OPWiki Documentation

Release 0.1.4

OpenPilot community

October 16, 2016

CHAPTER 1

LibrePilot/OpenPilot User Manual

1.1 Downloading and Installing Ground Control Station

1.1.1 LibrePilot

You can use the LibrePilot Ground Control Station (GCS) both to configure your controller board and to control and monitor your aircraft during flight. More commonly, you would use a conventional radio control transmitter to control your vehicle, but the GCS is also capable of doing so.

LibrePilot GCS is currently available for the Windows, Mac, and Linux operating systems.

Downloading the GCS installer

The first step is to download the appropriate GCS installer. Using the release table below, select the Download Link that corresponds to your computer's operating system.

If updating from a previous release, you may wish note the current settings for your vehicle first by creating a .UAV file or taking screenshots of vehicle settings.

Important: The software on this page comes with no warranty. Use it at your own risk and please be careful!

Visit the LibrePilot forums if you have questions and/or suggestions. Extensive documentation about how the system works and how to install and configure it is available in the LibrePilot wiki and here.

Latest version can be found here: https://librepilot.atlassian.net/wiki/display/LPDOC/Downloads.

RELEASE – 15.09 – First LibrePilot Release - Supermoon Eclipse

This is the first LibrePilot release.

This version supports the CopterControl, CC3D, Atom, Revolution, Revolution Nano, Platinum GPS (V9) and OPLink Modems. The main focus of this release is to bring back support for CC3D and a general re-branding of the GCS.

Some key addition in this release:

- Easytune : A TxPID feature for 'offline' tuning,
- Improved motor scaling,
- Different Acro+ factors for roll and pitch
- Fixed wing: Roll differential and setup wizard improvements,
- Import/Export and manage templates,

  - OSX fix for crashes running ‘El Capitan’,
  - ...

The full list of bug fixes and enhancements in this release is available here: 15.09 release JIRA list.

Windows

LibrePilot-15.09-win32.exe

Uninstall previous version first.

OSX

LibrePilot-15.09-osx.dmg

PPA for Ubuntu 15.04+

```
$ sudo add-apt-repository ppa:librepilot/release
$ sudo apt-get update
$ sudo apt-get install librepilot
```

Arch Linux

Available on the AUR or via repository by adding

```
[librepilot]
SigLevel = Optional TrustAll
Server = http://download.librepilot.org/repo/archlinux/librepilot/$arch
```

to /etc/pacman.conf.

Fedora

Repository available here.

Source code

- Archives are available at http://download.librepilot.org/source/
- The git mirror at https://github.com/librepilot/LibrePilot.git

Bootloaders

Bootloader Update

Installation of the GCS

Open the GCS installer file that you downloaded and follow these steps:

1. Choose a language from the drop-down list on the first page, then click OK. (You can cancel installation at any point by clicking Cancel.)
2. The LibrePilot Welcome screen appears. Click Next.

3. Review the conditions of the license agreement and then click I Agree to accept the terms.
4. You can select which components to install in the Choose Components dialog box. Click Next to accept the default selections.

Note: Note that the Mesa OpenGL driver may be required by older operating systems and is a required component for the GCS interface. If major elements of the GCS user interface fail to display, re-installation with selection of the Mesa OpenGL driver may help.

5. You can specify where to install LibrePilot GCS in the Choose Install Location dialog box. Click Browse to choose a location or Install to install the software in the default location shown in the text box.

   Previous installations of the LibrePilot GCS were installed in the Documents and Settings directory on Windows machines. The latest default GCS location is the standard Program Files location on Windows machines.
6. LibrePilot GCS installs on your computer. Click Next when installation is complete.
7. If you have have chosen the default setup, Windows will now install the CDC driver for the Virtual Comm Port of your LibrePilot board.

Note: Note that the CDC driver is not required to connect and configure your LibrePilot board with the GCS. You need the CDC driver for Virtual Comm Port support.

8. Click Finish to complete installation. (Clear the check box if you don't want LibrePilot GCS to run immediately.)
9. If you choose to run LibrePilot GCS immediately, click OK to load the default configuration file.
10. The LibrePilot GCS start page appears. Congratulations! You can explore LibrePilot GCS or proceed to the next step, Installing or Updating Your Firmware.

1.1.2 OpenPilot

You can use the OpenPilot Ground Control Station (GCS) both to configure your controller board and to control and monitor your aircraft during flight. More commonly, you would use a conventional radio control transmitter to control your vehicle, but the GCS is also capable of doing so.

All OpenPilot products use the same Ground Control Station. OpenPilot GCS is currently available for the Windows, Mac, and Linux operating systems.

Warning: OPENPILOT WAS DISCONTINUED!
Use LibrePilot instead.

Downloading the GCS installer

The first step is to download the appropriate GCS installer. Using the release table below, select the Download Link that corresponds to your computer's operating system.

If updating from a previous release, you may wish note the current settings for your vehicle first by creating a .UAV file or taking screenshots of vehicle settings.

Important: The software on this page comes with no warranty. Use it at your own risk and please be careful!

Visit the OpenPilot forums if you have questions and/or suggestions. Extensive documentation about how the system works and how to install and configure it is available in this wiki.

Latest version can be found here: http://www.openpilot.org/download/.

RELEASE – OP-15.05.02 – Revolution Nano, Revo, OPLM and Platinum GPS

Note: Please note that **CC3D is NOT supported by this release**. To find the latest release for the CC3D, please see Release – 15.02.02 below.

Windows NSIS Installer

OpenPilot-RELEASE-15.05.2-win32.exe

Uninstall previous version first.

Mac OS X distribution image

OpenPilot-RELEASE-15.05.2-osx.dmg

Open as a standard distribution image.

Linux 64 bit Debian package

openpilot_15.05.2-1_amd64.deb

Uninstall previous version first, then use your system package installer to install.

Note: Please note that Linux .debs will only work with Linux versions Ubuntu 14.04 or later, due to lack of popularity we are no longer building 32 Bit .debs but if needed you can build them easily using the source below and the 'make package' command.

Source code

SRC-OpenPilot-RELEASE-15.05.2.tar.gz

```
Note that the CC3D is not supported by this release.

This release fixes an important bug of Nano attitude drift with Basic
attitude estimation.

All Revolution hardware running 15.05.01 should upgrade to 15.05.02.

Note that this is a hotfix; to get the new defaults a full erase settings
is required.

Stabilization settings and banks can be imported back after upgrade

Furthermore, please review your vtolpathfollowersettings:HorizontalVelMax;
a value of around 4m/s would be more appropriate for preliminary trialing of
a new release and will be changed in future.

This version supports Revolution Nano, Revolution, OPLink Modems, and
Platinum GPS.
```

(Previous) RELEASE – OP-15.02.02 – Revo, CC3D, Atom, CC and v9 GPS

This version supports the CopterControl, CC3D, Atom, and Revo as well as the OPLink Modems.

Windows NSIS Installer

OpenPilot-RELEASE-15.02.02-win32.exe

Uninstall previous version first.

Mac OS X distribution image

OpenPilot-RELEASE-15.02.02-osx.dmg

Open as a standard distribution image.

Linux Debian packages

32 bit openpilot_15.02.02-1_i386.deb

64 bit openpilot_15.02.02-1_amd64.deb

Uninstall previous version first, then use your system package installer to install.

Note: Please note that Linux .debs will only work with Linux versions Ubuntu 14.04 or later.

```
This release fixes a bug that prevents revo onboard mag to work correctly.

Release Notes - OpenPilot - Version RELEASE-15.02.02

The full list of bugfixes in this release is accessible here:
https://progress.openpilot.org/issues/?filter=12262

** Bug
* [OP-1820] - fix onboard mag orientation
* [OP-1821] - Tricopter tail servo wrong speed on wizard
* [OP-1827] - Version ID wrong in Windows uninstaller
* [OP-1857] - PPM on Flexi does not work on CC/CC3D

** Task
* [OP-1831] - due to oneshot higher pid values ki now shows "red" warning in
stabilization page
```

Installation of the GCS

Open the GCS installer file that you downloaded and follow these steps:

1. Choose a language from the drop-down list on the first page, then click OK. (You can cancel installation at any point by clicking Cancel.)
2. The OpenPilot Welcome screen appears. Click Next.
3. Review the conditions of the license agreement and then click I Agree to accept the terms.
4. You can select which components to install in the Choose Components dialog box. Click Next to accept the default selections.

Note: Note that the Mesa OpenGL driver may be required by older operating systems and is a required component for the GCS interface. If major elements of the GCS user interface fail to display, re-installation with selection of the Mesa OpenGL driver may help.

5. You can specify where to install OpenPilot GCS in the Choose Install Location dialog box. Click Browse to choose a location or Install to install the software in the default location shown in the text box.

   Previous installations of the OpenPilot GCS were installed in the Documents and Settings directory on Windows machines. The latest default GCS location is the standard Program Files location on Windows machines.

6. OpenPilot GCS installs on your computer. Click Next when installation is complete.

7. If you have have chosen the default setup, Windows will now install the CDC driver for the Virtual Comm Port of your OpenPilot board.

Note: Note that the CDC driver is not required to connect and configure your OpenPilot board with the GCS. You need the CDC driver for Virtual Comm Port support.

8. Click Finish to complete installation. (Clear the check box if you don't want OpenPilot GCS to run immediately.)

9. If you choose to run OpenPilot GCS immediately, click OK to load the default configuration file.

10. The OpenPilot GCS start page appears. Congratulations! You can explore OpenPilot GCS or proceed to the next step, Installing or Updating Your Firmware.

Warning: OPENPILOT WAS DISCONTINUED!
Use LibrePilot instead.

1.2 Installing or Updating Your Firmware

You should check the bootloader version of any CC or CC3D and know the Bootloader version. If not already on Version 4, you MUST upgrade your Bootloader via the documentation linked below before using any firmware from 13.xx and newer.

For directions on checking bootloader version and installation instructions, please see the Bootloader Update page.

Note: Please ensure you download the correct v4 Bootloader. CC, CC3D/Atom, OPLink and Revo all use different bootloader files!

The firmware for your OP boards is packaged within the GCS download. To update the firmware on the various OpenPilot boards follow these steps:

1. Uninstall any previous versions of OpenPilot GCS

2. Install the latest OpenPilot GCS version (available from the previous page of this Wiki, the Downloading and Installing Ground Control Station page)

3. Read through the informational documentation for CC/CC3D and/or Revolution page by using the links at the bottom of this page.

4. Once familiar with your board's functions, find the **Welcome** tab of the GCS and with your board plugged in, proceed through the **Vehicle Setup Wizard** for Multirotor or Fixed Wing.

For vehicles other than Multirotor or Fixed Wing, or to install firmware manually (without the wizard) you must take the following steps to ensure that your firmware is updated correctly:

1. Uninstall any previous versions of OpenPilot GCS

2. Install the latest OpenPilot GCS version (available from the previous page of this Wiki, the Downloading and Installing the Ground Control Station page)

3. Open the OpenPilot GCS and go to the **Firmware** Tab

4. With the board unplugged, click the **Rescue** button
5. Connect the board
6. (There's a chance that the new drivers might not finish installing before the timer runs out. If this happens, wait until the drivers finish their install, then unplug the board, click Rescue again, connect the board and continue)
7. and click the **Erase Settings** button.
8. Wait approximately 30 seconds
9. Click the **Upgrade & Erase** button and follow the on-screen instructions
10. Firmware install/update complete.

> **Warning:** Never install any firmware on your board that came from untrusted source! Trusted sources are the OpenPilot Downloads page or firmware that you have built yourself from the official git repository.

Next, select your flight controller:

- CC / CC3D / Atom
- Revolution

1.3 CopterControl / CC3D / Atom Hardware Setup

1.3.1 CopterControl / CC3D / Atom Hardware Setup

CopterControl3D (CC3D and Atom) and CopterControl Introduction

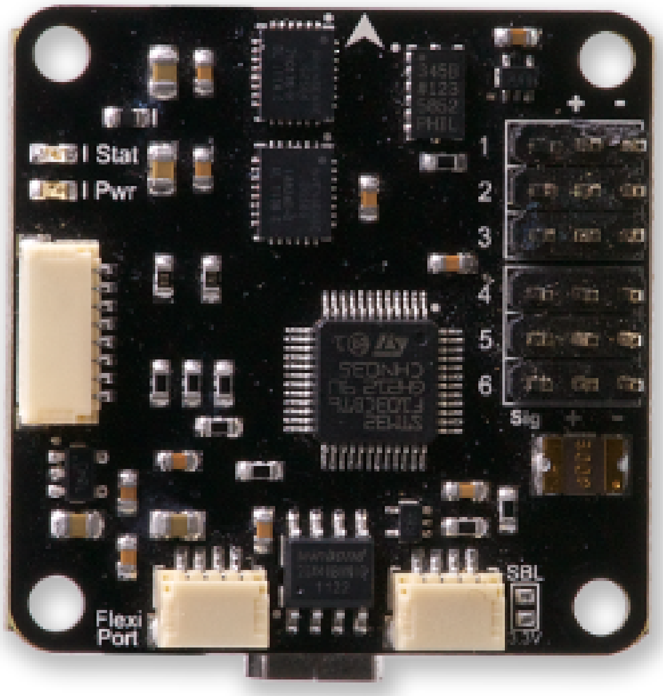
The CopterControl, CC3D and Atom flight controllers are all types of stabilisation hardware which run the OpenPilot firmware. They can be configured to fly any airframe from fixed wing to an octocopter using the OpenPilot Ground Control Station (GCS) software. If you haven't already installed the Ground Control software, see Downloading and Installing Ground Control Station.

