

SuperFly ESP8266/EM7180-Based Integrated Flight Controller

By Greg Tomasch

This document provides the background information necessary to set up, calibrate and successfully use the [SuperFly integrated flight controller](#) for very small brushed-motor quadcopters powered by 1S Lipo batteries.

Flight Controller Board and Software Capability Overview

The [SuperFly integrated flight controller](#) uses the Expressif [ESP8266](#) WiFi-enabled microcontroller unit (MCU) and EM Microelectronic's EM7180 "Sentral" register-based sensor fusion coprocessor. Accelerometer, gyroscope and magnetometer sensing is provided by Invensense's MPU9250 integrated MEMS sensor. Finally, barometric altitude is estimated using Measurement Specialties' MS5637 temperature-compensated barometric pressure sensor. The SuperFly board also provides four channels of on-board low on-resistance (20 mOhm) n-type [DMN2041L](#) MOSFET motor drivers suitable for 1S LiPo 6-8mm brushed DC motors, making it an outstanding solution for flight control for tiny multirotors approximately 125mm in size (~50 g weight) or smaller.

The flight controller software for this board is written in "Arduino C++" and is intended to be open source and fully visible to the user. The basic flight mode architecture, radio control (RC) signal handling, serial interface and PID loop architecture are heavily borrowed from the MultiWii open source flight controller project. However, this particular piece of software is written and specifically optimized to take full advantage of the EM7180 co-processor. The sensor fusion and inertial measurement unit (IMU) portions of the code bear little resemblance to the original MultiWii. Uploading the Arduino flight control sketch is accomplished using an "FTDI" 3.3V logic USB to serial converter (<https://www.sparkfun.com/products/9873>) or similar. Setup, configuration, calibration and diagnosis of the SuperFly flight controller are accomplished by an HTML/JavaScript GUI application running under Google Chrome and using the same USB to serial converter. This application is a minor variant of the "Configurator" application written for the Baseflight open source flight controller project with additional elements borrowed from the Cleanflight open source project as well. The GUI application has been optimized to serve the particular needs of EM7180-based flight controllers.

The combination of the ESP8266 and the EM7180 is capable of providing Euler angle stabilized and other enhanced performance flight modes. The simplest multirotor flight controllers only use gyroscopes for flight stabilization. This has the effect of holding the craft in the last attitude that the pilot commanded, irrespective of gravity. That is, the craft will not be self-righting and will not return to level attitude if the flight controls are set to neutral. In general, it is also true that the flight controls reverse if the "Nose" of the craft is pointed toward the pilot instead of the "Tail".

By adding accelerometers and magnetometers to the gyroscope sensors, the EM7180 is able to accurately estimate the absolute orientation state of the quadcopter. This allows the craft to be self-righting; returning the flight controls to neutral and the craft will return to level attitude. The EM7180 is also capable of accurately estimating compass heading within approximately $\pm 2^\circ$, even in regions of high magnetic distortion. The heading estimation algorithm does a superior job of compensating for non-ideal magnetometer response and transient magnetic interference. This enables “Headless” flight mode; that is, no matter how the front of the quadcopter is pointed, the flight controls do not reverse. This is also referred to as “Carefree” operation. Although Euler angle stabilization and carefree flight capabilities are available on flight controllers that are designed for much larger and more expensive multirotor craft, it has been a challenge to duplicate similar results for very small quadcopters using brushed DC motors.

The SuperFly flight controller is also equipped with an enhanced automatic altitude control algorithm. Automatic altitude control of multirotor helicopters using only barometric pressure sensors is difficult to achieve. Altitude resolution tends to be poor, sensor response time is slow and baseline drift of several meters over 10 – 20min is common. The SuperFly flight controller utilizes a novel algorithm to fuse double-integrated vertical linear acceleration with barometric altitude estimates to provide enhanced automatic altitude control. The fused composite altitude control signal possesses greatly improved response time and significant noise reduction compared to barometer-only solutions. Furthermore, baseline drift of the pressure-based altitude signal is largely compensated by a carefully designed bi-quad high-pass digital filter. Although tuning the altitude control loop requires some degree of patience, altitude hold accuracy and stability is adequate to use indoors with conventional ceiling height. Now it is fair to say that a 125mm quadcopter flies as well as ~450mm (or larger) craft costing much more.

Radio control of the SuperFly flight controller can be accomplished using a six-channel, pulse-position-modulation (cPPM) receiver or the embedded WiFi capability. The OrangeRX R514X receiver in conjunction with a six channel Japan Radio (JR) DSM2/DSMX protocol RC transmitter is an excellent choice (http://www.hobbyking.com/hobbyking/store/_47012_OrangeRx_R415X_DSMX_DSM2-Compatible_4Ch_Micro_2_4Ghz_Receiver.html). However, in principle any six channel 3.3V logic cPPM RC system should work.

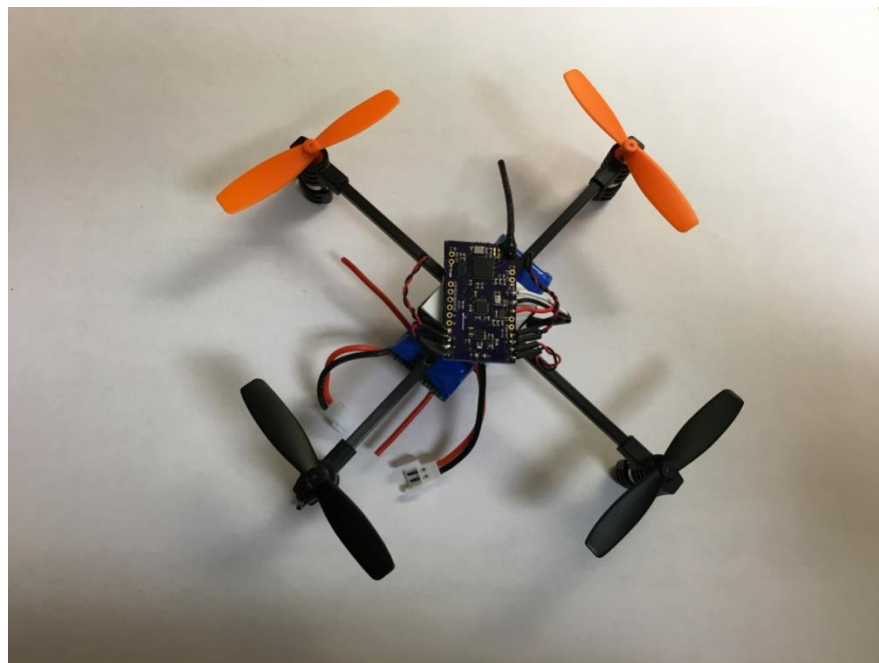
The integrated WiFi capability is used in two possible modes: 1) Radio Control Over Internet Protocol (RCOIP) transceiver for flight control and 2) Debug telemetry to transmit real-time flight data to a ground station PC. The debug telemetry feature is a stand-alone utility at this time. It is not integrated into the RCOIP radio control mode and can only be used as a diagnostic when the cPPM receiver option is chosen. If the WiFi RCOIP option is selected, the SuperFly flight controller will receive RC commands from an iOS device (iPhone or iPad) using the RCTx app (<https://itunes.apple.com/us/app/rctx/id567423127?mt=8>). The SuperFly flight controller configures its WiFi transceiver as a wireless access point that will appear on the WiFi setup menu of the iOS device.

General Construction Considerations, Flight Controller Connections and Airframe Integration

This section will cover the integration of the flight controller board into a simple X-copter and document the key configuration aspects of the airframe. The motor placement and propeller direction conventions must be followed for successful flight stabilization. It should also be noted that the rotation direction for the brushed DC motors indicated by the motor manufacturer should be observed for maximum motor life.

The SuperFly integrated flight controller is simple to mount and connect to a quadcopter airframe. Fig. 1 shows the board installed onto a UDI RC U816 125mm X-copter frame.

Figure 1. A 125mm X-copter built using the SuperFly integrated flight controller and airframe from a UDI RC U816 quadcopter. This particular craft uses the cPPM receiver option. An OrangeRX R415X receiver is mounted to the underside of the battery holder.



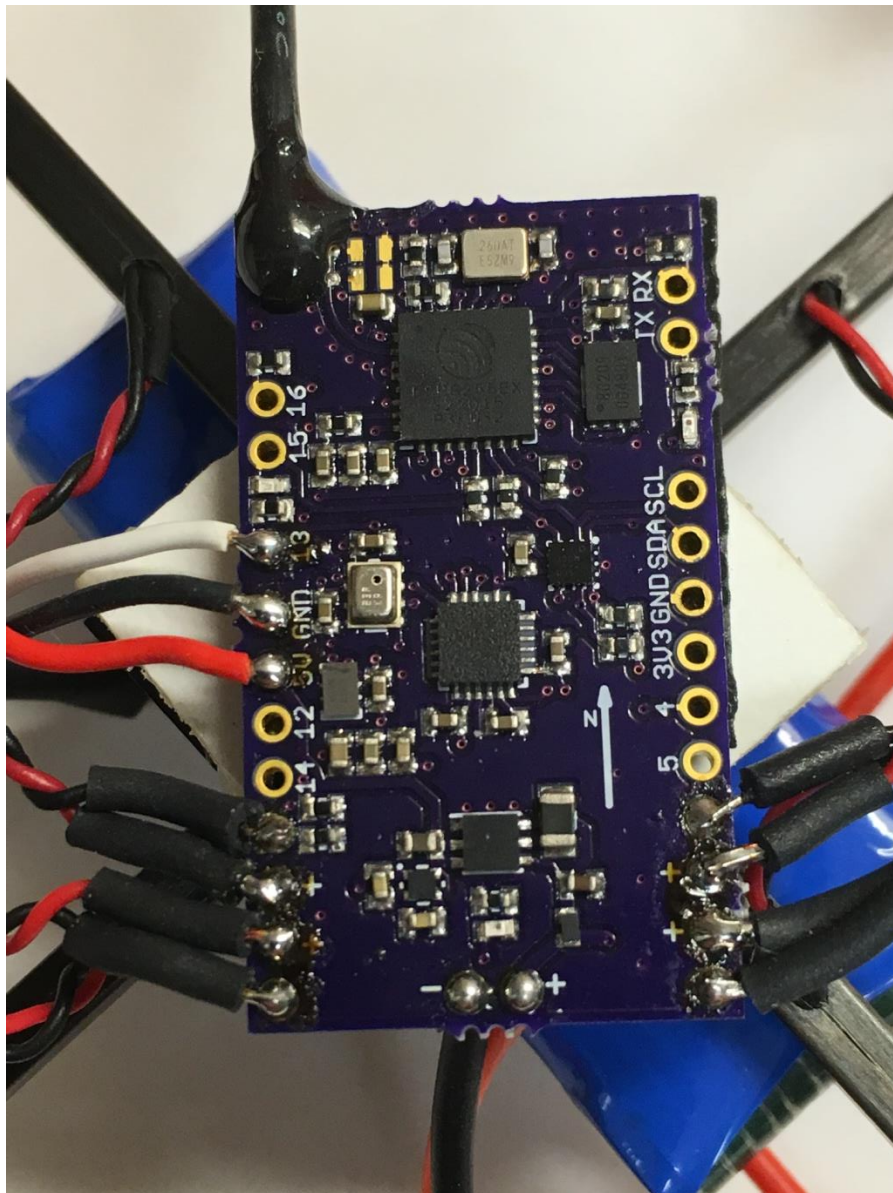
All that is required to mount the flight controller board is a relatively smooth and flat surface that is parallel to the plane of the rotors. Typically, double-sided foam tape is used to attach the board to the copter frame. Multiple layers can be used to provide additional leveling and electrical isolation from metal fasteners.

