ESC Documentation

The RPI sends pulse width modulation (PWM) signals proportional to the demanded rotor speed. The electronic speed controller (ESC) converts these signals to DC voltage, which is then sent to the motor. Two thick wires, Vin and GND, take in the battery power. Three output wires deliver power to the motor. These output wires are unlabeled. If the motor spins the wrong way, switch any two of the wires and it will reverse the motor direction. There are three smaller wires that serve as the controls. The white wire carries the PWM signal, and it should be connected to one of the GPIO pins that allows PWM operation (we currently are using 18, 23, 13, and 21. The black wire is ground. The red wire (labelled X below) should not be used. Clip it and tuck it away.

A drawing of a circuit board

Description automatically generated

The ESC has internal firmware that requires a specific signal to “arm” the unit. Before it is armed, it will not pass any signal to the motors (this is a safety feature). The arming procedure is implemented in the main code automatically. To arm the drone, the relay is connected, and the PWM values are set to minimum, and the code is paused for several seconds. You’ll hear 3 short beeps and 1 long beep. At this point, the drone is armed, and you’ll be prompted to press any key to continue.

Different ESCs have different calibration and arming procedures, so be weary. When use new ESCs (even the same make/model), you may have to work out the calibration procedure. For our ESCs that should be something like the following:

1. Disconnect power to ESC
2. Set PWM to high
3. Connect power
4. Wait
5. Set PWM to low
6. Wait

But we don’t currently have a working script for this. Fortunately, the calibration appears to be more important for a remote-control application, where the actuator may be tuned incorrectly. The direct control from the RPI does not appear to need calibration, since the PWM values are always between 1100 and 1900.

The code pertaining to the ESC is in ESC.py. It uses the libraries RPi.GPIO and pigpio. Note: every time you open a new ssh session, you’ll have to run the command sudo pigpiod before you can run these commands. Otherwise you’ll get an error reminding you. There are several functions in ESC.py:

* Init: initializes and configures the PWM pins to control the motors
* connectMotorsPigpio: runs the arming procedure
* writeMotors: writes the desired PWM values to the ESC
* StopMotors: writes the minimum PWM value to the ESC and trips the relay. Either of these two actions is sufficient to make the motors stop.