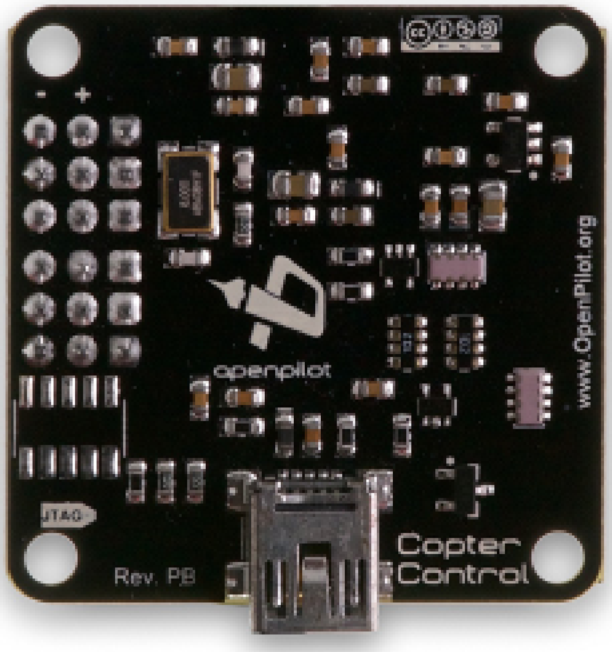
The CopterControl was the first generation board, which ceased manufacture in 2012 due to lack of availability of the gyro sensors used for stabilisation. The board design was then revised and released with an improved gyro sensor which is less affected by temperature changes. This revision is called the CC3D, and apart from the gyro sensor change is identical to the original CopterControl. The Atom is the latest edition to this family - it has the full functionality of the CC3D, but in a smaller form factor and was made available in August 2014 by getfvp.com and readymaderc.com.

All three of these boards are 100% compatible with the latest firmware release when updated to the latest bootloader. All documentation referring to the original CopterControl board is applicable to both the CC3D and Atom boards.

Getting to know your board

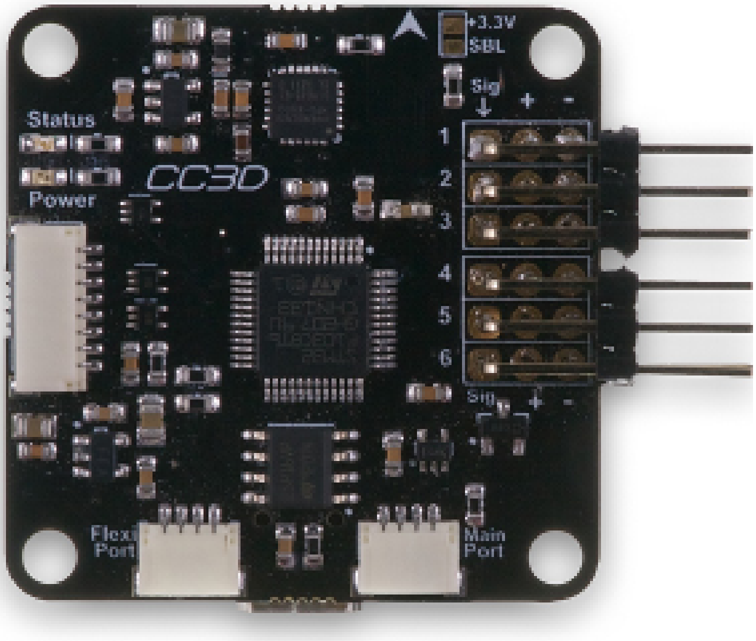
CopterControl

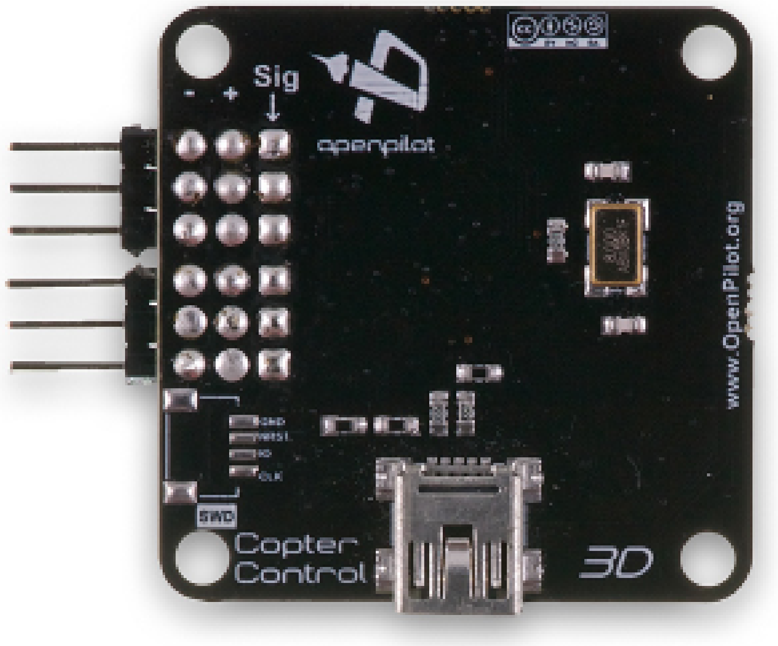




Weight: 8g

CopterControl3D/CC3D

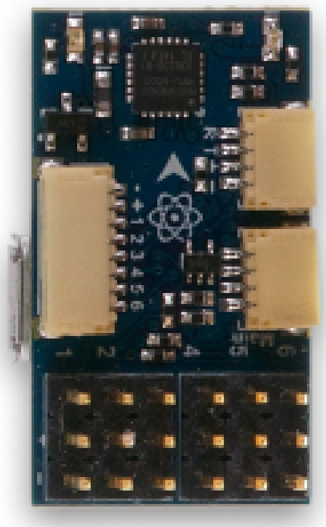


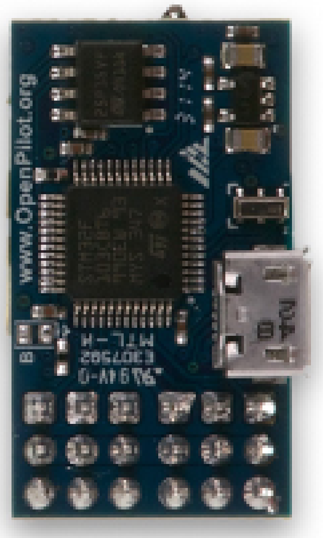


Weight: 8g

The CC3D and the Atom are both available with right angle or straight output connectors.

ATOM



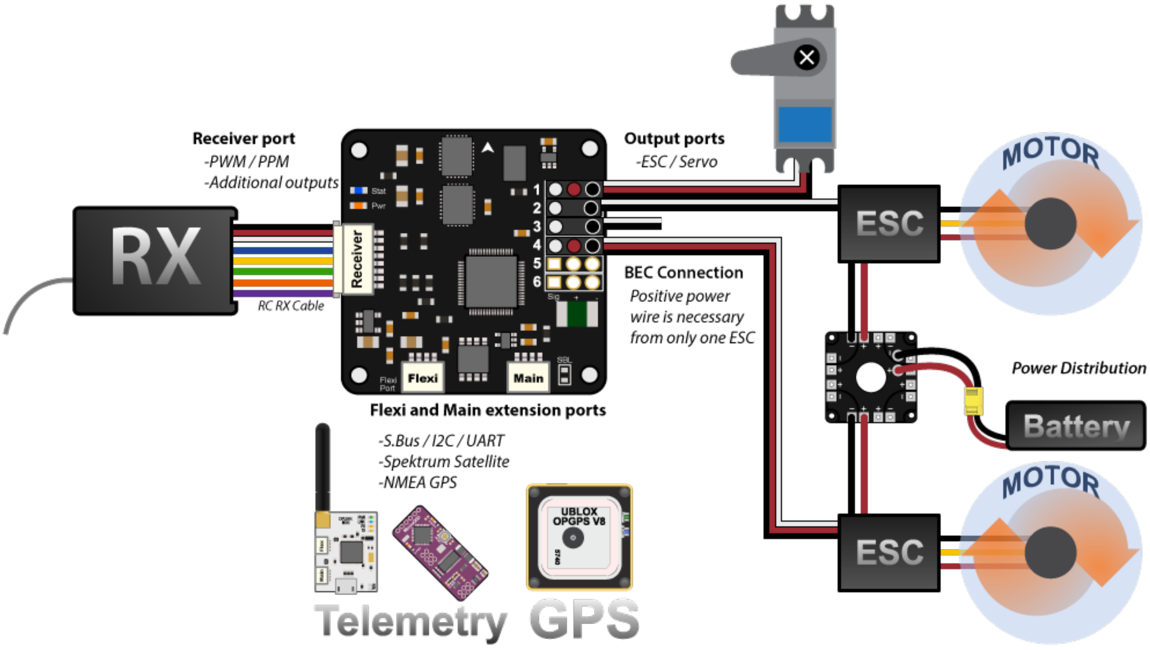


Weight: 4g

The Atom doesn't have mounting holes. Cases designed to mount and protect all of the flight controllers are available from a number of sources. For example, for the Atom, www.readymaderc.com have lightweight cases with and without feet.

Connection diagram

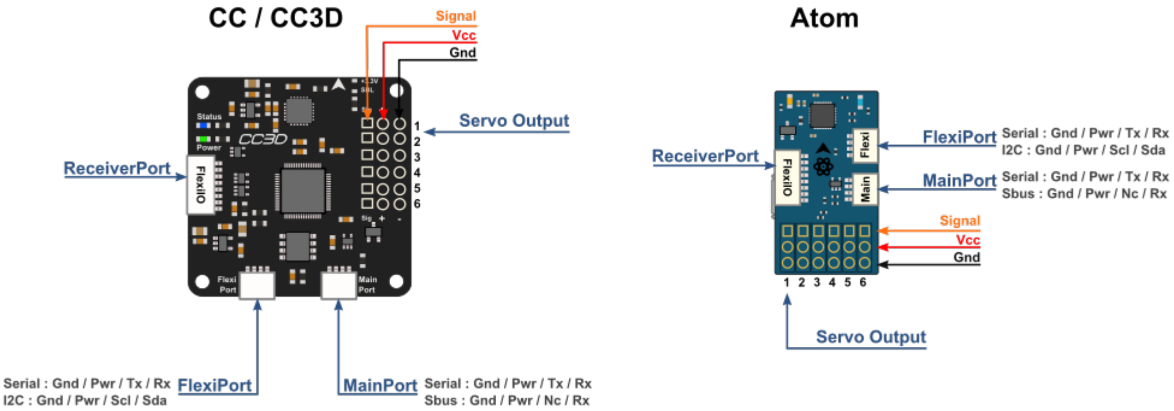
The diagram below summarizes how the overall CopterControl system is connected.



Connection Details

Ports

CopterControl / CC3D / Atom have 4 ports.


CC / CC3D
Signal
Vcc
Gnd
Servo Output
ReceiverPort
FlexiIO
Serial : Gnd / Pwr / Tx / Rx
I2C : Gnd / Pwr / Scl / Sda
FlexiPort
MainPort
Serial : Gnd / Pwr / Tx / Rx
Sbus : Gnd / Pwr / Nc / Rx
Atom
ReceiverPort
FlexiPort
Serial : Gnd / Pwr / Tx / Rx
I2C : Gnd / Pwr / Scl / Sda
MainPort
Serial : Gnd / Pwr / Tx / Rx
Sbus : Gnd / Pwr / Nc / Rx
Signal
Vcc
Gnd
1 2 3 4 5 6
Servo Output

(click image to see full size)

- **Servo Output 1-6**: These are the PWM outputs that go to servos or ESCs. Power is typically applied through these headers from only one of the ESCs. The positive (Vcc) and negative (Gnd) pins are indicated on this diagram and the board.