(http://solutions.3m.com/wps/portal/3M/en_US/NAHomeEnergy/Home/Products/~3M-Outdoor-Mounting-Tape?N=7579606+3294348381+3294529207&rt=rud). Double sided foam tape is convenient because it provides secure but flexible mounting that can easily be changed or reworked in the future. Molex type two-connector power plugs are convenient because many 1S LiPo batteries come equipped with the mating connector. **WARNING! BE ABSOLUTELY CERTAIN THAT THE POSITIVE POLE OF THE 1S LiPo BATTERY WILL BE**

CONNECTED TO THE POSITIVE POWER PAD OF THE FLIGHT CONTROLLER!

Connecting the flight controller to the battery with reverse polarity will cause permanent damage! Other style battery connectors can be used as well so long as the polarity convention is correct. Fig. 2 shows a close-up view of the SuperFly flight controller attached to the airframe.

Figure 2. A close-up view of the SuperFly flight controller board mounted to a UDI RC U816 airframe. Note that the arrow is pointing to the front of the craft.

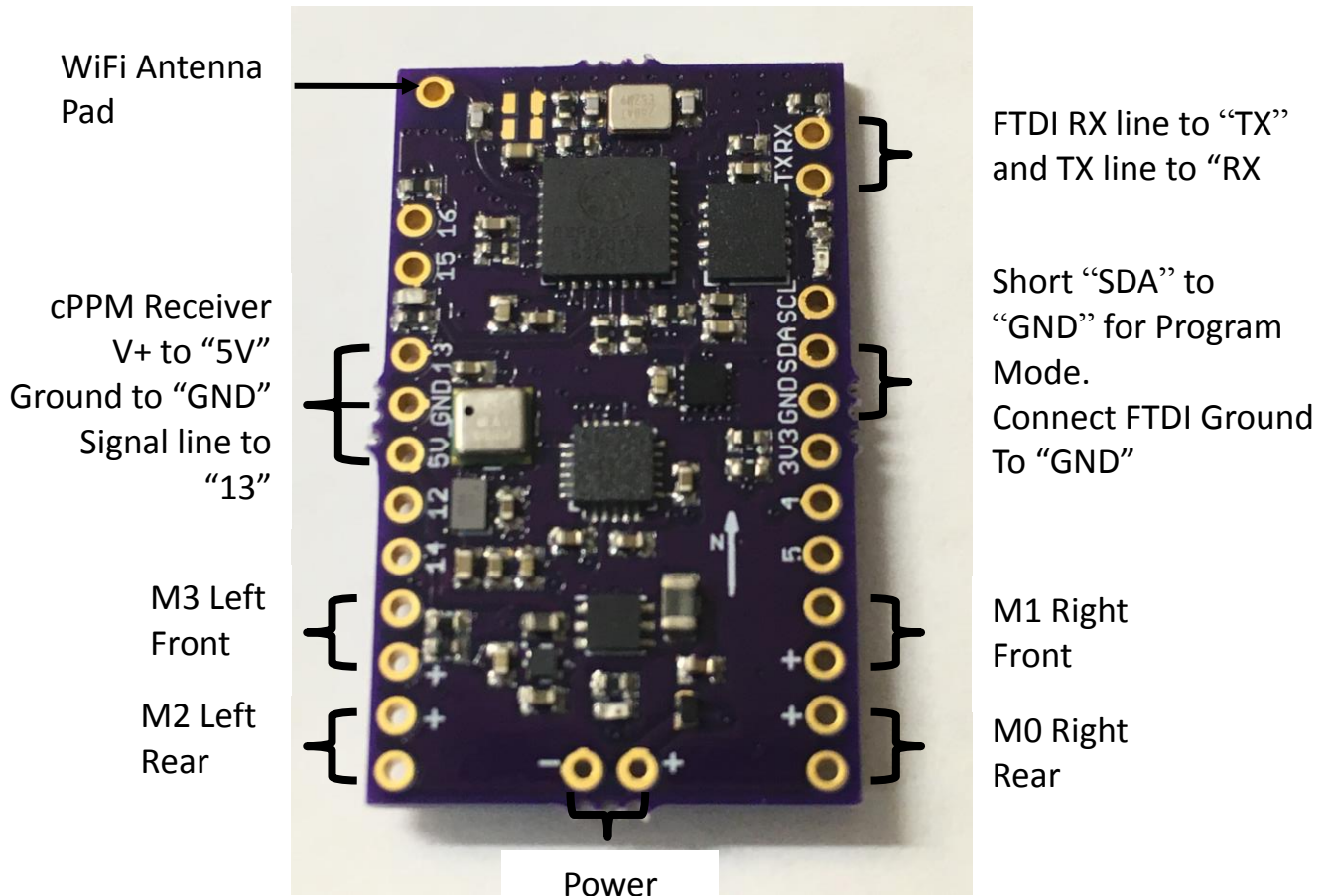


This particular craft uses the OrangeRx R415X cPPM receiver option. The R415X is mounted on the underside of the battery holder for convenience and balance. The V+ node of the receiver (red wire) is connected to the “5V” pad, the ground node of the receiver (black wire) is

connected to the “GND” pad and the cPPM signal node of the receiver (white wire) is connected to the “13” pad on the upper left side of the SuperFly board. It should also be noted that a 1.25” long wire is soldered to the WiFi antenna pad. This wire segment serves as the WiFi transceiver antenna. The motor lead connections are intended to be intuitive; the right front pair of connections goes to the right front motor, etc. It is highly recommended that the motor leads be twisted together. In general, keeping the separation between positive and negative power leads to a minimum will minimize stray magnetic fields.

Fig. 3 summarizes the electrical connections to the SuperFly flight controller board. In general, three connections from the “FTDI” USB-serial converter are required: 1) Transmit or “TX” 2) Receive or “RX” and 3) Signal Ground or “GND”.

Figure 3. Connection detail for the SuperFly flight controller board

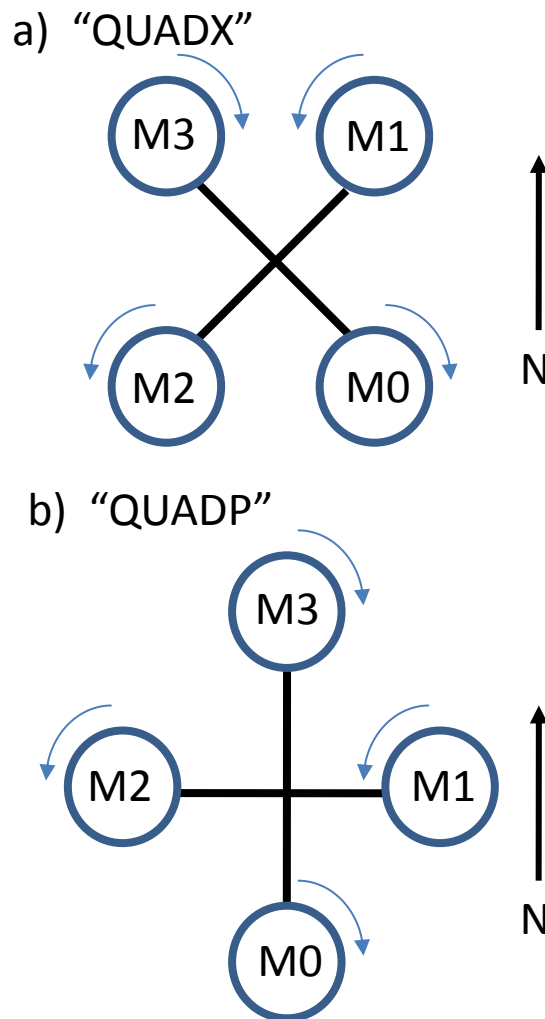


To establish serial communication with the SuperFly, the “TX” of the FTDI must be connected to the “RX” of the SuperFly and similarly the “RX” of the FTDI must be connected to the “TX” of the SuperFly. The “GND” nodes of the FTDI and SuperFly must be connected to so that both boards share the same signal reference. **WARNING! FTDI AND OTHER USB-SERIAL CONVERTERS COME IN 5V LOGIC AND 3.3V LOGIC VERSIONS. THE SUPERFLY**

IS DESIGNED TO WORK WITH THE 3.3V LOGIC VERSION ONLY. USING A 5V LOGIC USB-SERIAL CONVERTER WITH THE SUPERFLY BOARD MAY RESULT IN PERMANENT DAMAGE! Finally, to upload the Arduino flight controller sketch into the SuperFly board the “SDA” and “GND” nodes must be shorted together prior to powering up the SuperFly board. This will put the ESP8266 MCU into the correct mode to accept firmware upload from the Arduino IDE.

The flight controller software assumes certain conventions with respect to the airframe configuration in order to properly calculate aircraft attitude and properly route power to the four motors of the quadcopter. Fig. 4 shows the correct motor placement and rotation for both the “QUADX” (X-copter) and “QUADP” (Plus-copter) configurations. The SuperFly flight control sketch is capable of supporting either airframe configuration.

Figure 4. Proper motor placement and propeller rotation for a) “QuadX” and b) “QuadP” airframe configuration



The “N” direction arrow indicates that the similar arrow on the SuperFly board (Figs. 2,3) should point to the front of the craft when the board is properly installed.

