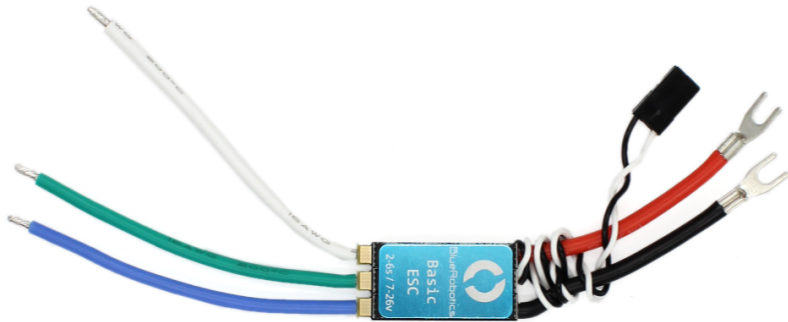


Basic ESC Documentation (New)



Introduction

NOTE: This is the new version of the Basic ESC, the documentation for old version is located here (<http://docs.bluerobotics.com/besc/>).

This new version of the Basic ESC is based on the BLHeli_S ESC design, providing an upgrade with newer technology, additional features, and improved performance. The Basic ESC is a simple brushless sensorless speed controller pre-programmed with custom firmware that allows forward and backward operation. The BLHeli_S firmware is open source and available on Github (https://github.com/bitdump/BLHeli/tree/master/BLHeli_S%20SiLabs).

Safety

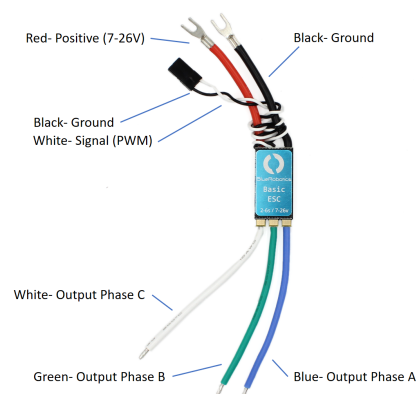
A When working with electricity, especially in water, always practice caution. Always ensure that connections are secure and watertight. Keep your body away from spinning motors and propellers.

Quick Start

1. Connect the three motor wires to the motor. The order of connections does not matter, however, switching any two wires will change the direction of the motor. The output phases A, B, and C are completely interchangeable
2. Connect the red power wire and black ground wire to a power source like a battery. You will hear three beeps in rising pitch from the motor indicating all three phases are connected.
3. Connect the signal cable to your signal source like an RC radio receiver or microcontroller board. The white wire is the signal wire.
4. Send a stopped signal (1500 microseconds) for a few seconds to initialize the ESC. You will hear two tones indicating initialization, and then you can send a signal from 1100-1900µs to operate the thruster.

Specifications

Diagram



Specification Table

Electrical		
Voltage	7-26 volts (2-6S)	
Max Current (Constant)	30 amps (depending on cooling)	
Physical		
Length	35 mm	1.38 in

Electrical		
Width	17.1 mm	0.67 in
Height	5.5 mm	0.22 in
Weight	16.3g	0.036lb
Power Connectors	Spade terminals for No. 6 screw	
Motor Connectors	Tinned wire ends	
Signal Connector	3-pin servo connector (0.1" pitch) (ground, blank, signal)	
Pulse Width Signal		
Signal Voltage	3.3-5 volts	
Max Update Rate	400 Hz	
Stopped	1500 microseconds	
Max forward	1900 microseconds	
Max reverse	1100 microseconds	
Signal Deadband	+/- 25 microseconds (centered around 1500 microseconds)	

3D Model

File Type	Link
SolidWorks Part (.sldprt)	BESC30-R3.SLDPRT (/bescr3/cad/BESC30-R3.SLDPRT)
STEP (.step)	BESC30-R3.STEP (/bescr3/cad/BESC30-R3.STEP)
IGES (.igs)	BESC30-R3.IGS (/bescr3/cad/BESC30-R3.IGS)
STL (.stl)	BESC30-R3.STL (/bescr3/cad/BESC30-R3.STL)

Installation

Thermal Considerations

Like all ESCs, the Basic ESC can generate a significant amount of heat when operated. It's important to consider this when mounting and operating the ESC to ensure that it is not damaged by overheating. Most of the heat is generated in the MOSFETs, which are underneath the blue aluminum heat spreader. Here are a few helpful tips:

1. If possible, make sure the heat spreader is exposed to free air or attached to a larger heat sink.
2. *Do not* use any adhesives that may insulate the heat generating components, such as silicone sealant.

Example Code

Arduino

This example uses the Arduino Servo library to control the speed controller. This provides an update rate of 50 Hz and can use any pin on the Arduino board as the "servoPin".

If you've never used Arduino before, we suggest checking out some tutorials! (<https://www.arduino.cc/en/Tutorial/HomePage>)

Note: If you power the Arduino before powering the ESC, then the ESC will miss the initialization step and won't start. Power them up at the same time, power the ESC first, or press "reset" on the Arduino after applying power to the ESC.

```
#include <Servo.h>

byte servoPin = 9;
Servo servo;

void setup() {
  servo.attach(servoPin);

  servo.writeMicroseconds(1500); // send "stop" signal to ESC.
  delay(1000); // delay to allow the ESC to recognize the stopped signal
}

void loop() {
  int signal = 1700; // Set signal value, which should be between 1100 and 1900

  servo.writeMicroseconds(signal); // Send signal to ESC.
}
```

Advanced

Firmware Files

The firmware configuration settings file for BLHeliSuite can be downloaded here:

BR Basic ESC BLHeli_S (/bescr3/cad/BLHeli_BR%20Basic%20ESC%20-%20R_H_15%20-%20Rev.%2016.6%20-%20Multi_170921.ini)

Detailed documentation for BLHeli_S operation and settings can be downloaded here:

BLHeli_S Version 16.x manual (/bescr3/cad/BLHeli_S%20manual%20SiLabs%20Rev16.x.pdf)

Firmware Update and Customization

The Basic ESC uses version 16.6 of the BLHeli_S firmware (https://github.com/bitdump/BLHeli/tree/master/BLHeli_S%20SiLabs) which is open source and editable. There are many parameters that can be changed to alter the performance of the speed controller by flashing the ESC through the PWM signal wire using a programming tool like the Turnigy USB Linker (http://www.hobbyking.com/hobbyking/store/__10628__turnigy_usb_linker_for_aquastar_super_brain.html), the AfroESC Programmer (http://www.hobbyking.com/hobbyking/store/__39437__afro_esc_usb_programming_tool.html), or an Arduino. The BLHeliSuite (<https://blhelisuite.wordpress.com/>) software can be used for accessing the ESC firmware and configuring it.

The Basic ESC uses an SiLabs microcontroller and is Oneshot 125 capable.