  Servo output pin layout is

  - Outside –> ground
  - Middle –> 5V - 15V
  - Inside –> signal

- **MainPort (also previously known as Telemetry)**: JST-SH 4-pin. This is a serial USART whose baud rate can be adjusted through the GCS. Optionally Futaba S.Bus receiver, Spektrum/JR satellite receiver or GPS can be mapped to the MainPort. Default configuration is Telemetry for connecting an RF modem.
- **FlexiPort**: JST-SH 4-pin. The function of this port also depends on the configuration and can be configured for I2C or Serial. The default configuration doesn't use this port but it can be used for Telemetry, GPS, Spektrum satellite receivers (all working), and other I2C peripherals (under development).
- **ReceiverPort** : JST-SH 8-pin. The receiver port can act as an input or output port depending on the configuration which is set in the Hardware Settings. Configuring the receiver port as an output port allows the user to assign more output channels then the 6 standard servo outputs.

  ReceiverPort use depends on the type of RC receiver in use, and whether OneShot125 or PWM Sync output is desired:

  - **PWM**
    - **PWM+NoOneShot** should be used with a normal PWM type receiver. The 6 rightmost wires of the receiver port carry the signal for each channel individually.
  - **PPM - Pin 3**
    - **PPM+NoOneShot** is used with modern PPM type receivers, that combine the control signal to one wire. The PPM stream should be sent to the first input through the white wire connected to CC ReceiverPort wire/pin 3. For a PPM receiver, only one pin is used for signal - the remaining wires connected to CC ReceiverPort wires 4-8 are left unused.
    - **PPM+PWM+NoOneShot** combines the two modes above, wire/pin 3 is used for PPM and the rest, wires/pins 4-8 are used as PWM inputs.
    - **PPM+Outputs+NoOneShot** enables PPM input in ReceiverPort wire/pin 3, and PWM output in wires/pins 5-8. These work as output channels 7-10.
  - **PPM - Pin 8**
    - **PPM_PIN8+OneShot** is a new mode, where PPM input wire is moved from previous ReceiverPort pin 3 to pin 8 to allow PWM Sync and OneShot125 to be used as ESC output modes.
  - **RECEIVER PORT AS OUTPUTS**
    - **Outputs+OneShot** makes ReceiverPort pins 5-8 work as output channels 7-10. ReceiverPort pins 3 and 4 are unused. This and the Disabled option can be used if the control communication is via a spektrum satellite receiver or directly through telemetry.
  - **DISABLING**
    - **Disabled+OneShot** basically disables the ReceiverPort.

  Default settings

  By default, the Vehicle Setup Wizard will set receiver port as PPM_PIN8+OneShot when PPM type receiver is selected.

Note: Please note that the output rate on the output channels from the ReceiverPort cannot be set individually. If servos are connected to this outputs, you must ensure that they can work with the defined output rate for the bound channel. E.g. if you choose a high output rate to support an octocopter configuration, the update rate from the output channels from the ReceiverPort are bound to the update rate from channels 5 & 6. In this case, you cannot connect analogue servo's to these outputs since an analogue servo only supports an output rate of 50Hz. The output rates are set in GCS Outputs page.

Power

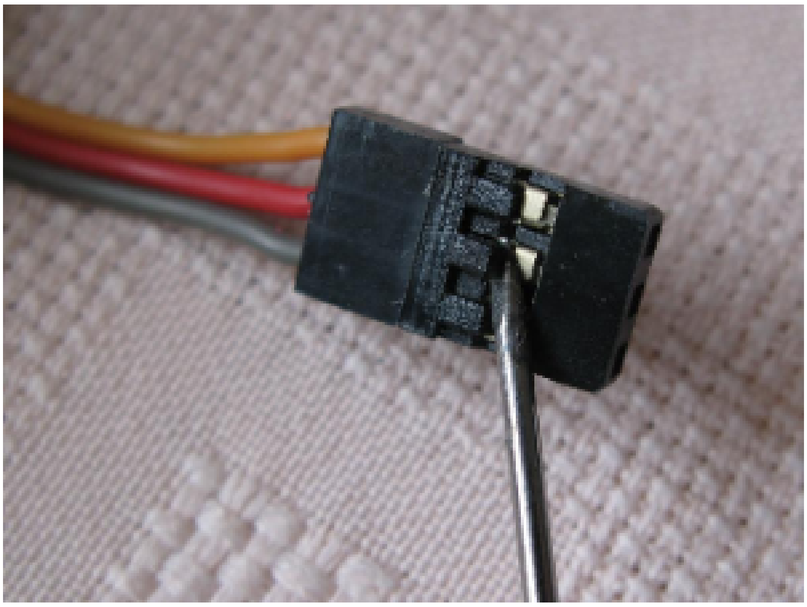
Warning: MAKE SURE YOU ARE CONNECTING POSITIVE AND NEGATIVE CORRECTLY.

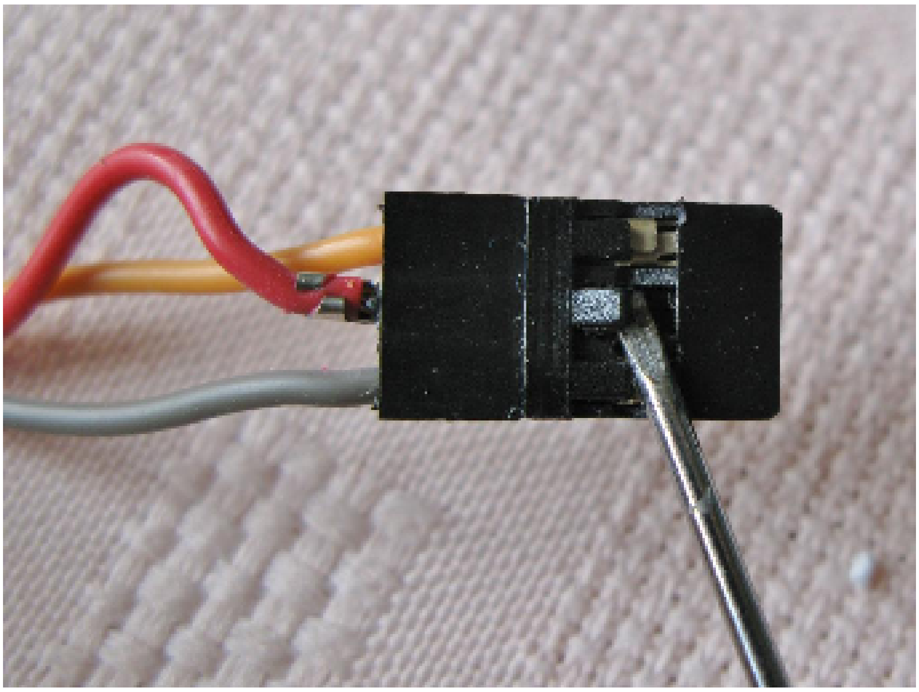
- CopterControl can be powered in several ways. Via the USB port, through the power pins on the servo headers or through the ReceiverPort connector (see the ports section for the port location). When powered by USB, peripherals connected (receiver, serial ports, servos, ESCs) will not be powered to protect your computer from too much current draw through the USB.
- The minimum allowed input voltage for CopterControl is 4.8V, the maximum allowed input voltage is +15V.
- Power consumption = ±70mA.
- You can connect the USB and the receiver (with the power) at the same time.

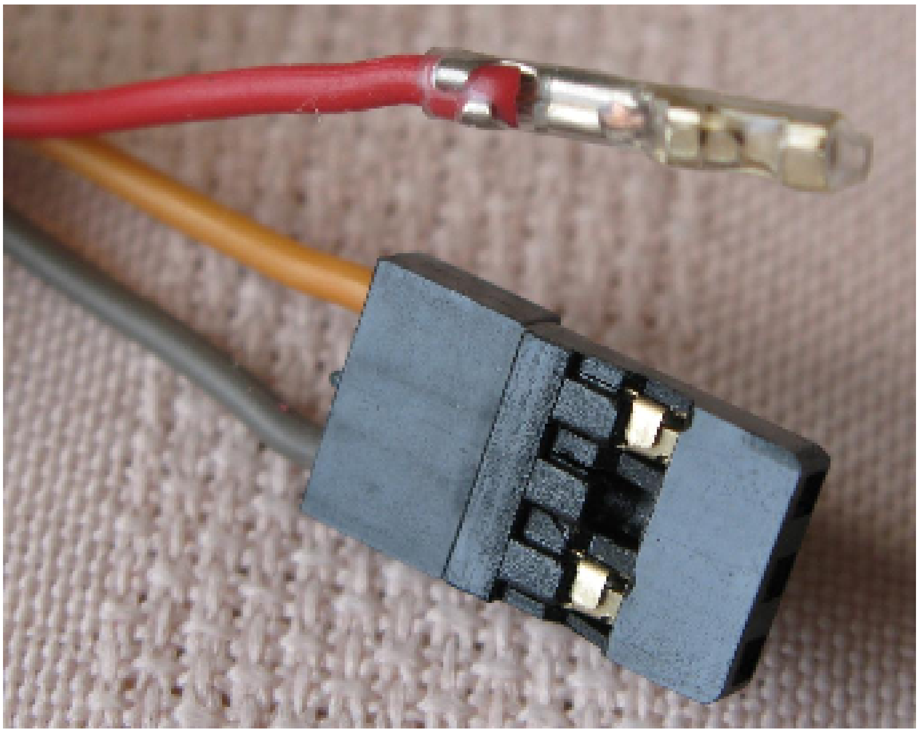
Caution: The PWR Out pins provide unregulated voltage to the ports. If the CC is powered from a +15V (max. allowed) source then +15V will be on the PWR Out pins and can damage connected receivers, GPS, telemetry modems or other add-on boards.

If you power the flight controller through the servo connectors (utilising the BEC function of the speed controller), the positive power lead from only one ESC is truly necessary. In most cases, all the wires can be left intact and plugged into the board without any problem. If you experience problems with setup or know for a fact that your particular ESC model requires it, you may remove the positive and negative pins from all but one of the ESC servo connectors. In some ESCs (very few, actually), connecting multiple voltage regulators (built in to the ESC's) in parallel could cause problems. Also, in rare cases, connecting multiple ground wires could cause ground loops so remove the extra ground pins only if experiencing weird problems.

These photos show how to remove and insulate the positive wire from the ESC. Remove the positive & negative wire leaving only the signal cable connected for all but one of your ESC's. A small flat blade screwdriver (or X-Acto knife could be used) and 2mm heat shrink tube was used in this example. This modification can easily be reversed by removing the heat shrink and inserting the positive wire back in to the ESC plug. Also, remove the ground wire when removing the hot and insulate separately from the hot wire.