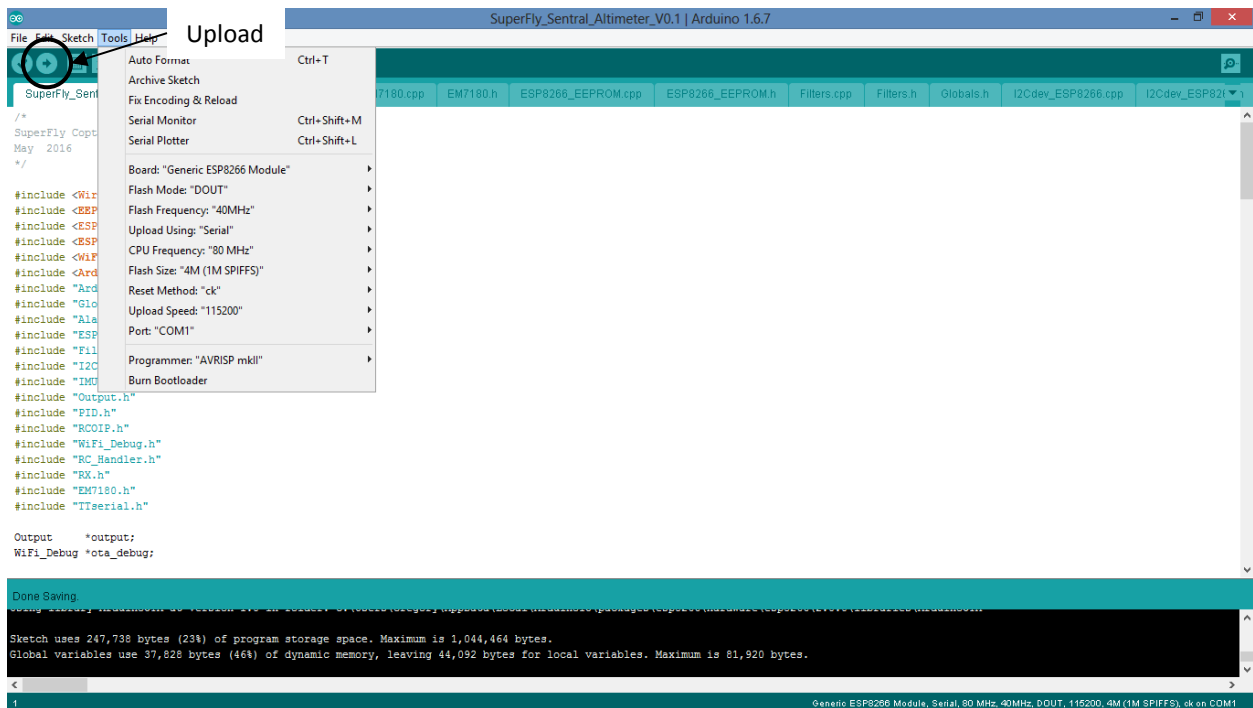
Setting Up the Arduino IDE for the SuperFly Board and Configuring/Uploading the Sketch

This section covers configuring the Arduino IDE to compile and upload the SuperFly flight control sketch. Basic configuration of the flight control sketch for specific use cases is also discussed

The key to the SuperFly’s superior flexibility is that the flight controller software is composed in the user-friendly Arduino IDE. However, to use Arduino to build executable code for the ESP8266 MCU, it is necessary to install the correct processor-specific core and library functions into Arduino. The ESP8266 Arduino core can be found at: <https://github.com/esp8266/Arduino> and the specific installation instructions are located at: <https://github.com/esp8266/Arduino#installing-with-boards-manager> .

Fig. 5 is a screen shot of the Arduino IDE “Tools” menu showing the correct board configuration for the SuperFly flight controller.

Figure 5. The Arduino IDE “Tools” menu after installation of the ESP8266 core and configuration for the SuperFly board.



The proper board configuration variables are:

Board: "Generic ESP8266 Module"

Flash Mode: "DOUT"

Flash Frequency: 40MHz"

Upload Using: "Serial"

CPU Frequency: "80MHz"

Flash Size: "4M (1M SPIFFS)"

Reset Method: "ck"

Upload Speed: "115200"

Programmer: "AVRISP mkII"

The port should be set to the "COM#" designation corresponding to the USB-serial converter being used for board programming.

At this point, your Arduino IDE installation should be properly configured to support uploading the flight controller sketch. However, the exact flight controller configuration should be checked prior to upload. The vast majority of the configuration should be correct as supplied. However, there are few items that may need to be changed depending upon your airframe configuration and your location. The following is a brief discussion of which configuration parameters are safe to edit and what values are permissible:

- Navigate to the "config.h" tab of the sketch
- Under the "Type of Multicopter" section the standard value is to define "QUADX" and comment out all other options with "//". If you are installing the SuperFly on a "QUADP" airframe (Fig. 4) uncomment this option and comment out the "QUADX" option
- Under the "Radio Control Receiver" section uncomment "#define PPM_RX" if you are using a cPPM radio control receiver. If you wish to control the craft using an iOS device (e.g iPhone or iPad) uncomment "#define RCOIP_RX" and comment out the "PPM_RX" option. Only one receiver option can be active, the other must be commented out
- Under the "Magnetic Declination" section update the magnetic declination for your location. It is not absolutely necessary to do this but it is helpful for the magnetometer/heading calibration. The current magnetic declination information for your location can be obtained at <http://www.magnetic-declination.com/>
- As a final note, the last section of the "config.h" file gives the option to activate the "WIFI_DEBUG" telemetry function. This is currently an advanced troubleshooting utility that works under the "PPM_RX" receiver option. It transmits four debug variables

from the SuperFly integrated flight controller to an auxiliary ESP8266 board via WiFi transceiver link. Additional information on its setup and use will be included in future releases

- Other changes to “config.h” entries are not advised except for advanced users with specialized needs
- Once any desired changes are made, save the SuperFly sketch

In order to upload the flight controller sketch to the SuperFly board:

1. Connect the USB-serial “TX”, “RX”, and “GND” lines to the SuperFly board “RX”, “TX”, and “GND” as described in the previous section
2. Connect the “SDA” pad to the “GND” pad (Fig. 3) before powering up the board to put the ESP8266 in BOOT mode
3. Power up the board and press the upload button on the Arduino IDE (Fig.5)
4. Once the sketch has been uploaded, remove the jumper from the “SDA” pad to the “GND” pad. The SuperFly flight controller board should be ready for use

Successful upload of the flight controller sketch will be indicated by the “Done Uploading” message at the top of the message bar window at the bottom of the Arduino IDE screen.

Initial SuperFly Functional Testing and the Google Chrome Configurator GUI

There are several key topics covered in this section:

- Basic functional testing of the SuperFly board after uploading the flight control sketch
- Installation of the SuperFly configurator GUI Google Chrome APP
- Detailed descriptions of the items included in each category tab
- Explanation of how configurator GUI items influence the behavior of the SuperFly flight controller and the associated quadcopter

This is a wide-ranging set of topics but many key concepts of the various flight modes and features are more practically understood in the context of the configurator GUI App.

Basic Functional Testing

Once the SDA/GND jumper has been removed, power up the board. If the flight controller is working properly, you should see the blue LED flash a few times on power-up. Let the board sit still for a few seconds after applying power and the initial flashes of the blue LED have stopped. One feature of the SuperFly is that the blue LED will blink if the board is tilted more than 25° from level in either the pitch or roll axes. As a gross test of flight controller functionality, pick up the board and tilt it > 25° in pitch/roll and verify that the blue LED blinks. This indicates that the sensors and EM7180 are functioning properly and that valid Euler angles are being calculated.

Configurator GUI application Installation

For full testing and calibration of the SuperFly it is necessary to connect it to the “Configurator” GUI application. The configurator application should be included with the SuperFly Arduino sketch located in a folder named “Configurator-SuperFly” If the configurator files are compressed, extract them. Open an instance of Google Chrome and go to the:

“Tools->More tools->Extensions”

menu as shown in Fig.6. The Chrome “Extensions” page will appear. Make sure the “Developer Mode box is checked. Once in the “Extensions” menu press the “Load unpacked extension...” button, select the “Configurator-SuperFly” folder in the pop-up dialog box and click the “OK” button as shown in Fig 7.

