Instructions for making and flashing ESC firmware

Firmware Update and Customization

The Basic ESC uses the tgy firmware which is open source and editable. There are many parameters that can be changed to change the performance of the speed controller.

Firmware Compilation

To compile the firmware, you'll need the avra AVR Assembler.

Mac: (Uses Homebrew)

```
brew update
brew install avra
make blueesc.hex
```

To compile the files with multiple I2C addresses, you can use the following:

```
make build_blueesc_addresses
```

Linux (Ubuntu 14 LTS):

```
sudo apt-get install avra
git clone https://github.com/bluerobotics/tgy
cd tgy
make blueesc.hex
```

(This part has already been done)

Firmware Flashing

Open the Arduino IDE and open preferences. Next to "show verbose output during:", check "upload".

Upload any sketch to an Arduino (over USB is fine).

Scroll through the output and find the avrdude command. It should be in white text and look similar to this:

"/Applications/Arduino.app/Contents/Java/hardware/tools/avr/bin/avrdude -C/Applications/Arduino.app/Contents/Java/hardware/tools/avr/etc/avrdude.conf -v -patmega328p -carduino -P/dev/cu.usbmodem14101 -b115200 -D -Uflash:w:/var/folders/q8/zc_5dcx92bv9r3htwd2_g9sw0000gn/T/arduino_build_579764/Blink.ino.hex:i"

"/Applications/Arduino.app/Contents/Java/hardware/tools/avr/bin/avrdude" is the path for the

avrdude executable.
"-C/Applications/Arduino.app/Contents/Java/hardware/tools/avr/etc/avrdude.conf" defines the path for the avrdude config file.

Whenever you want to use avrdude, paste these two paths into the terminal, then enter the other necessary flags.
By calling avrdude in this way, you can use it the way the Arduino IDE uses it, without having to install it yourself.
Refer to the avrdude documentation for using flags. (http://www.ladyada.net/learn/avr/avrdude.html)

The other flags needed to flash the firmware are:

```
-p <partno>
-c <programmer>
-P <port>
```

When using the (programmer), these should be set to:

```
-p atmega8
-c usbasp
-P USB
```

The option to flash a hex file to the target is:

```
-U <memtype>:r|w|v:<filename>[:format]
```

Example:

```
$ /Applications/Arduino.app/Contents/Java/hardware/tools/avr/bin/avrdude -C/Applications/
Arduino.app/Contents/Java/hardware/tools/avr/etc/avrdude.conf -v -p atmega8 -c usbasp -P usb -
U lfuse:w:0x3F:m -U hfuse:w:0xCA:m -U lock:w:0x3f:m -Uflash:w:/Users/matt/tgy-bluesc/
blueesc_id0.hex:i
```

Choose id0 for ESC 1, id1 for ESC2, etc.