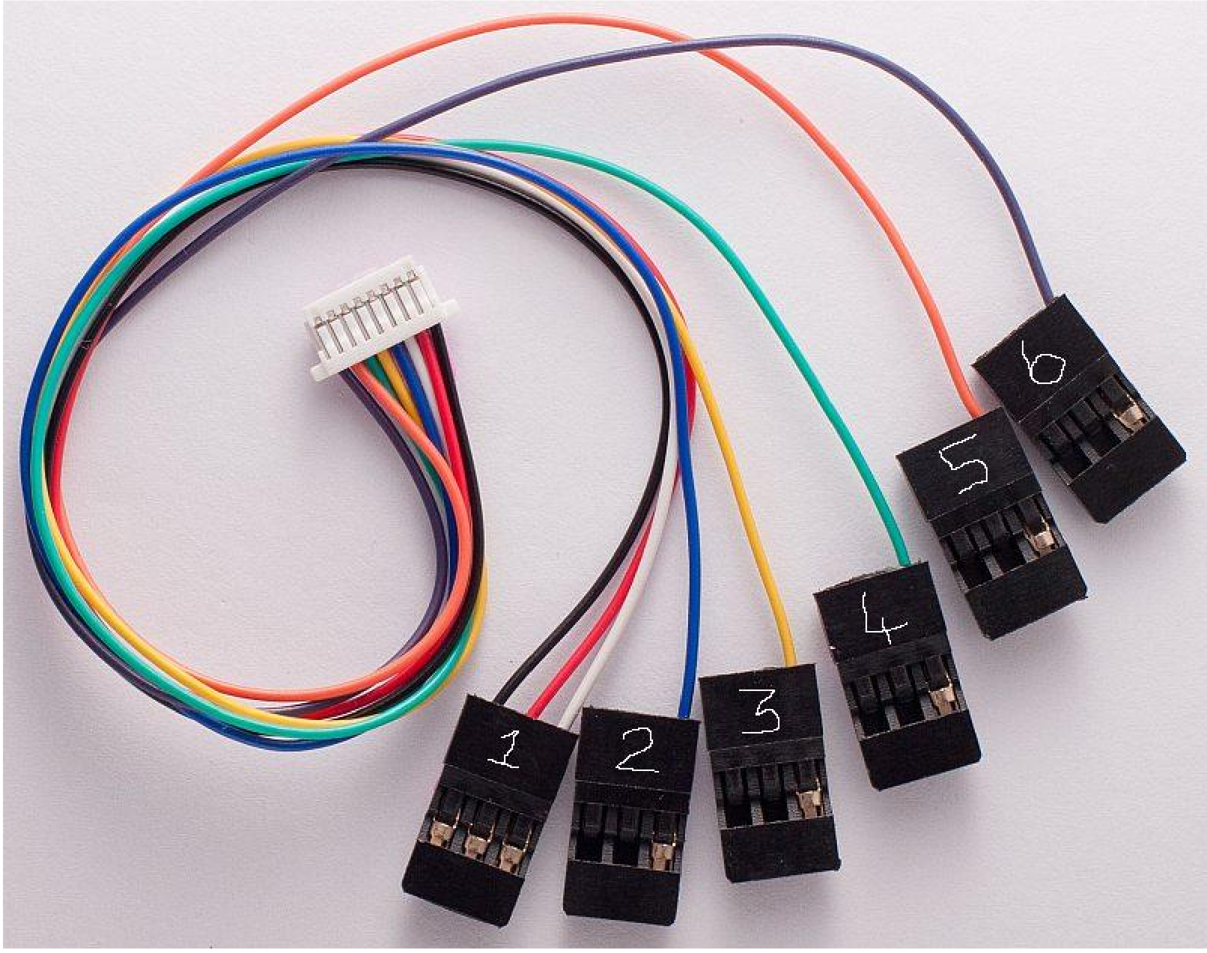






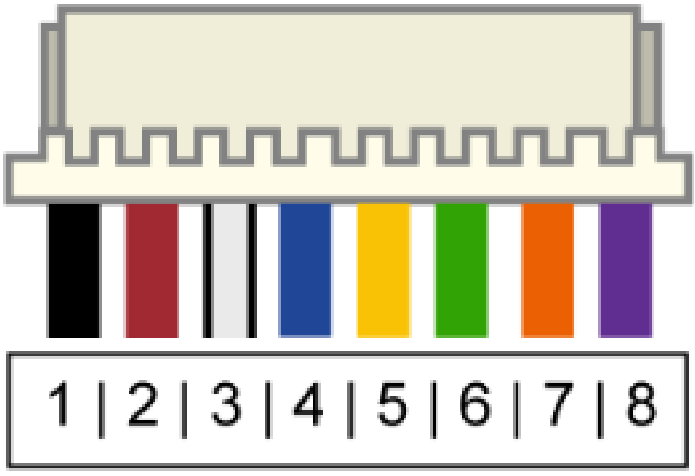
Cables, colors & pin-outs

CopterControl uses the JST-SH series headers. A CopterControl board comes standard with one 8-pin connection cable as shown below to connect your receiver. Additionally, one 4-pin JST-SH cable is supplied to connect to the MainPort or FlexiPort. You can easily cut the 4-pins cable and connect your telemetry or Spektrum satellite.



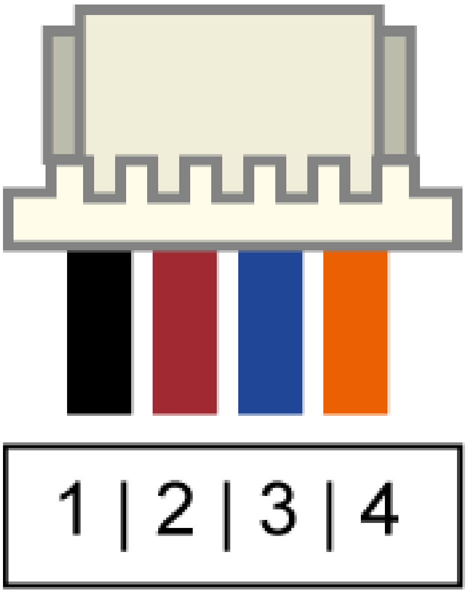
ReceiverPort

Pin	Color	PWM+NoOneShot Function	PPM+NoOneShot Function	PPM+PWM+NoOneShot Function	PPM+Outputs+NoOneShot Function	PPM_PIN8+OneShot Function	Outputs+OneShot Function
1	Black	GND	GND	GND	GND	GND	GND
2	Red	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)
3	White	PWM input 1	PPM input	PPM input	PPM input	—	—
4	Blue	PWM input 2	—	PWM input 2	—	—	—
5	Yellow	PWM input 3	—	PWM input 3	PWM output 7	—	PWM output 7
6	Green	PWM input 4	—	PWM input 4	PWM output 8	—	PWM output 8
7	Orange	PWM input 5	—	PWM input 5	PWM output 9	—	PWM output 9
8	Purple	PWM input 6	—	PWM input 6	PWM output 10	PPM input	PWM output 10



MainPort and FlexiPort serial cable pinout

Pin	Color	Voltage	Serial Function	I2C Function	Spektrum	S.Bus
1	Black	GND	GND	GND	GND	GND
2	Red	4.8V - 15V	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)
3	Blue	3.3V	TX	SCL	—	—
4	Orange	3.3V (5V Tolerant)	RX	SDA	TX (Signal)	TX (Signal)



Caution: The Spektrum adapter should only be powered by 3.3V, a step down adapter must be used.

Caution: The PWR Out voltage is dependent on the CC supplied voltage. Verify that you use the correct voltage for your S.BUS receiver.

Receiver PWM connection

There are several ways to connect your receiver to CopterControl. You can connect any plug from the CopterControl receiver cable to any channel of your receiver. The correct channel mapping is done in the GCS https://librepilot.atlassian.net/wiki/display/LPDOC/Input+Configuration. However as a guideline for a standard PWM receiver, you may want to connect it as follows:

For Futaba and Hitec

Channel 1	AILERON or ROLL	White	Signal 1
Channel 2	ELEV or PITCH	Blue	Signal 2
Channel 3	THROTTLE	Yellow	Signal 3
Channel 4	RUDDER	Green	Signal 4
Channel 5	GEAR - Flight mode	Orange	Signal 5
Channel 6	AUX1	Purple	Signal 6

For JR and Spektrum

Channel 1	THROTTLE	White	Signal 1
Channel 2	AILERON or ROLL	Blue	Signal 2
Channel 3	ELEV or PITCH	Yellow	Signal 3
Channel 4	RUDDER	Green	Signal 4
Channel 5	GEAR - Flight mode	Orange	Signal 5
Channel 6	AUX1	Purple	Signal 6

Note: If you are unsure about the type of your receiver (PPM, PWM, Spektrum Satellite...) or where to connect it, please refer to this page where the different options are explained.

Sensors and Components

- 3-axis Gyroscope array: IDG-500 and ISZ-500 [1]
- 3-axis Accelerometer: ADXL345 [1]
- Supports several common RC inputs: 6 PWM channels, combined PPM, Spektrum/JR DSM2, DSMJ, DSMX satellites, and Futaba S.Bus receivers
- Simultaneous support for multiple receivers
- ReceiverPort functions (configurable): 6 PWM input channels or combined PPM stream, 4 PWM output channels
- MainPort functions (configurable): serial telemetry (default), GPS, S.Bus, Spektrum/JR satellites
- FlexiPort (configurable): serial telemetry, GPS, Spektrum/JR satellites, or I2C peripherals (under development)
- 10 PWM outputs to servos or ESC's, or for camera stabilization
- Camera stabilization: supports up to 3-axis camera mounts with stabilization and manual control from any of configured receivers
- Onboard USB connectivity for easy configuration
- USB and serial telemetry and configuration (including wireless with optional radio modules)
- Supported by powerful OpenPilot GCS

[1] On CC3D the IDG-500, ISZ-500 and ADXL345 is replaced by the MPU6000.

- 4 Mbit onboard memory
- 3C Quaternion based complementary filter running at 500Hz

DIY Boards

Schematics, PCB Layout, Gerbers, BOM for **CopterControl**: `CopterControl.zip`

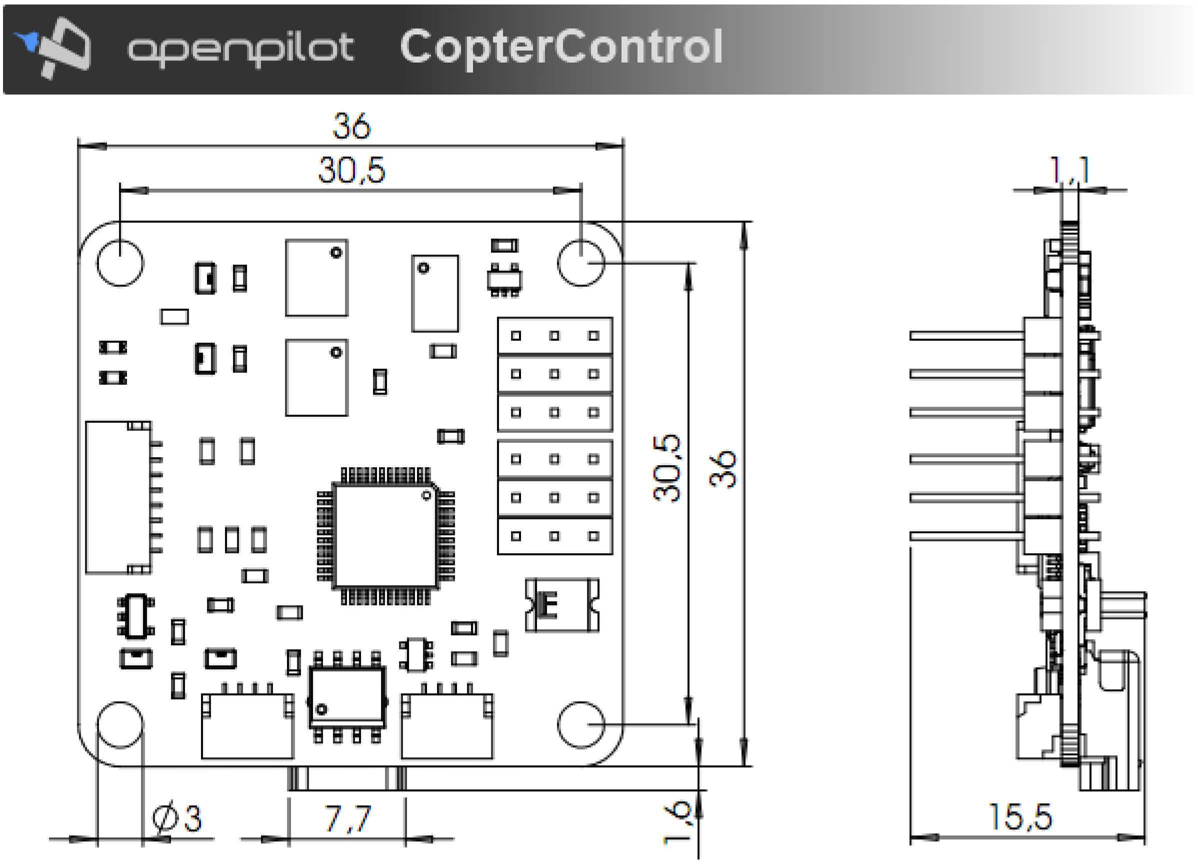
Schematics, PCB Layout, Gerbers, BOM for **CopterControl 3D**: `CopterControl 3D.zip`

Schematics, PCB Layout, Gerbers, BOM for **Atom**: `Atom.zip`

Other Information

Dimensions

CopterControl & CC3D used the standard OpenPilot footprint, and hence has the same dimensions and mounting holes as the OpenPilot Revo, GPS, OSD and PipX boards.



1.3.2 How to configure a GPS for CC3D

CC3D or Atom flight controllers do not support GPS functions such as Return to base or Position Hold. The original CopterControl (CC) board does not have memory left for GPS use.

The only purpose for GPS use is telemetry using radio modems (OPLink) or GPS position send to OSD device.

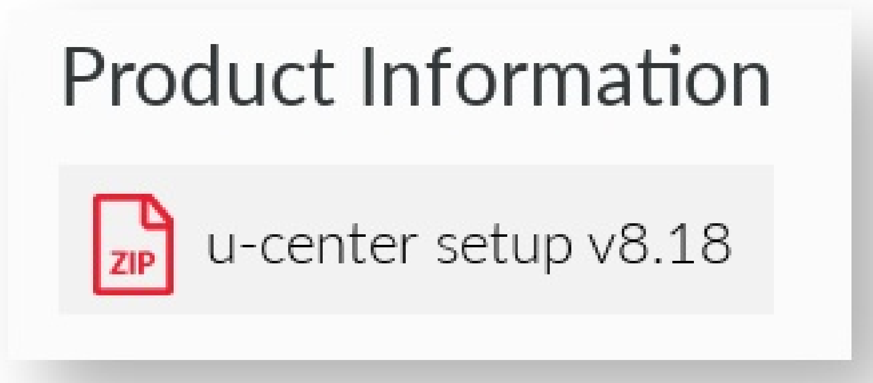
NMEA is not supported anymore because is less memory efficient, this page only refers to UBlox GPS and UBX protocol.