Figure 6. Navigation to the “Extensions” menu of Google Chrome

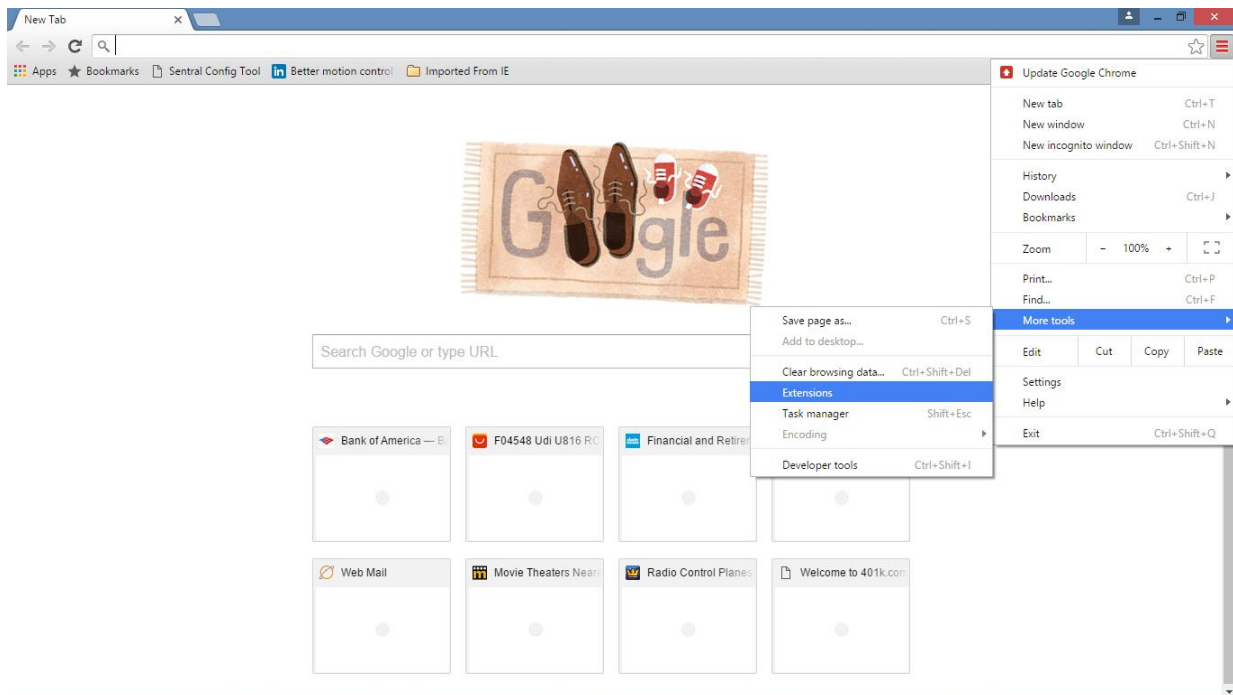
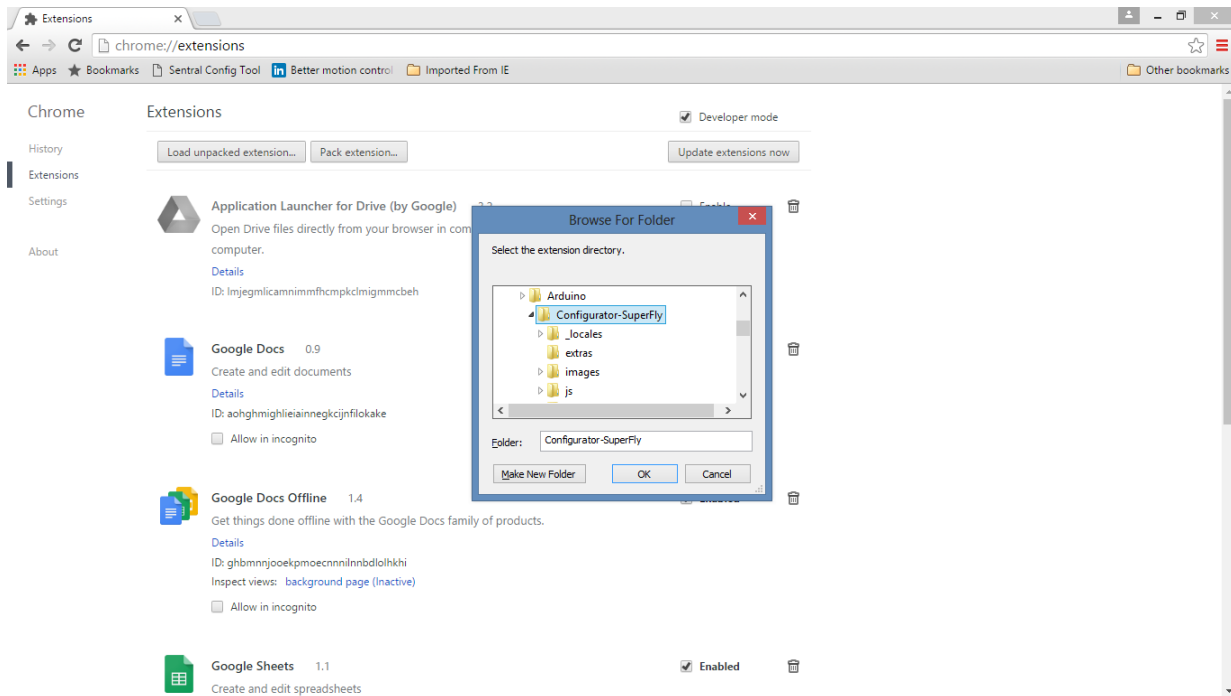
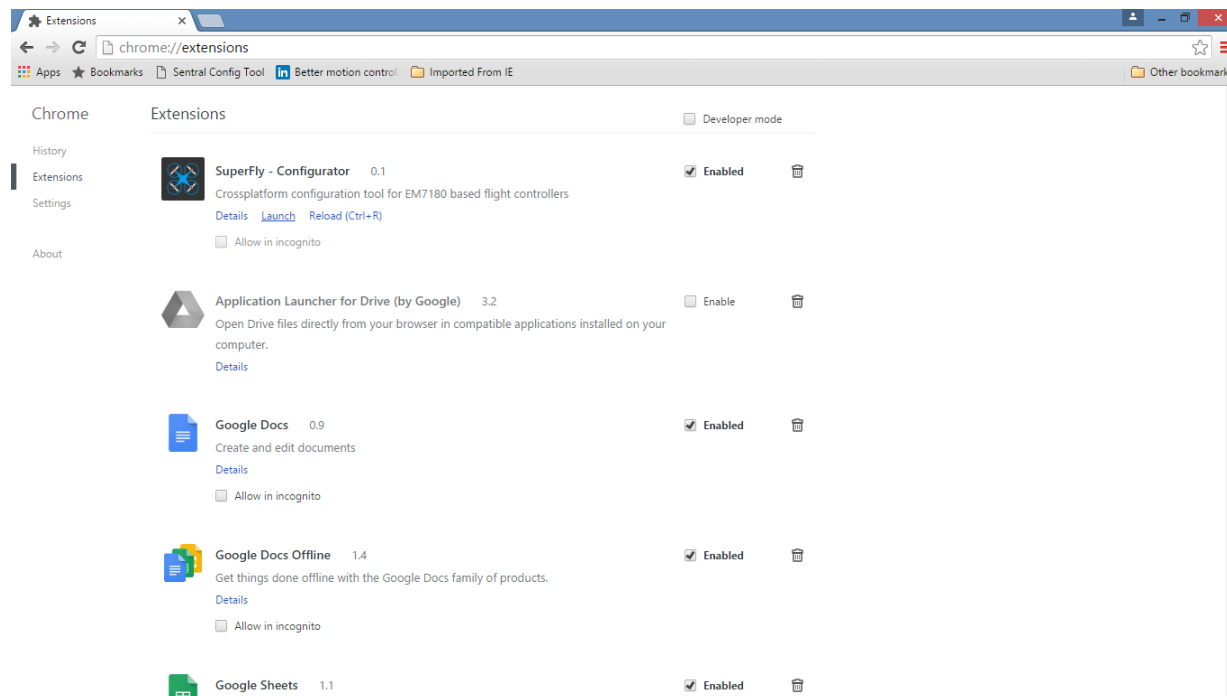


Figure 7. Click the “Load unpacked extensions...” button, select the “Configurator-Superfly” folder and click the “OK” button to install the configurator App.



Google Chrome will add the App to the “Chrome Extensions” page as shown in Fig. 8. The SuperFly configurator GUI App is now properly installed and can be launched from the “Extensions” menu of Google Chrome by clicking the “Launch” link shown in Fig. 8. It can also be launched from the “Chrome App Launcher” shortcut on the Windows desktop. The “Developer Mode” box can be unchecked at this point. It is unclear that this is essential but sometimes Chrome will query to turn off developer mode, indicating that unstable results may occur...

Figure 8. The SuperFly Configurator can be started by clicking the “Launch” link. Developer mode may be de-selected at this point...



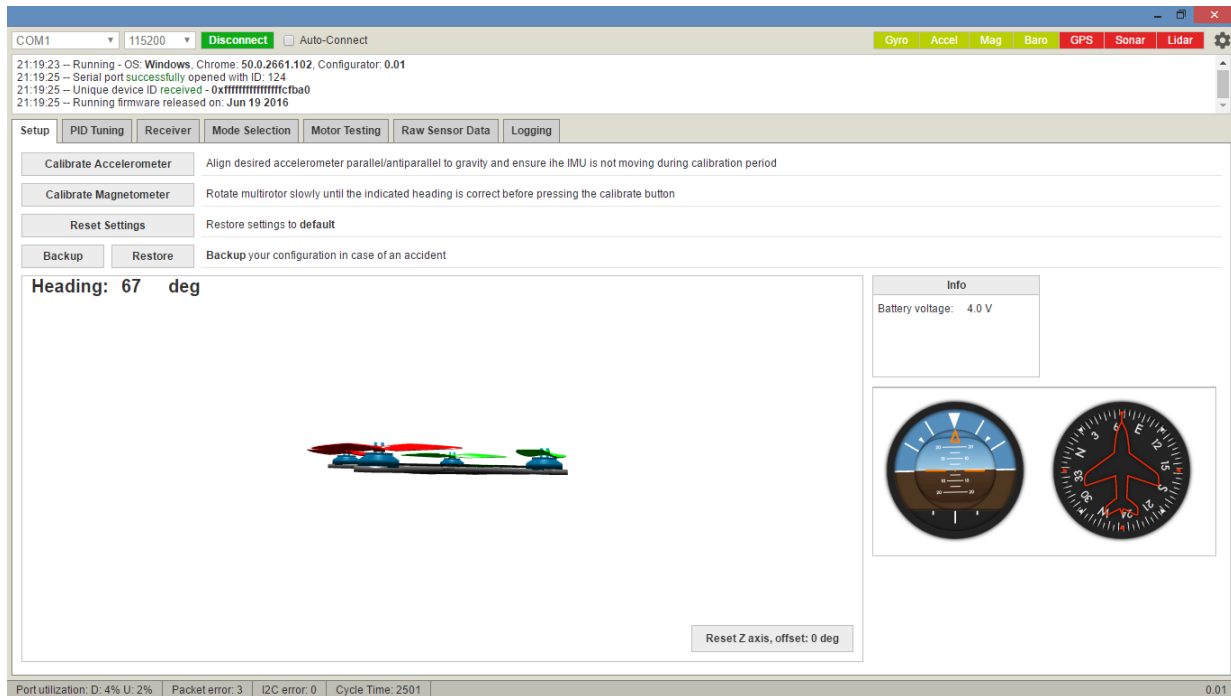
Configurator GUI Overview and Field-by-Field Explanation

Launch the SuperFly configurator App at this time. Ensure the USB-serial converter is not connected to the SuperFly Board, re-connect it now and power-up the board. You should see the screen in Fig. 9 and you should also see the correct “COM#” designation for your USB-serial converter (COM1 in this instance) in the uppermost left dropdown menu of the configurator. Once you are certain the correct “COM#” port is selected, press the “Connect” button. The “Setup” tab will be displayed on the screen and you should see the 3D quadcopter icon appear in the lower left center portion of the screen. If you tilt/rotate the SuperFly board you should see the 3D icon respond to attitude changes as well as the artificial horizon and compass dial on the right side of the display. The “Info” block has some features disabled due to a lack of relevant sensors but the “Battery voltage” is supported.

