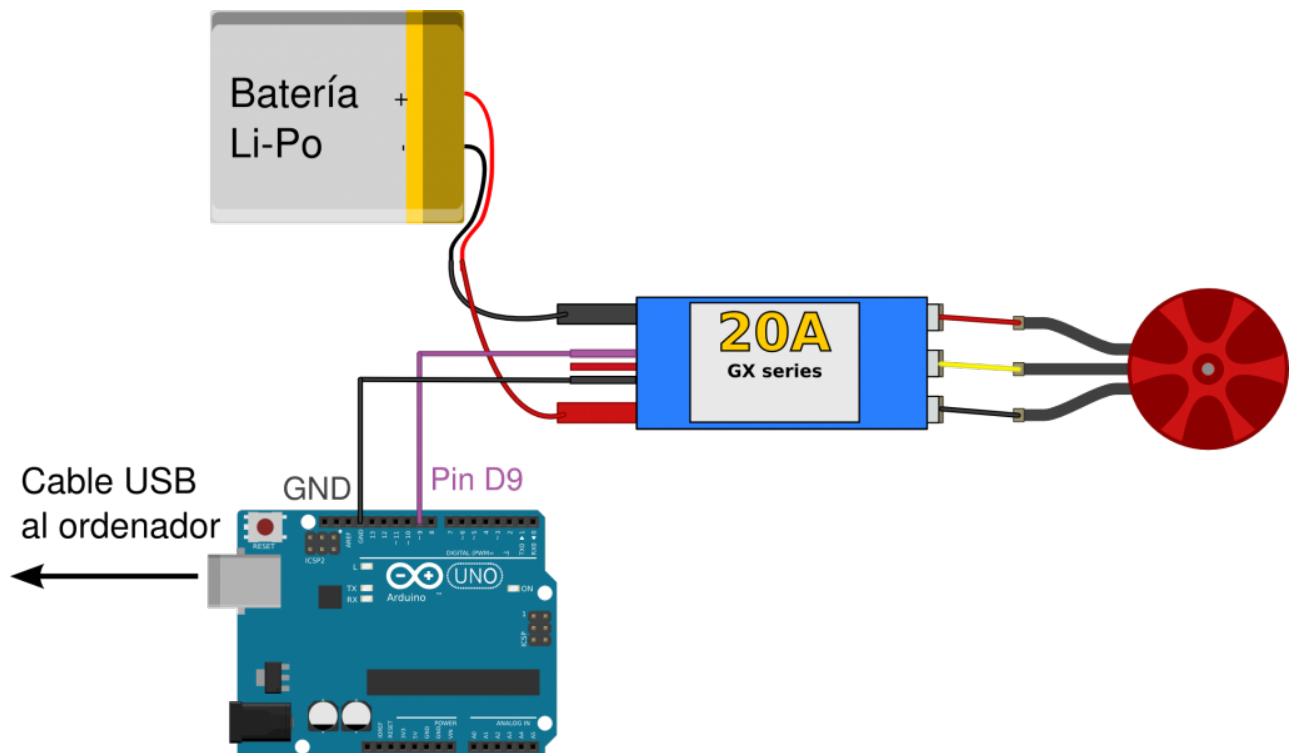


(<http://robologs.net/wp-content/uploads/2016/02/ESC.jpeg>)

Lista de piezas

- **Motor Brushless**
- **Batería LiPo/NiMH:** vigila que tenga voltaje, amperaje y velocidad de descarga para soportar el motor
- **ESC:** la razón por la que estás aquí. Yo tengo un **ESC GX de 20A**. Si vas a utilizar otro modelo, asegúrate de que pueda proporcionar amperaje suficiente para el motor brushless.
- **Placa Arduino**
- **Jumpers Macho-Macho:** para conectar el **ESC** a Arduino.
- **Regletas:** para hacer las conexiones entre el **ESC**, el motor y la batería si no quieres soldar. Necesitarás también un **destornillador**.