Note: Others boards like Revolution support **auto-configuration** and **no need** this manual configuration using UCenter.

This means also if you connect your 'CC3D GPS' to a Revolution board, your previous configuration is lost!

Prerequisites

- Download the UCenter software here, check Zip icon in bottom page.

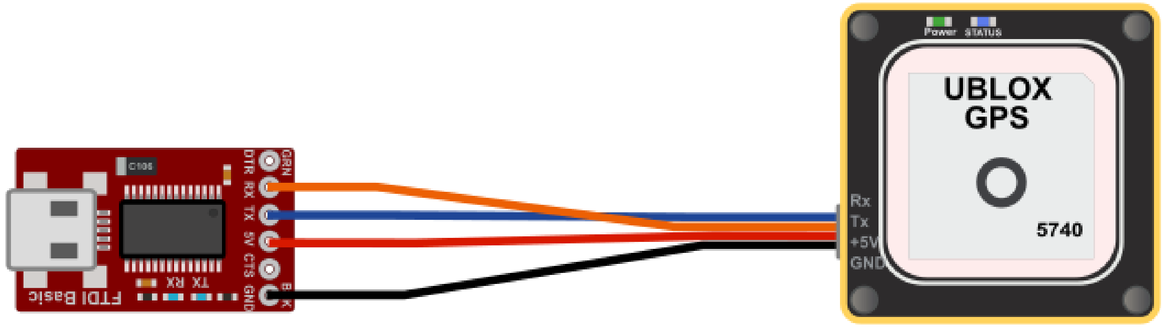


Product Information

ZIP u-center setup v8.18

Note: UCenter is for Windows only.

- Install UCenter software
- Setup a serial connection with GPS using USB port (if any) or USBSerial (Gnd, +5V, Tx, Rx).

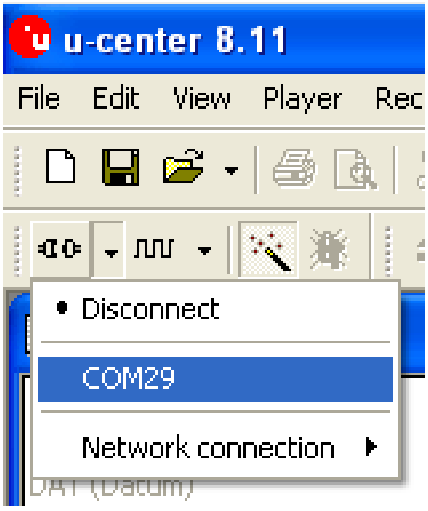


Configure your GPS

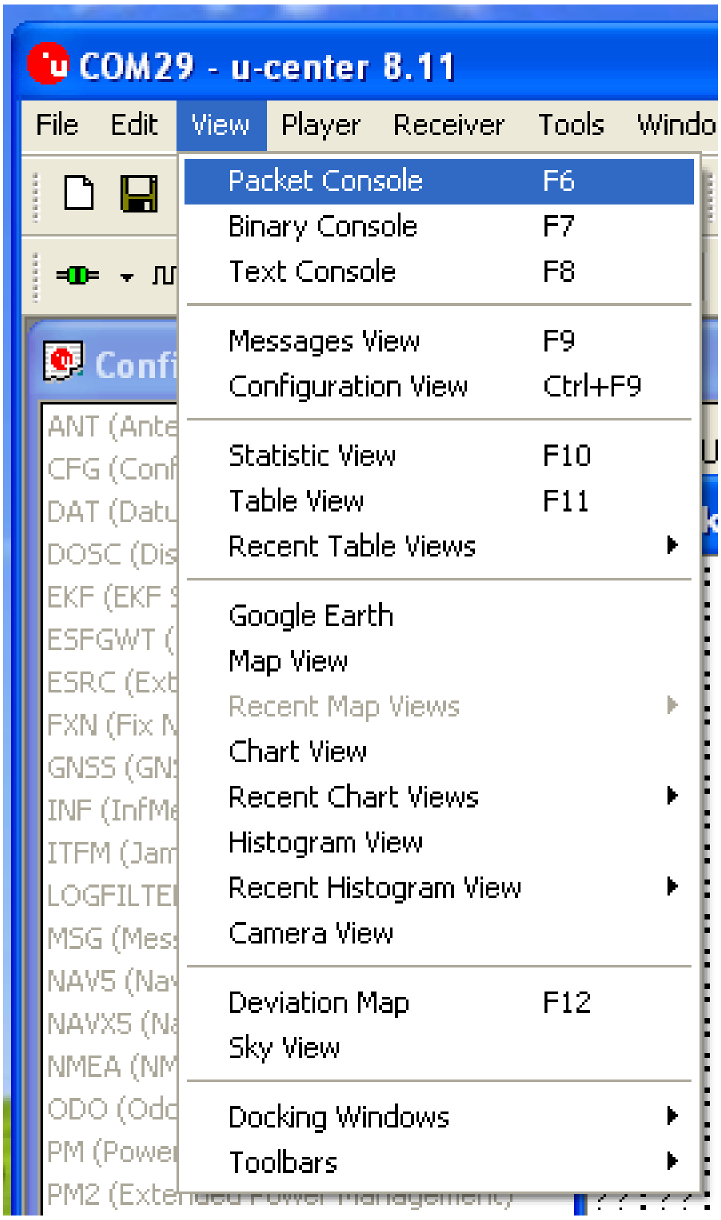
Connect

- Connect GPS to USB serial or USB.
- Start UCenter software.

- Select COM port used for GPS.

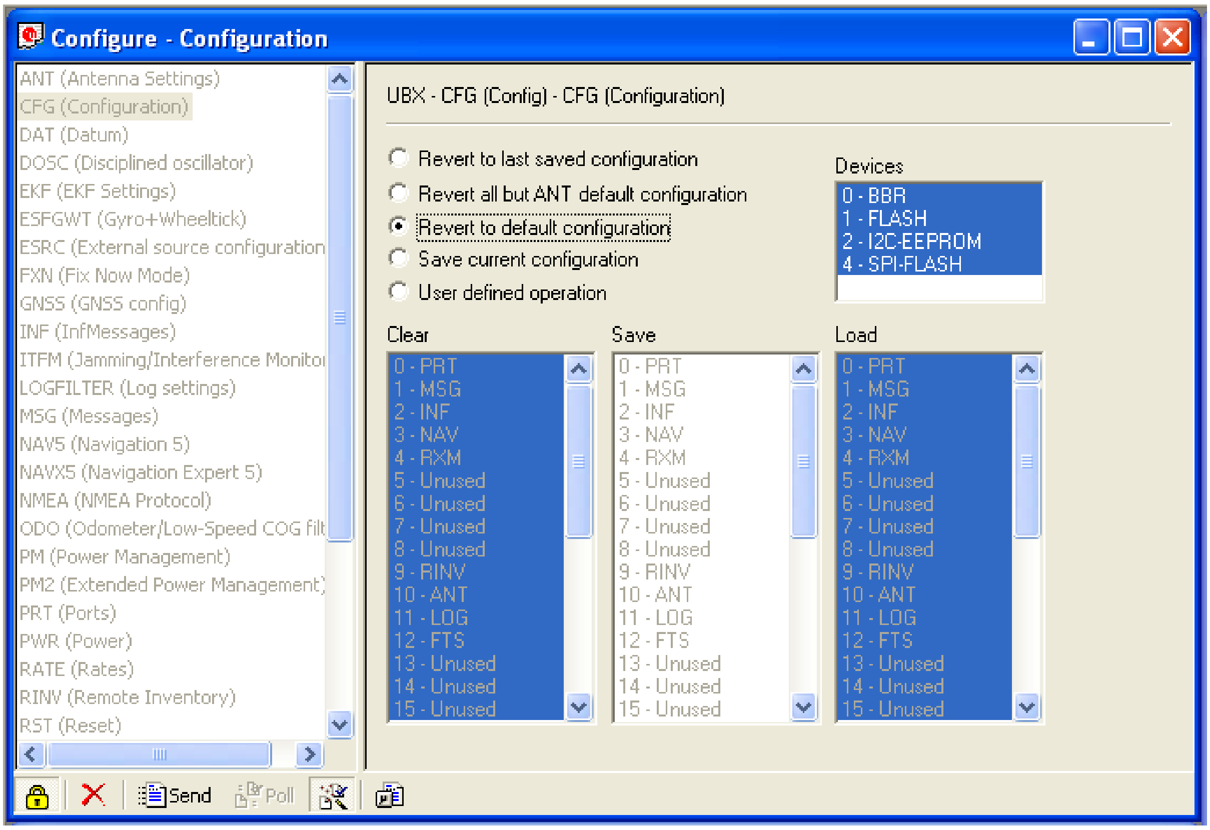


- Open console (F6 key) or **View** > **Packet Console** and check if link is correct, you should see messages output from GPS.



Reset to defaults

- Open Configuration View window: **View** > **Configuration View**.
- Reset current GPS config to defaults: Check **Revert to default configuration**.



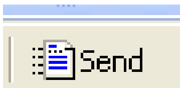
UBX - CFG (Config) - CFG (Configuration)

○ Revert to last saved configuration
○ Revert all but ANT default configuration
● Revert to default configuration
○ Save current configuration
○ User defined operation

Devices

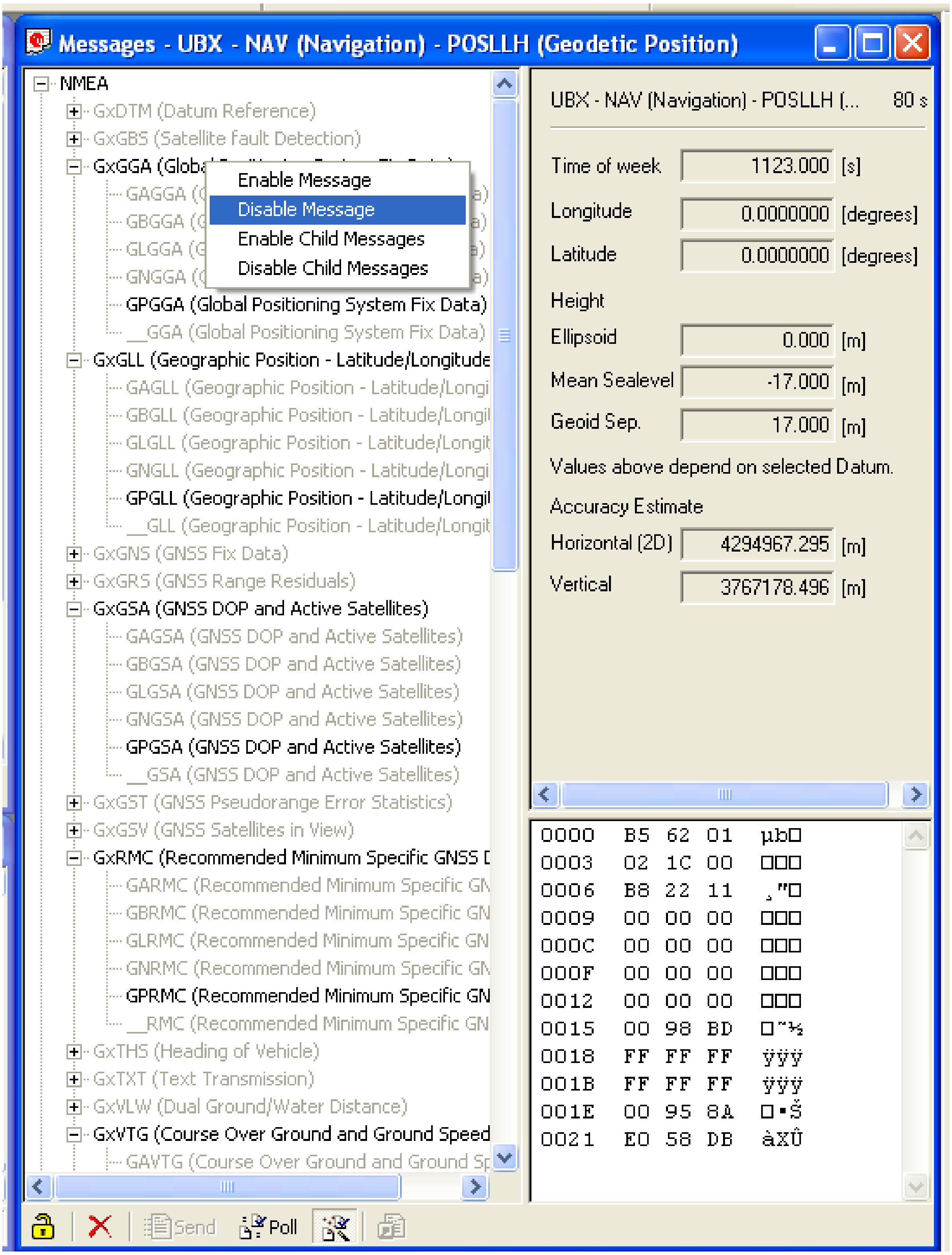
0 - BBR
1 - FLASH
2 - I2C-EEPROM
4 - SPI-FLASH

Clear	Save	Load
0 - PRT	0 - PRT	0 - PRT
1 - MSG	1 - MSG	1 - MSG
2 - INF	2 - INF	2 - INF
3 - NAV	3 - NAV	3 - NAV
4 - RXM	4 - RXM	4 - RXM
5 - Unused	5 - Unused	5 - Unused
6 - Unused	6 - Unused	6 - Unused
7 - Unused	7 - Unused	7 - Unused
8 - Unused	8 - Unused	8 - Unused
9 - RINV	9 - RINV	9 - RINV
10 - ANT	10 - ANT	10 - ANT
11 - LOG	11 - LOG	11 - LOG
12 - FTS	12 - FTS	12 - FTS
13 - Unused	13 - Unused	13 - Unused
14 - Unused	14 - Unused	14 - Unused
15 - Unused	15 - Unused	15 - Unused

- Send config: 

Select set of messages

- Open Messages View window: **View** > **Messages View**.
- **Disable** all active NMEA messages.