The control buttons in the upper left portion of the “Setup” tab are very important for sensor calibration and SuperFly configuration management. The “Calibrate Accelerometer” button executes the necessary data collection and EEPROM storage to properly compensate for scale and offset errors in the X, Y, and Z accelerometers. When valid accelerometer calibration data is stored in the SuperFly’s EEPROM, it is automatically loaded into the EM7180 at boot-up and corrections are applied to the accelerometer raw data. Although this appears similar to the accelerometer calibration function for MultiWii-family flight controllers, it is actually radically

different because of the architecture of the EM7180 coprocessor. The proper procedure for calibrating the SuperFly’s accelerometers will be discussed in detail in a later section.

Figure 9. Configurator GUI App Successfully connected to the SuperFly flight controller using a USB-serial converter on COM1, showing the “Setup” tab view

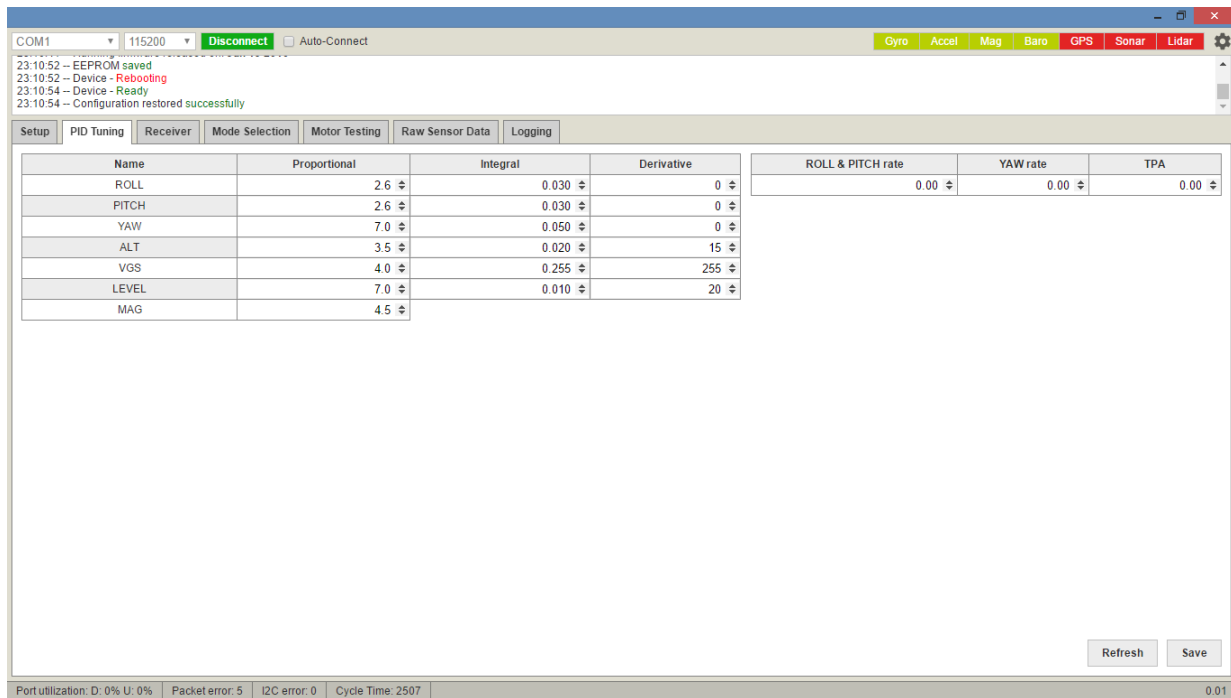


The “Calibrate Magnetometer” button is used to store the current state of the EM7180’s heading estimation algorithm once it has sampled sufficient 3D orientations to properly compensate for magnetometer sensor calibration errors and hard/soft iron field distortion. As with the accelerometers, once the algorithm has determined the correct compensation, it applies the corrections to the raw magnetometer data. If valid algorithm constants are stored in the SuperFly’s EEPROM, they are automatically loaded into the EM7180 at boot-up. A detailed procedure for properly “Teaching” the heading estimation algorithm will be discussed in detail in a later section.

The three remaining buttons are for configuration management. The “Reset Settings” button will restore the configuration and tuning parameters to the defaults in the SuperFly flight controller sketch. The “Backup” button will open a file dialog box and allow saving the current configuration of the SuperFly to backup file on the PC. Similarly, the “Restore” button allows the retrieval of a previous backup configuration and re-loads it into the SuperFly.

Fig. 10 shows the screen associated with the “PID Tuning” tab. This view allows editing of the “PID” gain tuning constants for the various control loops within the SuperFly.

Figure 10. The “PID Tuning” tab of the SuperFly configurator App Showing the current PID loop gain values in the SuperFly’s EEPROM.



The control loops include:

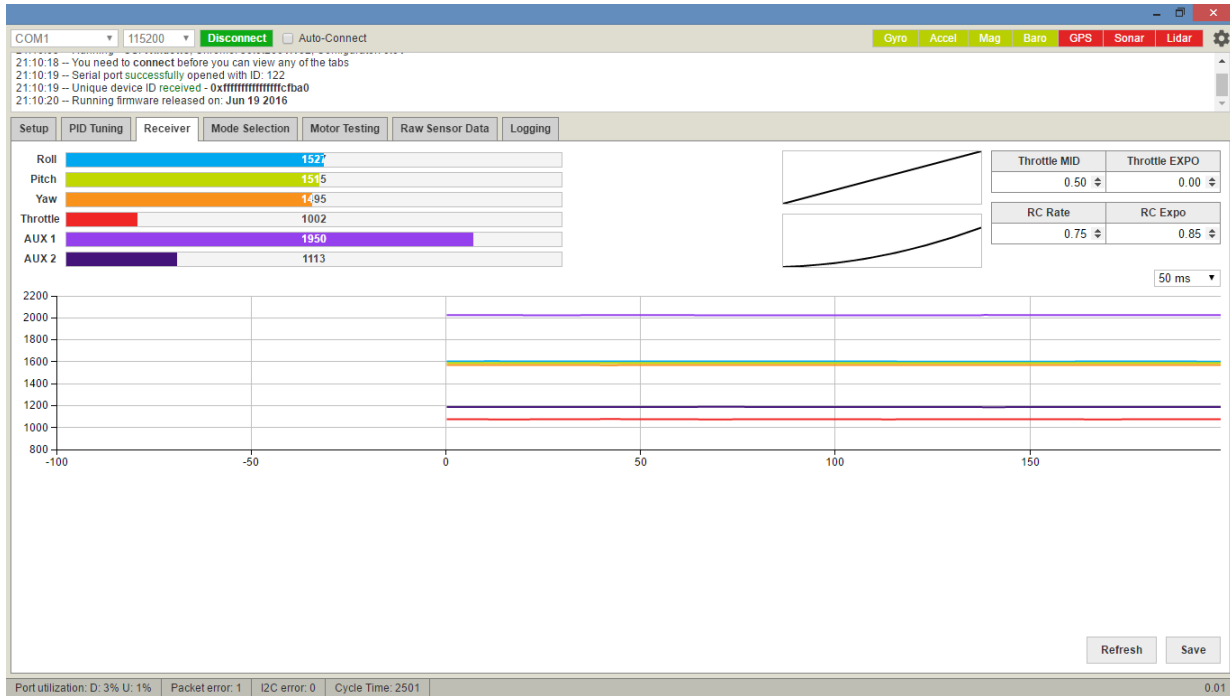
- ROLL - This is the angular rate control loop driven by the roll gyroscope
- PITCH - This is the angular rate control loop driven by the pitch gyroscope
- YAW - This is the angular rate control loop driven by the yaw gyroscope
- ALT - This is the automatic altitude control loop. It is controlled using a combined barometric and inertial signal created by a sensor fusion algorithm
- VGS - This is not strictly a control loop but provides Vertical Gain Scheduling (VGS) for the altitude control loop. Simply stated, it provides for increased or “Scheduled” throttle response to the control error signal when the craft is BELOW set point. This is because throttle response is asymmetric; it takes a larger increase in throttle for the craft to rise than a decrease in throttle for it to fall
- LEVEL - This is the PITCH/ROLL Euler angle stabilization loop. The output of this control loop causes the craft to be self-righting when in the “Angle” or “Horizon” flight modes
- MAG – Proportional gain that changes the yaw of the craft back to the heading set point

The detailed behavior of the PID control loops, their tuning constants and tuning methodology will be discussed in more detail in a later section. The “ROLL & PITCH rate”, “YAW rate” and

“TPA” fields are for throttle-based gain scheduling of the various PID control loops. These are primarily legacy items carried over from MultiWii and are not commonly used.

Fig. 11 shows the “Receiver” tab screen view. A real-time bar graph view with numerical indication of each channel’s pulse duration (in μs) is shown in the upper left.

Figure 11. A screen shot of the SuperFly configurator “Receiver” tab.

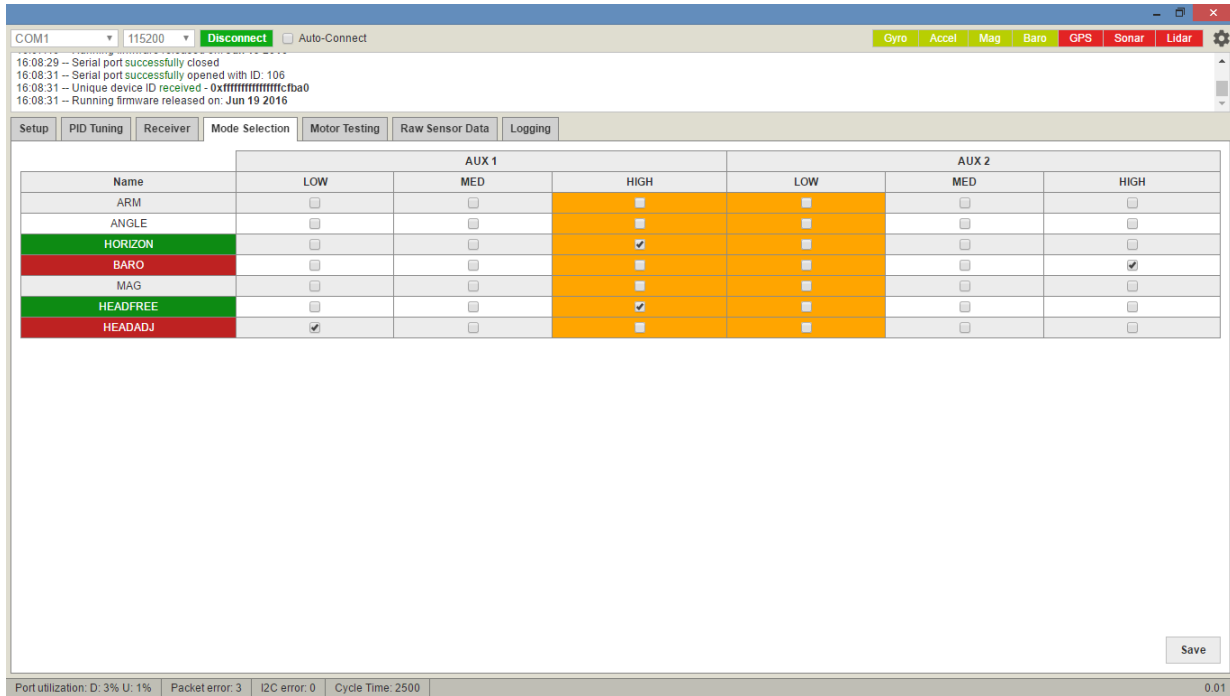


A real-time line graph of each channel’s pulse duration is also shown for visualization and diagnostic purposes.