Conexiones



(<http://robologs.net/wp-content/uploads/2016/02/ESC.png>)

2 de 5

24/11/16 06:31

Debes tener cuidado al conectar el motor. Los motores brushless tienen 3 cables, al igual que el **ESC**. **El cable central del**

motor debe ir conectado al cable central del ESC. Sino, puedes dañar los componentes. <http://robologs.net/2016/02/01/programacion-de-...>

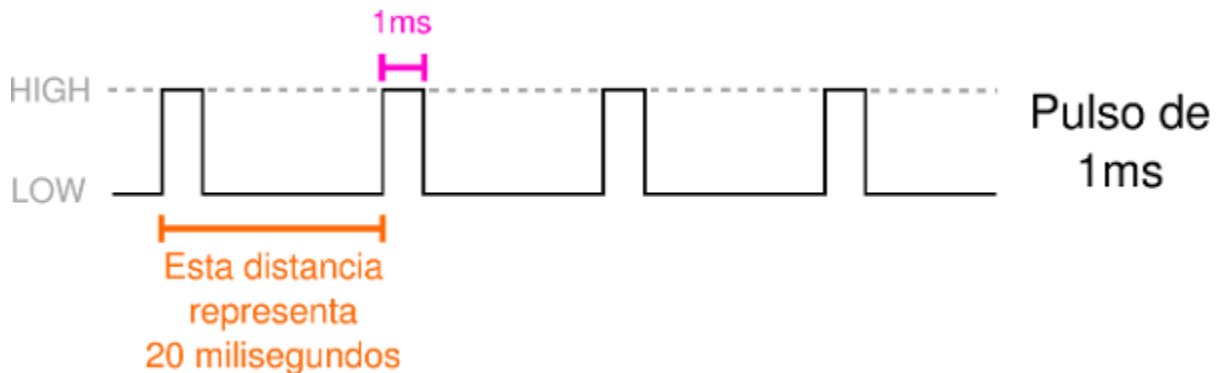
Pero no te preocupes si giras los otros dos cables (los extremos): sólo cambiará la **dirección de giro** del motor.

Rituales de activación

Los **ESC** se controlan mediante **pulsos**. Podría dedicar todo un artículo a hablar de pulsos, pero voy a dar una explicación rápida.

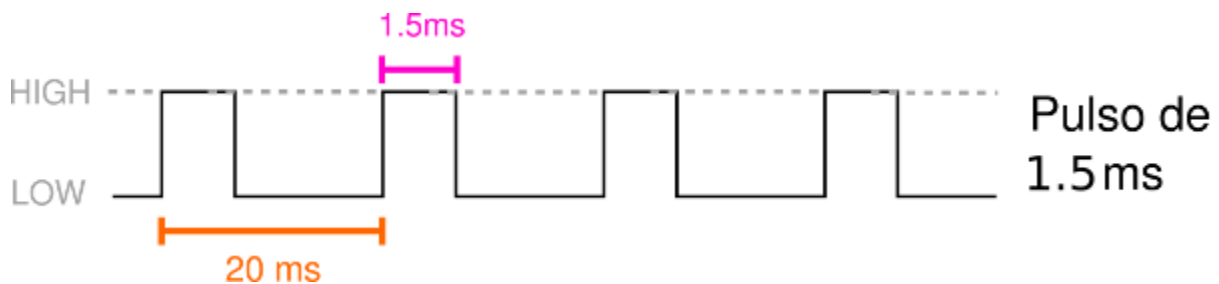
Un pulso es una señal que se envía cada cierto tiempo, normalmente cada 20ms. Si tienes un pin digital a LOW, y cada 20ms lo pones a HIGH durante un milisegundo, estás creando **un pulso de 1ms**.

Esta imagen ilustra como varía la señal de un pulso de 1ms en función del tiempo:



(http://robologs.net/wp-content/uploads/2016/02/pulso_1ms.png)

Un pulso de 1.5ms se vería así:



Los ESC se controlan con pulsos de entre 1 y 2 milisegundos. A partir de ahora, me voy a referir a los pulsos de 1ms como pulsos de **amplitud mínima** y a los de 2ms como pulsos de **amplitud máxima**.