Messages - UBX - NAV (Navigation) - POSLLH (Geodetic Position)
NMEA
GxDTM (Datum Reference)
GxGBS (Satellite Fault Detection)
GxGGA (Global Positioning System Fix Data)
GAGGA
GBGGA
GLGGA
GNGGA
GPGGA (Global Positioning System Fix Data)
__GGA (Global Positioning System Fix Data)
Enable Message
Disable Message
Enable Child Messages
Disable Child Messages
GxGLL (Geographic Position - Latitude/Longitude
GAGLL (Geographic Position - Latitude/Longi
GBGLL (Geographic Position - Latitude/Longi
GLGLL (Geographic Position - Latitude/Longit
GNGLL (Geographic Position - Latitude/Longi
GPGLL (Geographic Position - Latitude/Longi
__GLL (Geographic Position - Latitude/Longit
GxGNS (GNSS Fix Data)
GxGRS (GNSS Range Residuals)
GxGSA (GNSS DOP and Active Satellites)
GAGSA (GNSS DOP and Active Satellites)
GBGSA (GNSS DOP and Active Satellites)
GLGSA (GNSS DOP and Active Satellites)
GNGSA (GNSS DOP and Active Satellites)
GPGSA (GNSS DOP and Active Satellites)
__GSA (GNSS DOP and Active Satellites)
GxGST (GNSS Pseudorange Error Statistics)
GxGSV (GNSS Satellites in View)
GxRMC (Recommended Minimum Specific GNSS D
GARMC (Recommended Minimum Specific GN
GBRMC (Recommended Minimum Specific GN
GLRMC (Recommended Minimum Specific GN
GNRMC (Recommended Minimum Specific GN
GPRMC (Recommended Minimum Specific GN
__RMC (Recommended Minimum Specific GN
GxTHS (Heading of Vehicle)
GxTXT (Text Transmission)
GxVLW (Dual Ground/Water Distance)
GxVTG (Course Over Ground and Ground Speed
GAVTG (Course Over Ground and Ground Sp
Send
Poll

UBX - NAV (Navigation) - POSLLH (... 80 s

Time of week 1123.000 [s]

Longitude 0.0000000 [degrees]

Latitude 0.0000000 [degrees]

Height

Ellipsoid 0.000 [m]

Mean Sealevel -17.000 [m]

Geoid Sep. 17.000 [m]

Values above depend on selected Datum.

Accuracy Estimate

Horizontal (2D) 4294967.295 [m]

Vertical 3767178.496 [m]

```
0000  B5 62 01  µb□
0003  02 1C 00  □□□
0006  B8 22 11  ¸"□
0009  00 00 00  □□□
000C  00 00 00  □□□
000F  00 00 00  □□□
0012  00 00 00  □□□
0015  00 98 BD  □˜½
0018  FF FF FF  ÿÿÿ
001B  FF FF FF  ÿÿÿ
001E  00 95 8A  □•Š
0021  E0 58 DB  àXÛ
```

- **Enable** UBX messages needed:
  - **POSLLH**
  - **DOP**
  - **SOL**
  - **VELNED**

- Save current configuration using a little icon with a gear on top.

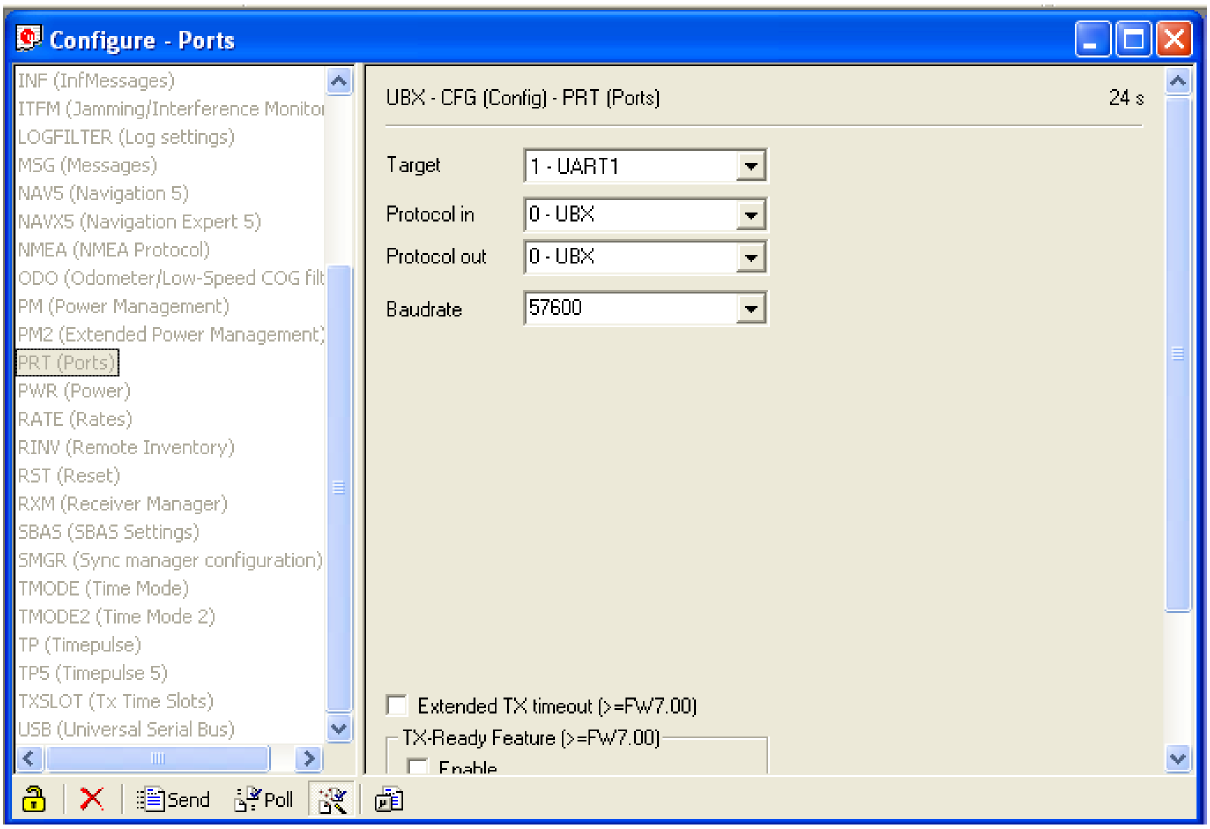
Note: Using this configuration you can’t display satellite in GPS plugin because GPS don’t send this data (usually SVINFO message).

CC3D works only with minimal messages. (POSLLH, DOP, SOL, VELNED)

Setup port and protocol

- Open Configuration View window: **View > Configuration View**. Move to **PRT (Ports)** in left column.
- Change message output in UART1: only UBX for protocol in/out.
- Change baudrate for UART1 port, same baudrate set for Flexi/Main GPS port.

Caution: You should use **9600** or **19200** baudrate for CC3D, or packets are dropped at higher speed.



Configure - Ports

UBX - CFG (Config) - PRT (Ports) 24 s

Target: 1 - UART1

Protocol in: 0 - UBX

Protocol out: 0 - UBX

Baudrate: 57600

□ Extended TX timeout (>=FW7.00)

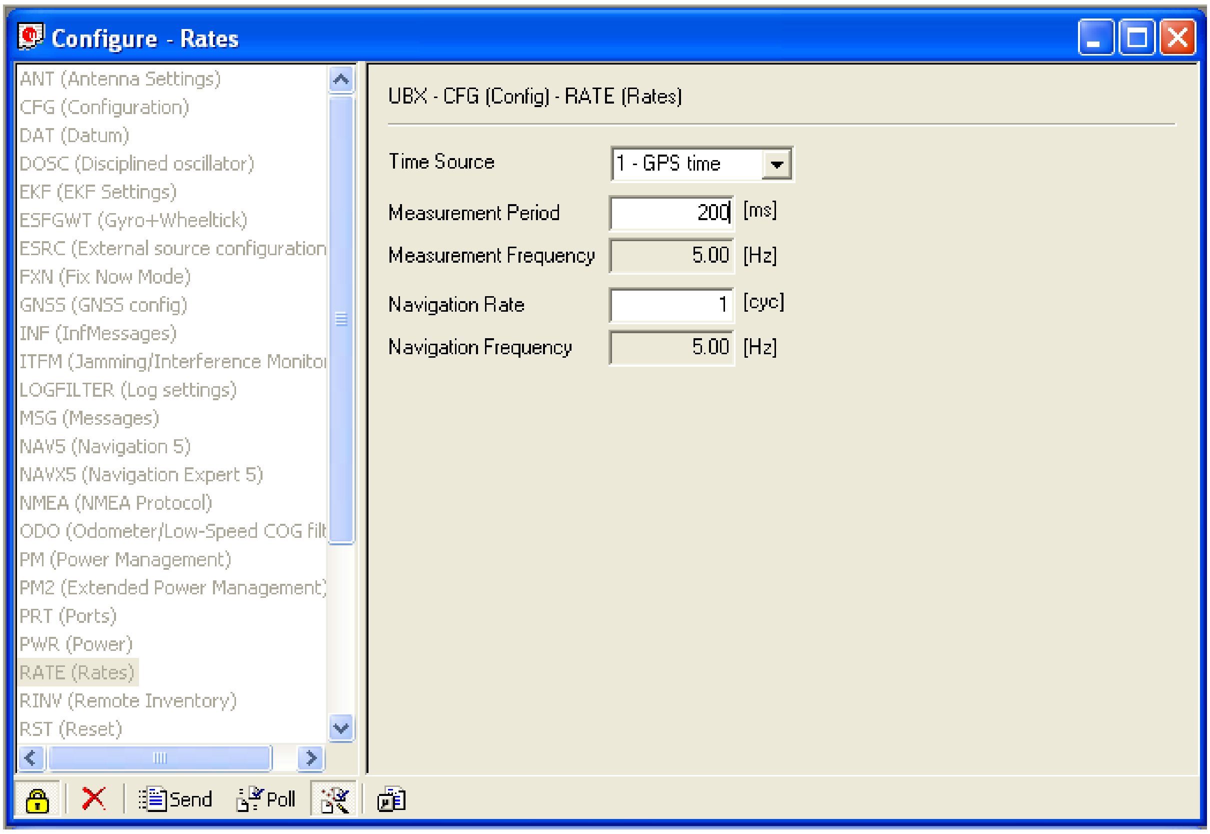
TX-Ready Feature (>=FW7.00)

□ Enable

- Send config: Send

Change refresh rate and save

- Move to **Rate (Rates)** in left column
- Change **Measurement Period** value to 200ms for a 5Hz refresh rate.