The “Throttle MID” and “Throttle EXPO” dialog boxes are used to make custom throttle response curves. Typically the “Throttle MID” is set to the percentage throttle required for the craft to hover. Increasing the “Throttle EXPO” makes a more and more pronounced “Plateau” in the throttle response curve about the “Throttle MID” value. When these parameters are used, if desired, the copter will hover at ~50% throttle. As the “Throttle EXPO” term is increased, throttle response about the hover point will be progressively desensitized. The “RC Rate” and “RC Expo” perform a somewhat similar function but for the pitch, roll, and yaw stick functions. The “RC Rate” term limits the pulse width control range for pitch roll and yaw. The “RC Expo” term flattens out the stick response near the center of the travel while recovering the full “RC Rate” response at full stick travel. This makes it easier to provide fine correction for precision hover while having full response available for large control stick motions.

Fig. 12 shows the operation of the highly important “Mode Selection” tab. This screen is where the basic flight modes of the copter and how their activation/deactivation controls are set up for your particular RC transmitter device.

Figure 12. A screen shot of the “Mode Selection” tab.



The screen will automatically populate the vertical columns with the available auxiliary RC channels. In this case, the six channel cPPM receiver option is active so the choices are the two “AUX 1” and “AUX 2” channels. For each auxiliary RC channel, there are three logical states possible, “LOW”, “MED” and “HIGH”. As a practical matter, this arrangement was contrived to get additional control capability from auxiliary channels controlled by three-position toggle switches. For most purposes, the “LOW” and “HIGH” states will be used.

The orange blocks indicate that at the moment, “AUX 1” is in the high state and “AUX 2” is in the low state. Since the “HORIZON” and “HEADFREE” flight mode boxes are checked in the “AUX 1, HIGH” column and indeed “AUX 1” is in the high state, the “HORIZON” and “HEADFREE” modes are highlighted in green to indicate they are active. Continuing with this example, toggling “AUX 1” low will deactivate these two flight modes and activate the “HEADADJ” mode because the box for this flight mode is checked in the “AUX 1, LOW” column. So flight modes can be assigned to the available auxiliary channel switches merely by checking the desired flight mode’s box in the appropriate column and clicking the “Save” button on the lower right. To test, merely actuate the auxiliary switches on the transmitter to see that the desired flight modes are active for the desired transmitter switch positions.

Briefly, the available flight modes are:

- ANGLE - Euler angle stabilized flight (pitch and roll). The craft will be self-righting
- HORIZON - A hybrid stabilized flight mode. For small stick movements, the craft acts just like it does in “ANGLE” mode. However, for large stick excursions the craft will be more responsive as if it were only using angle rate control from the gyroscopes. The craft is still self-righting but a bit more aerobatic
- ACRO - Although not explicitly shown on the configurator GUI, the “ACRO” flight mode is the default when neither “HORIZON” nor “ANGLE” modes are active. The craft is under angle rate control from the gyroscopes only and is not self-righting
- BARO - Automatic altitude control mode. When “BARO” mode is active, the craft will maintain altitude at the same value as when the mode was engaged. Leaving “BARO” mode will return the craft to direct throttle control
- MAG – The craft will control yaw to maintain the heading set point. Moving the yaw stick in this mode will change the heading. In this mode, the control axes are always with respect to the front of the craft... So if the heading is changed, the directional response of the controls also changes!
- HEADFREE - In this flight mode, the directional response of the controls remains oriented with respect to the direction the craft was pointed when it was armed. So if you stand behind the rear of the craft and arm it, pushing the pitch stick forward will always make the craft fly away from you, no matter which way the head is pointed
- HEAD ADJ - This flight mode updates the heading set point. It is useful in the event that the pilot becomes disoriented. Yaw the craft so the nose points away from the pilot and briefly activate the “HEAD ADJ” flight mode. This resets the heading set point and the “HEADFREE” reference so that the RC controls will return to expected behavior.

For beginning pilots, “HORIZON” and “HEADFREE” flight modes are recommended for the default configuration. It is also recommended to enter the “HEADADJ” mode when canceling the “HEADFREE” mode. This can allow the pilot to get the craft properly reoriented in the event that directional control response is not as expected.

Fig. 13 shows the “Motor Testing” tab view of the configurator GUI. In the original embodiment, it was intended to be able to control each motor independently using the slider controls below the individual motor vertical bar graph indicators. Currently, this feature has been deactivated but may be reinstated in a future release. Above the motor bar graph indicators there is a time trace graph that shows the raw sensor signals from the X, Y and Z accelerometers. This utility is helpful for assessing the vibration level to which the flight controller is being subjected. Excessive motor vibration can cause airframe “Twitching” and other unstable flight performance. The accelerometer time trace graphs are also useful for dynamic propeller balancing. This topic is considered to be for advanced users and is discussed in detail in numerous references on the internet.

Figure 13. A screen shot of the “Motor Testing” tab.

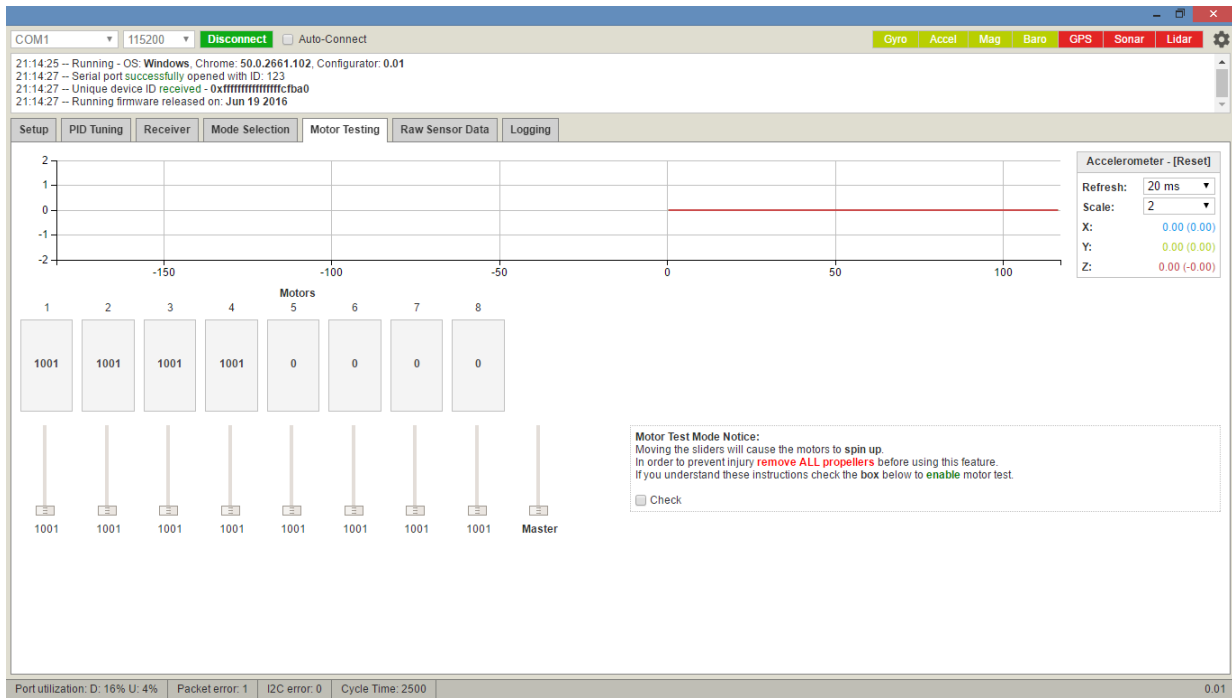


Fig. 14 shows the “Raw Sensor Data” tab view. This screen is primarily for diagnostics and assessing individual sensor performance and can be a highly useful tool.

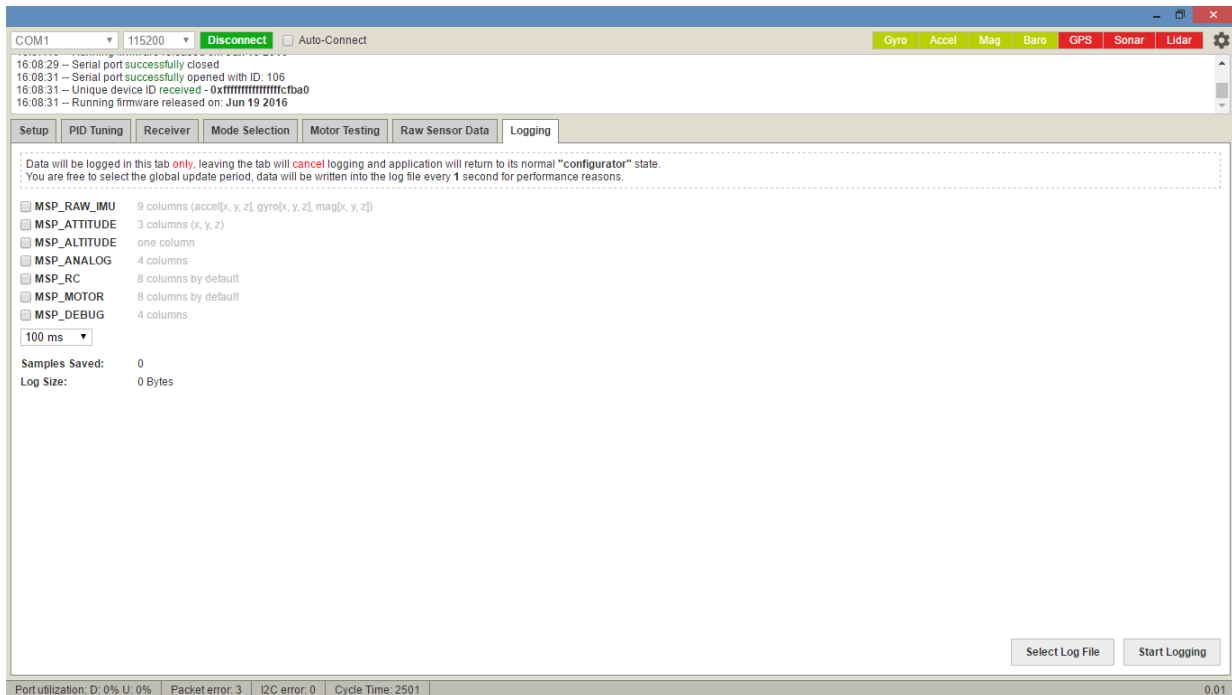
Figure 14. A screen shot of the “Raw Sensor Data” tab.