Antes de empezar a funcionar, los **ESC** esperan recibir un **pulso de activación**. Algunos se activan si reciben un **pulso de amplitud máxima durante 2-5 segundos**, y otros con un **pulso de amplitud mínima** durante esta misma cantidad de tiempo.

Por desgracia, los **ESC** están pensados para aficionados a RC, y no a la robótica. Por eso en la documentación casi nunca se especifica con qué amplitud de pulso debe activarse el controlador. Hay antiguas leyendas que hablan de **ESCs** bien documentados, pero yo nunca he dado mucho crédito a estos mitos.

3 de 5
He aquí las “especificaciones técnicas” de mi **ESC**:

24/11/16 06:31

- Programación de un ESC con Arduino - robocogs <http://robocogs.net/2016/02/01/programacion-de-...>
1. Encienda la emisora y mueva la palanca de gas **a su posición más baja**.
 2. Conecte el motor al variador.
 3. Conecte el variador al canal de gas en el receptor.
 4. Conecte la batería de alimentación.
 5. Espere la confirmación acústica del variador.
 6. El variador está listo para su utilización.

Lo importante aquí es el **punto 1**. Los **ESCs** están pensados para funcionar con mandos de radio, y lo que este punto dice es que **para activar el ESC hay que mover el mando a su posición más baja**. Y en su posición más baja, los mandos generan un **pulso de amplitud mínima** (excepto algunos modelos, pero esto no viene al caso).

Por lo tanto, **mi ESC se activa con un pulso de 1ms**.

Si tu documentación dice que hay que mover la palanca de gas a su **posición máxima**, significa que para activar el **ESC** hay que **generar un pulso de 2ms**.

Después, **la velocidad del motor se hace variar enviando pulsos** también entre **1ms (velocidad mínima)** y **2ms (velocidad máxima)**.

Programa con Arduino

Ya sabes activar un **ESC**, humano. Ahora sólo te queda escribir un programa con Arduino para mover el motor brushless.

Los pulsos se generarán con la función **servo.writeMicroseconds()** de la librería **Servo.h**.

```

1  // Programación de un ESC con Arduino
2  /*
3   *
4   * La velocidad del motor puede cambiarse enviando
5   * un entero entre 1000 (vel. mínima) y 2000 (vel. max.)
6   * por Serial.
7   *
8   * Programa escrito por Transductor
9   * www.robologs.net
10  */
11 #include<Servo.h>
12
13 Servo ESC; //Crear un objeto de clase servo
14
15 int vel = 1000; //amplitud del pulso
16
17
18 void setup()
19 {
20     //Asignar un pin al ESC
21     ESC.attach(9);
22
23     //Activar el ESC
24     ESC.writeMicroseconds(1000); //1000 = 1ms
25     //Cambia el 1000 anterior por 2000 si
26     //tu ESC se activa con un pulso de 2ms
27     delay(5000); //Esperar 5 segundos para hacer la activacion
28
29     //Iniciar puerto serial
30     Serial.begin(9600);
31     Serial.setTimeout(10);
32
33 }
34
35
36
37 void loop()
38 {
39     if(Serial.available() >= 1)
40     {
41         vel = Serial.parseInt(); //Leer un entero por serial
42         if(vel != 0)
43         {
44             ESC.writeMicroseconds(vel); //Generar un pulso con el numero recibido
45         }
46     }
47 }

```

Recomiendo desconectar **el ESC** de la batería a la hora de hacer el upload. Justo después de cargar el programa vuelve a conectar la batería.

El motor emitirá un pitido cuando el ESC esté activado. Después puedes enviar valores enteros entre **1000** y **2000** para cambiar la velocidad.

Si te ha surgido algun problema durante el tutorial, puedes escribirme un comentario más abajo y te ayudaré a resolverla.

Final de línea.