UBX - CFG (Config) - RATE (Rates)

Time Source: 1 - GPS time

Measurement Period: 200 [ms]

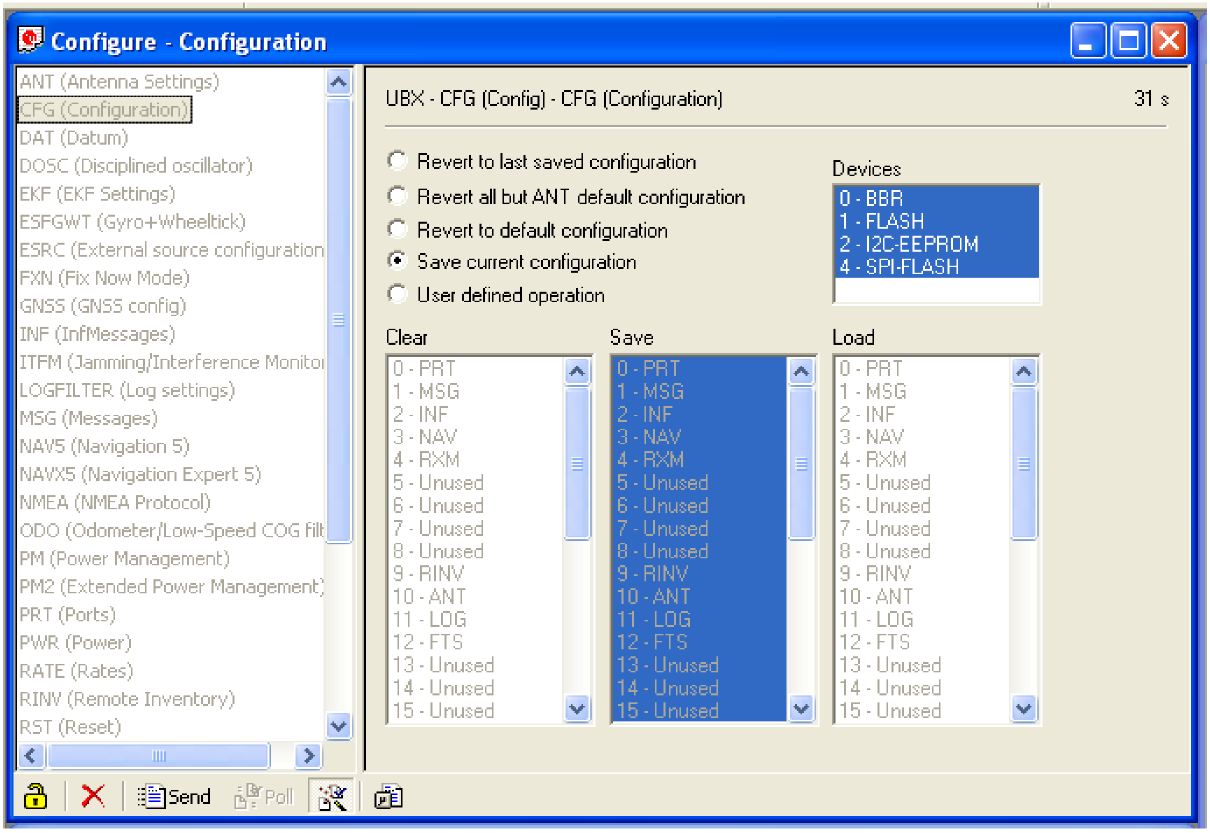
Measurement Frequency: 5.00 [Hz]

Navigation Rate: 1 [cyc]

Navigation Frequency: 5.00 [Hz]

- Send config:

- Save current configuration: Check **Save current configuration**.
- All devices selected in blue.



UBX - CFG (Config) - CFG (Configuration) 31 s

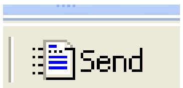
○ Revert to last saved configuration
○ Revert all but ANT default configuration
○ Revert to default configuration
● Save current configuration
○ User defined operation

Devices
0 - BBR
1 - FLASH
2 - I2C-EEPROM
4 - SPI-FLASH

Clear
0 - PRT
1 - MSG
2 - INF
3 - NAV
4 - RXM
5 - Unused
6 - Unused
7 - Unused
8 - Unused
9 - RINV
10 - ANT
11 - LOG
12 - FTS
13 - Unused
14 - Unused
15 - Unused

Save
0 - PRT
1 - MSG
2 - INF
3 - NAV
4 - RXM
5 - Unused
6 - Unused
7 - Unused
8 - Unused
9 - RINV
10 - ANT
11 - LOG
12 - FTS
13 - Unused
14 - Unused
15 - Unused

Load
0 - PRT
1 - MSG
2 - INF
3 - NAV
4 - RXM
5 - Unused
6 - Unused
7 - Unused
8 - Unused
9 - RINV
10 - ANT
11 - LOG
12 - FTS
13 - Unused
14 - Unused
15 - Unused

- Send config: 

1.4 Revolution Board Setup

1.4.1 Revolution Board Setup

Introduction

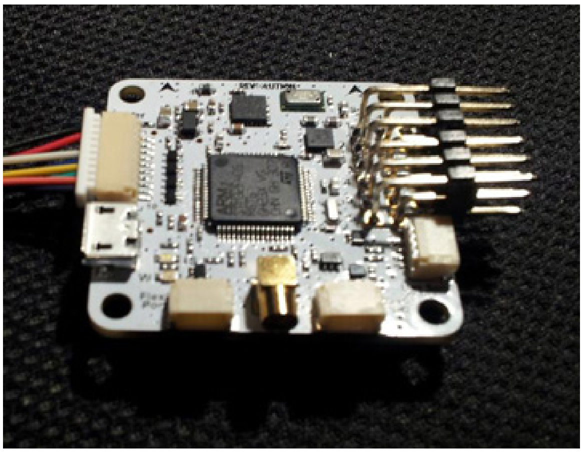
Welcome to the Revolution (Revo) board setup page. Here you will find a number of video tutorials to assist you in setting up your new Revo board.

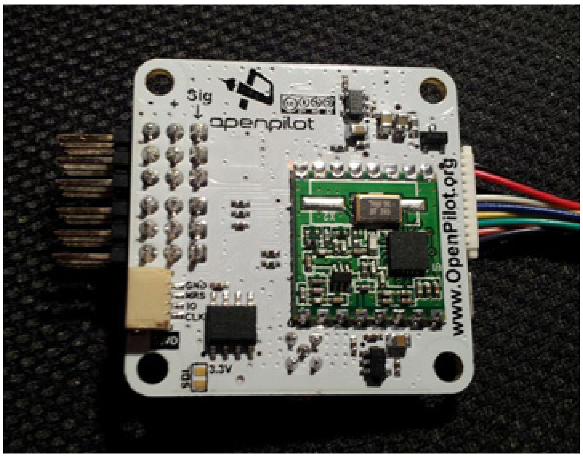
Along with the tutorials you will also find links to the relevant Wiki pages to assist you in setting up the best Flight Controller board on the market.

WELCOME TO THE OP EXPERIENCE.

The OpenPilot Revolution board, also called 'Revo', is a new breed of Autopilot using the STM32F4 series, 210MIPS ARM Micro-controller. This is important, as it contains a hardware floating point unit (FPU), which is a huge advancement for hobby-class autopilots. Of course, OpenPilot has been 32bit since day one, and the FPU is another step up the performance ladder. The FPU allows precise, low-latency processing of real-life measurements using advanced attitude estimation algorithms.

The Revolution is a flight control computer with autopilot, intended for multirotors, helicopters and fixed wings. It is a full 10DOF with gyroscope, accelerometer, magnetometer and pressure sensors.





Setting up your Revolution Board for the first time

You have just received a brand new Revolution board and are itching to mount it in your air frame, follow the Video Tutorial below.

https://vimeo.com/64065065

Vehicle Setup Wizard

Once you have mounted your Revo on your frame you need to configure it through the Ground Control Station (GCS) using the Vehicle Setup Wizard, follow the **Video Tutorial** below for setting up on a Multirotor.

https://vimeo.com/65454262

Sensor calibration

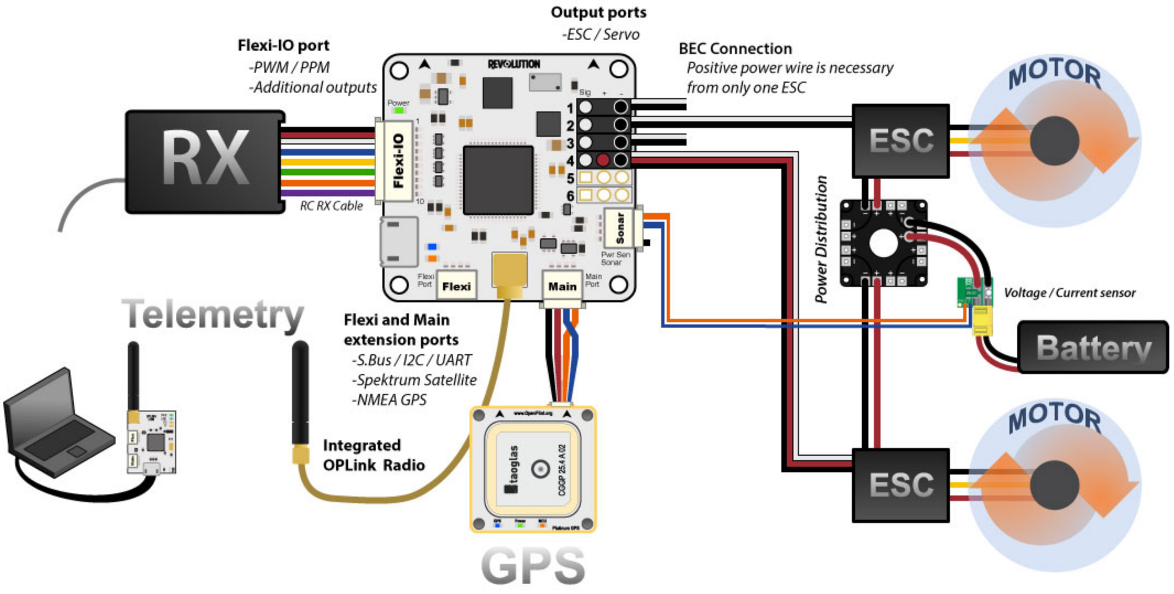
Revo - Sensor Calibration.

Tuning your Revolution

LibrePilot Wiki

Connection diagram

The diagram below summarizes how the overall Revolution system is connected.



Technical description

CPU

CPU is the STM32F405RGT6 chip, with ARM Cortex-M4 core at 210MIPS, FPU, and saturation arithmetics DSP functions.

The chip features a range of built-in hardware modules that can bo programmed once and function independentely, requiring little to no CPU overhead. These include 14x multichanel timers, 3x synchronous-sampling ADC serving up to 24 channels, 2x DAC, matrix memory controller with 16-stream DMA, and other. Communication modules include USB2.0, 3x I2C, 3x SPI, 4x USART, 2x CAN and SIDO. All these modules can be configured for accessing the chip pins using a flexible switch matrix, or disabled to save power.

It even contains a real time hardware calendar if you want a wake up flight.

The software and settings are loaded through USB connector and no-hassle update function in the GCS (Ground Control Station).

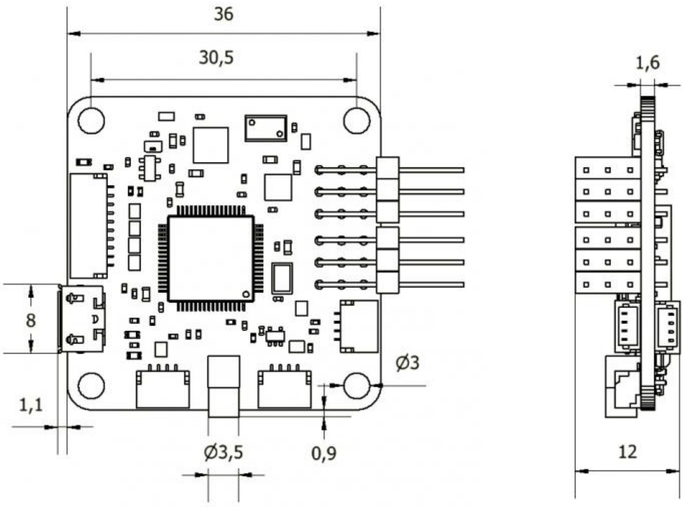
Modem

The board features built-in 433MHz OPLink Modem.