The check boxes along the top row select which sensor families will have real-time graphs displayed in the tab. The choices include “Gyroscope”, “Accelerometer”, “Magnetometer”, “Barometer” and “Debug” and are largely self-explanatory. For the first three options, there are individual sensors for the X, Y and Z orientations that are plotted simultaneously in the respective graph pane. In addition to the real-time graphs, the associated legend boxes on the right-hand side of the screen give real-time indication of the numerical values. The drop-down menus in the legend boxes allow selection of the Y-scale and the sampling period. The “Debug” option is quite useful for development and advanced troubleshooting. In the SuperFly flight control sketch the “debug[4]” integer array variable is defined as a global variable. Any quantity of interest in the flight controller code can be assigned to one of the debug[i] variables and plotted in the “Raw Sensor Data” tab.

Fig. 15 shows the “Logging” tab view. This screen accesses the data logging capabilities of the configurator GUI. This function can be extremely useful for advanced troubleshooting and flight controller code development. Data logging is accomplished by selecting any/all of the checkbox options. The data collection sample period can be selected using the dropdown on the left-center side of the screen. The “Select Log File” radio button on the lower-right pops up a file selection dialog box for specifying the data file and the “Start Logging” radio button commences data collection.

Figure 15. A screen shot of the “Logging” tab.



The data collection checkbox options are grouped together by the “MultiWii Serial Protocol” (MSP). In the event it is desired to collect a signal explicitly handled by the available MSP group

options, it is recommended to assign it to one of the “debug[i]” array variables and collect the “MSP_DEBUG” group. The available MSP group options include:

- MSP_RAW_IMU
 - Raw X, Y, and Z accelerometer data
 - Raw X, Y, and Z gyroscope data
 - Raw X, Y, and Z magnetometer data
- MSP_ATTITUDE
 - Roll angle
 - Pitch angle
 - Heading
- MSP_ALTITUDE
 - Estimated altitude
 - Variometer (vertical velocity)
- MSP_ANALOG
 - Battery voltage
 - Power meter sum (disabled)
 - RSSI (Receiver Signal Strength Indicator, disabled)
- MSP_RC
 - Pulse width for all active RC channels
- MSP_MOTOR
 - Motor command values for all active motors, M0 - M3
- MSP_DEBUG
 - Current values of the four debug[i] variables

Accelerometer Calibration Procedure

This section describes the SuperFly accelerometer calibration procedure in detail. The Invensense MPU9250 accelerometers are of very good quality but, like all accelerometers, are subject to scale and offset errors. The EM7180 coprocessor is capable of calculating the offset and scale biases and applying them to the accelerometer data in real time. This process results in superior roll and pitch Euler angle estimation. More importantly, accelerometer error correction is essential to the accurate calculation of vertical linear acceleration which enables inertial enhancement of the altitude controller. Taking reasonable care during the following procedure will go help ensure stable flight performance and altitude control.

The accelerometer calibration procedure is simple in concept. The object is to align each accelerometer axis parallel to gravity, record samples and store the average to EEPROM. This is done for the X, Y, and Z accelerometers both parallel and anti-parallel to gravity for a total of six calibration conditions. The keys to successful accelerometer calibration are: 1) Holding the quadcopter or SuperFly board precisely with respect to gravity and 2) Making sure that the copter/board doesn't move during data collection. Holding the board in the correct orientation can be accomplished by any convenient means. Fig. 16 shows examples of an X-copter held with the X, Y, and Z accelerometers held aligned to gravity. A "Third Hand" adjustable holding aid with an insulated alligator clip is used to hold the copter and can also be used to hold an un-mounted SuperFly board. Most any mechanically stable means can be used to hold the SuperFly board but **care should be exercised not to cause any mechanical damage to board components or electrical shorts.**

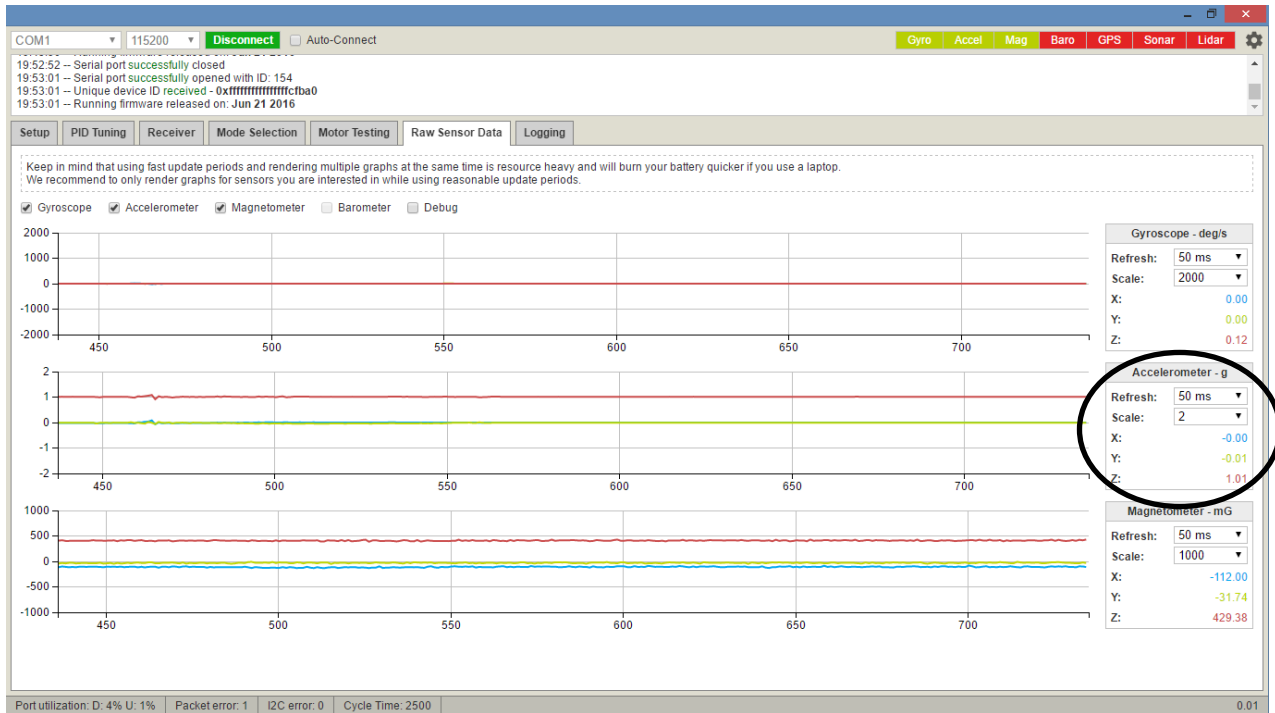
Figure 16. A U816 quadcopter held with each of the accelerometers aligned to gravity using an insulated "Third Hand" holding aid.



For each of the six calibration conditions, the accelerometer of interest needs to be precisely aligned to gravity prior to activating the data collection and storage process. The best way to accomplish this is using the "Raw Sensor Data" tab of the configurator GUI. Fig. 17 shows the X, Y, and Z accelerometer data with the Z accelerometer properly oriented. The Z accelerometer shows $\sim 1g$ while the X and Y accelerometers indicate $0.02g < |A_X|$ and $0.02g < |A_Y|$. Once the desired accelerometer is verified to be aligned properly it remains to collect the data, compute

the average and store the result in the EEPROM. This is done automatically from the “Setup” tab of the configurator GUI.

Figure 17. The Z accelerometer properly aligned parallel to gravity and verified using the configurator GUI. The accelerometer of interest (Z in this case) will show approximately $\pm 1g$ ($+1g$ in this case) while the other two accelerometers should indicate less than $\pm 0.02g$.



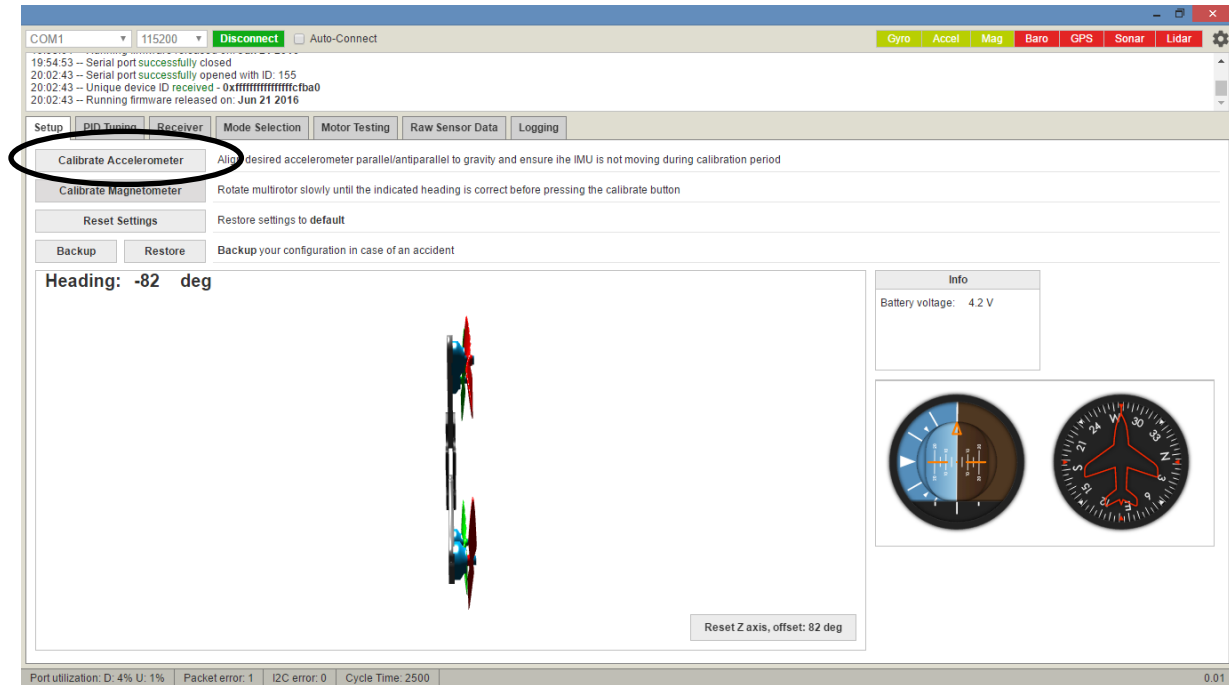
Once the accelerometer alignment is verified on the “Raw Sensor Data” tab navigate back to the “Setup” tab as shown in Fig. 18. Press the “Calibrate Accelerometer” button on the upper left. The SuperFly will then collect, average and store the accelerometer data for this orientation in a few seconds. The blue indicator LED will flash momentarily immediately after the data is written to the EEPROM. As mentioned earlier, this same basic procedure must be repeated for six orientations of the craft for the accelerometer calibration to be complete:

- Orient the craft so that the accelerometer of interest indicates $\pm 1g$ and the other two are reading less than $\pm 0.02g$
- Make sure the flight controller/craft are held completely still and press the “Calibrate Accelerometer” button on the “Setup” tab of the configurator GUI. Wait for the data collection to complete
- Repeat so that data is collected for the $A_x = +1g$, $A_x = -1g$, $A_y = +1g$, $A_y = -1g$, $A_z = +1g$ and $A_z = -1g$ conditions

The SuperFly flight controller sketch is written to check the accelerometer calibration values stored in the EEPROM at startup and determine if they are valid or not. If any of the six

calibration values are deemed invalid, the entire data set is not uploaded to the EM7180 and no accelerometer corrections are applied.

Figure 18. “Setup” tab view immediately after verifying proper accelerometer orientation on the “Raw Sensor Data” tab. In this case, the Y accelerometer is aligned with gravity. Click the “Calibrate Accelerometer” button to collect/store the calibration data for this orientation.



Heading/Magnetometer Calibration Procedure

Perhaps the most important capability of the EM7180 sensor fusion coprocessor is accurate estimation of compass heading. A great deal of mathematical processing is required to take the raw data from a set of (X, Y, Z) magnetometers and convert it into a meaningful heading estimate. Pitch and roll tilt, hard iron distortion, soft iron distortion and magnetic compass declination effects must be properly taken into account. The EM7180's sensor fusion algorithm does these tasks very well and furthermore detects and rejects transient magnetic interference. These complicating factors are particularly strong on very small craft. Brushed DC motors emanate particularly strong stray fields and they are very close to the magnetometers when compared to larger multirotor UAV's. When the EM7180's sensor fusion algorithm is properly optimized it effectively negates the imperfections of the Super Fly's magnetic environment.

However, since each magnetic environment presents its own unique challenges to accurate heading estimation, the EM7180's sensor fusion algorithm is adaptive. What that means is the EM7180 is always calculating two estimates of the general orientation; one is referenced to the (X, Y, Z) accelerometers and the other is referenced to the (X, Y, Z) magnetometers. Since

gravity is much more predictable and doesn't suffer from interference unlike the earth's magnetic field, the various magnetic correction weighting factors are subject to dynamic adjustment to bring the magnetic solution into agreement with the gravitational solution.

As a practical matter, the EM7180's heading algorithm "Learns" the magnetic environment and compensates for the particular non-idealities as the SuperFly board is slowly rotated in three dimensions. Depending upon various factors including quadcopter construction, power lead routing, motor placement and individual magnetometer imperfections it can take up to several minutes of rotating the quadcopter for the algorithm to relax to the correct, stable solution. If this "Learning" process had to be done every time the SuperFly board was powered up, that would be a significant operational limitation. Fortunately, the EM7180 supports the export of all the algorithm's key adjustment parameters to EEPROM storage and import of the same parameters from the EEPROM back into the EM7180 at boot-up.

This feature is referred to as "Warm Start" and is extremely useful from a practical perspective. It enables a simple and highly accurate heading calibration procedure that automatically applies the calibration results at boot-up. Furthermore, like with the accelerometer calibrations, the magnetic corrections are applied to the raw magnetometer data in real time. Finally, the flight controller sketch checks the EEPROM during startup to determine if valid warm start data has been stored. If yes, it will upload the warm start parameters to the EM7180 as the SuperFly initializes. If not, warm start parameter upload is skipped.

The heading/magnetometer calibration procedure is simple to execute. It is recommended to complete construction of your quadcopter and mount the SuperFly board to the frame. To get the best calibration results, it is a good idea to have the SuperFly board in the exact local environment it will experience during flight. It is also a good idea to update the "config.h" file with the proper magnetic declination for your location, as described earlier in this document. Some reasonable indication of true north is helpful to know when the SuperFly's indicated heading is correct. This could be from a known good compass or smart phone/tablet compass App. The steps to performing the heading/magnetometer calibration are:

1. Power-up the SuperFly controller and allow a few seconds for the initialization process to complete
2. Slowly rotate the quadcopter randomly in all three dimensions for a minute or so
3. Place the quadcopter on a flat surface well away from any computer monitors or disk drives (or other ferromagnetic items) and point it true north (or any other known heading)
4. Connect the SuperFly board to the USB-serial converter, establish the serial connection with the configurator GUI and navigate to the "Setup Tab"
5. Observe the heading (indicated in degrees East of North) and wait until it is stable. It should agree with the known heading of the quadcopter within a few degrees
6. If the indicated heading is not satisfactory, repeat step 2 and check it again. Eventually, the indicated heading should converge to the correct value

7. As a final check, see if the Z magnetometer component is symmetric about 0. Navigate to the “Raw Sensor Data” tab and find the “Magnetometer” time trace graph as shown in Fig. 19
8. M_Z should be the red trace; it is indicating the vertical component of the earth’s magnetic field. Slowly turn the quadcopter upside down and then right side up again. If the algorithm has properly converged on the correct solution, M_Z should be symmetrical about the origin for the upright and inverted conditions
9. If the indicated heading is correct and M_Z is symmetrical about the origin for the upright and inverted positions the heading algorithm has been properly “Trained”. If not, repeat step 2 and check again. Except in instances where there is a clear malfunction, the heading algorithm will eventually converge to give accurate heading and symmetrical M_Z response
10. Navigate back to the “Setup” tab and press the “Calibrate Magnetometer” button as shown in Fig 20. It is recommended to not disturb the copter while the calibration function is active.

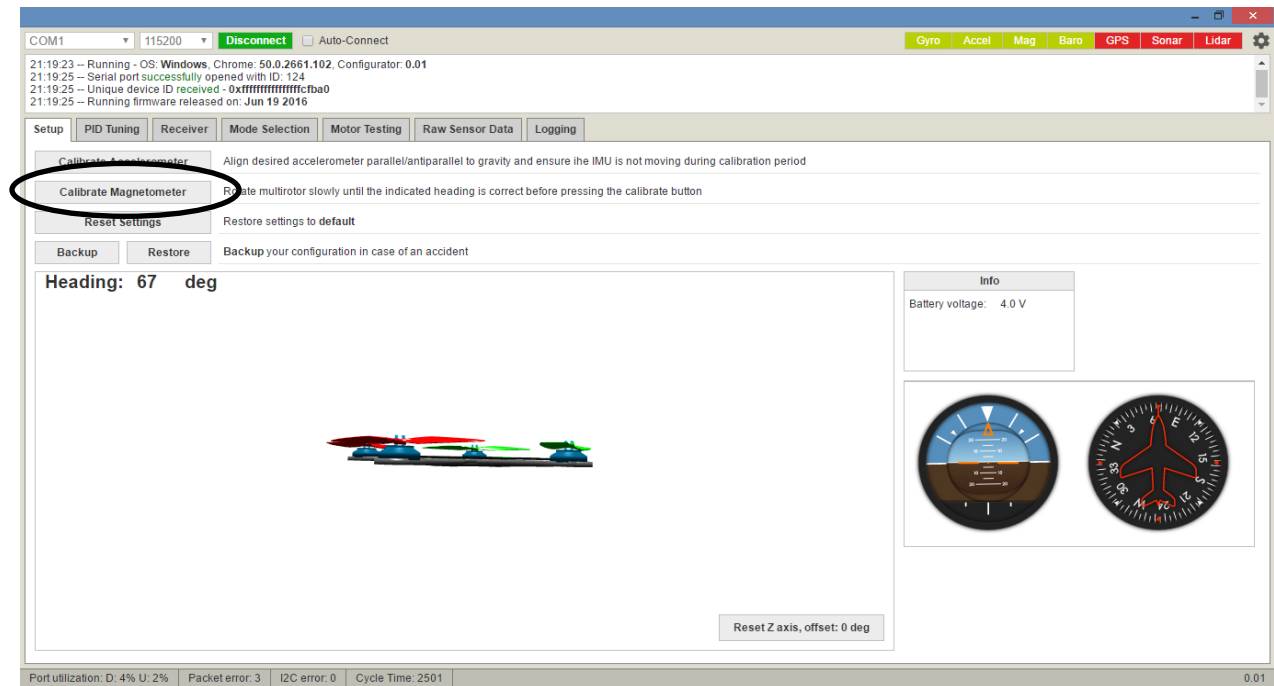
Figure 19. A time trace graph of the X, Y, and Z magnetometer signals showing that M_Z is symmetrical about the origin as the quadcopter is inverted. This indicates that the heading algorithm has converged to the correct state.



The warm start parameter transfer and EEPROM storage macro will complete in a few seconds and real-time operation of the configurator GUI should resume.

It is simple to test whether or not the heading/magnetometer calibration procedure was successful. Power cycle the SuperFly, re-connect to the GUI and navigate to the “Setup” tab. The heading should immediately be reading the expected value within a few degrees.

Figure 20. Once the heading value is indicating properly, click the “Calibrate Magnetometer” button.



Now that the SuperFly flight controller has been properly set up and calibrated, it is now ready for test flight and PID gain tuning.