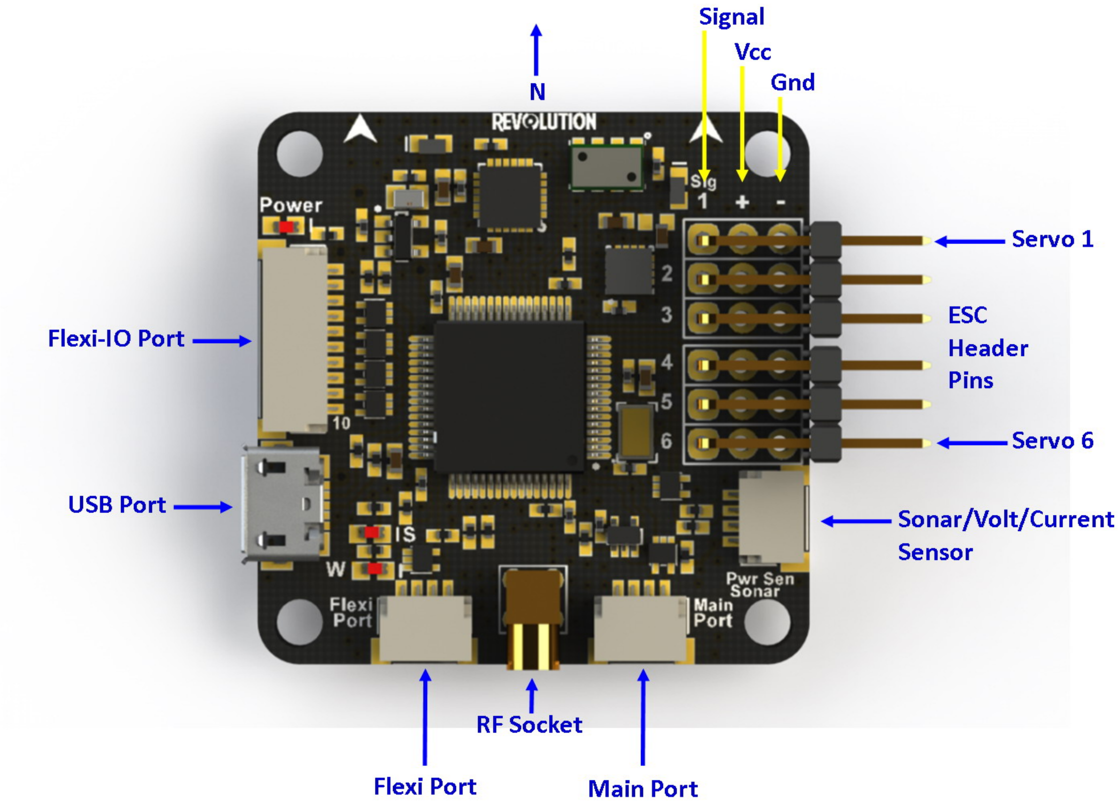
Dimensions

OpenPilot products use the standard OpenPilot footprint, and hence has the same dimensions and mounting holes as the OpenPilot Revo, GPS, OSD and PipX boards.

(All dimensions are in millimeters.)



Ports



- **Servo 1-6**: These are the PWM outputs that go to servos or ESCs. Power is typically applied through these headers from only one of the ESCs. The positive (Vcc) and negative (Gnd) pins are indicated on this diagram and the board.

  Servo output pin layout is:

  - Outside –> ground
  - Middle –> 5V - 8.4V
  - Inside –> signal

- **Flexi-IO Port**: JST-SH 10-pin. The receiver port can act as an input or output port depending on the configuration which is set in the Hardware Settings. Configuring the receiver port as an output port allows the user to assign more output channels then the 6 standard servo outputs.

  PWM -vs- PPM Recievers

  Please be aware that not all receivers can be configured to use a PPM output. It is the user's responsibility to research this feature in regards to the desired receiver they wish to use for PPM and ensure it can be used as such. Many hours of frustration can occur while trying to troubleshoot why you can't get your radio to connect to the board with PPM if using a receiver than isn't designed with that feature! Simply make sure the receiver can do it before trying to set it up that way.

- **MainPort**: JST-SH 4-pin. This is a serial USART whose baud rate can be adjusted through the GCS. Optionally, Futaba S.Bus receiver, Spektrum/JR satellite receiver or GPS can be mapped to the MainPort. Default configuration is Telemetry for connecting an RF modem.

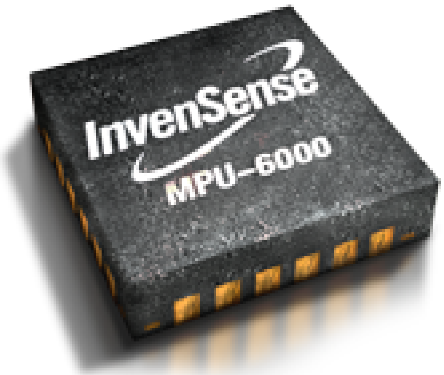
- **FlexiPort**: JST-SH 4-pin. The function of this port also depends on the configuration and can be configured for I2C or Serial. The default configuration doesn't use this port, but it can be used for Telemetry, GPS, Spektrum satellite receivers (all working), and other I2C peripherals (under development).

- **RF Socket**: Antenna connection socket for on-board OPLink Modem.
- **Pwr Sen/Sonar Port**: JST-SH 4-pin. This port can be configured to accommodate an Autopilot current sensor and a low cost Sonar sensor such as the HC-SR04. It can also be used as a general purpose input/output port or as a one or two channel analog input port.

Note: Please note that the output rate on the output channels from the ReceiverPort cannot be set individually. If servos are connected to this outputs, you must ensure that they can work with the defined output rate for choose a high output rate to support an octocopter configuration, the update rate from the output channels from the ReceiverPort are bound to the update rate from channels 5 & 6. In this case, you cannot connect analog servo's to these outputs since an analog servo only supports an output rate of 50Hz.

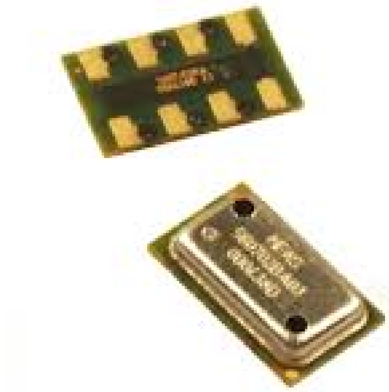
Sensor suite

- 3 Axis Gyro
- 3 Axis Accelerometer
- 3 Axis Magnetometer
- Barometric pressure sensor



MPU

The MPU-6000 combines a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die. This sensor can also be found on the CC3D and already has a proven track record of great flight performance.



Pressure Sensor/Altimeter

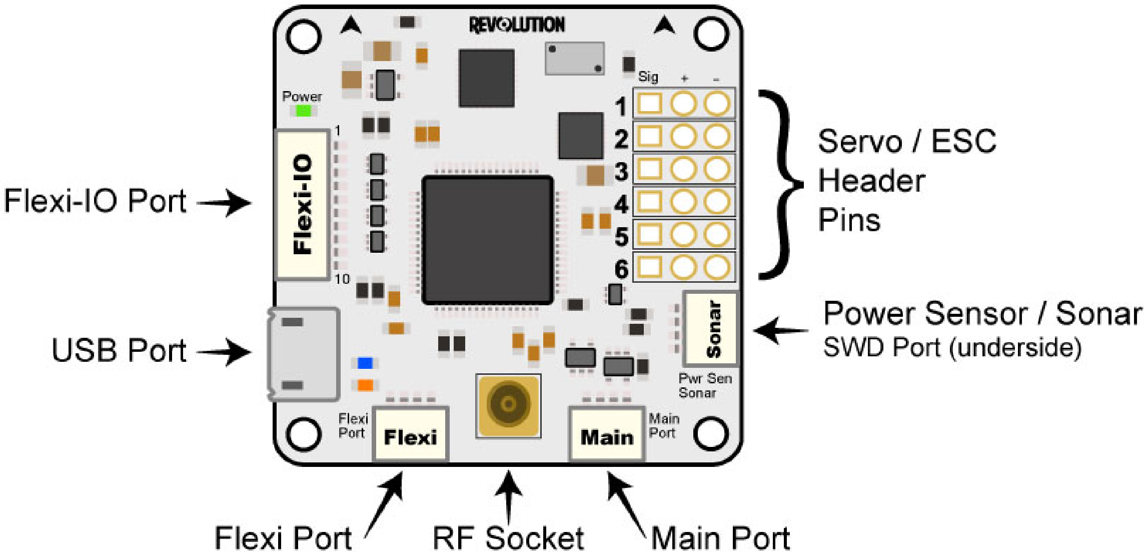
When it came time to select a barometric pressure sensor, there were many to choose from and many were not up to the task. For the Revolution, OpenPilot selected the Measurement Specialties MS5611.

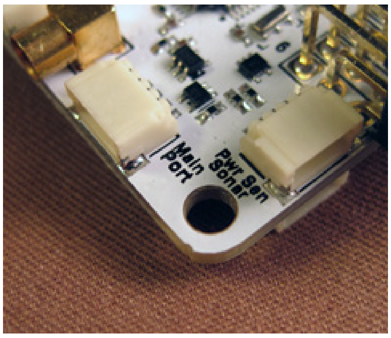
The MS5611 is not just any barometric pressure sensor, it is purpose-built, and has a very high resolution. As a result, it is ideal for use as a UAV altitude sensor. The sensor is so sensitive that it can sense a vertical shift of only 10 cm. The Revolution uses the newer MS5611-01BA03 version of this sensor that is far less susceptible to light interference than the older, plastic case versions.

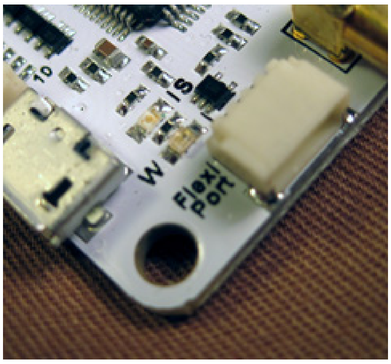
Magnetometer Whether you are in 'Position Hold' while taking aerial photographs, or you are flying a fixed-wing UAV on a pre-planned flight path, it's vitally important to have accurate heading information. The Honeywell HMC5883L is a three-axis digital compass module which provides rapid updates to changes in orientation which are accurate to a tolerance of 1° to 2°.

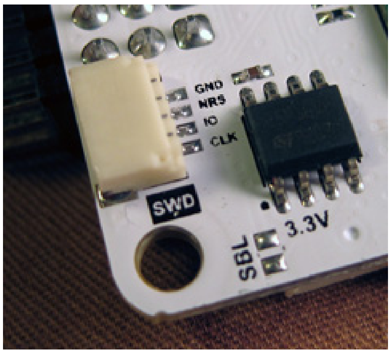
Connectivity

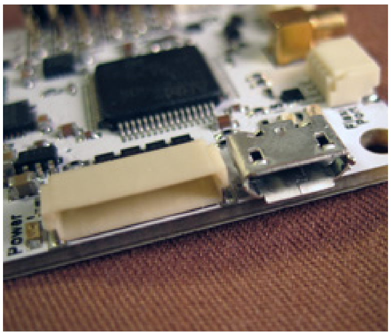
Just like CC & CC3D, the Revo has many ports, but a key new addition is the Flexi-IO Port. A lot of thought went into creating a small device that's flexible for use with multirotor platforms, helicopters and fixed wing aircraft, as well as making connectivity as future proof as possible.











Flexi-IO

The Flexi-IO port uses a 10 pin JST-connector and is designed to perform several different functions. The port is used for control input and output and can be configured to serve 6 PWM inputs or outputs. A mixture of other modes can also be configured.

- 6 channel PWM input for receiver + 2 GPIO
- 1 channel PPM input for receiver + 5 PWM I/O channels for motor/servo or sensors
- 4 pin SPI + 1 UART + 2 PWM I/O (PPM mode)
- CAN with external transceiver, UART, 4 PWM I/O / PPM

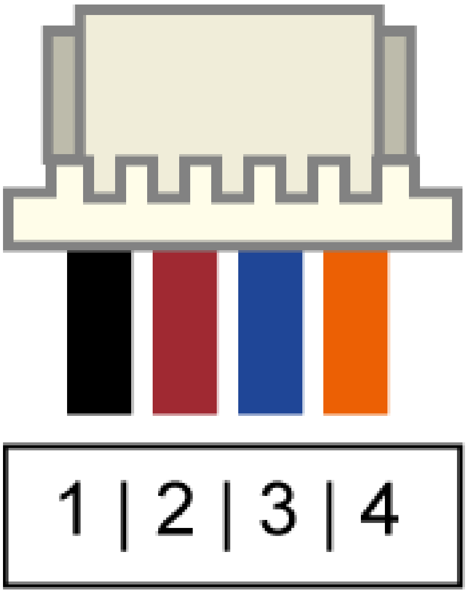
The full unshared SPI I/O port can be used as master or slave, allowing you to interface the Revolution board to any other embedded system like, for example, the Overo Linux boards. This also allows use of the Revo for advanced robotic applications.

An additional board can provide raw CPU power and high level functions, ranging from networking and data storage, up to video analysis, autonomous behaviour and artificial intelligence. The Revolution board will act as a reliable real time hardware controller and sensor platform, keeping your autonomous vehicle safely under control at all times.

Pinout

Pin	Color	PWM Function	PPM Function	PPM+Telemetry Function	PPM+Outputs Function	Outputs Function	Telemetry Function
1	Black	Ground	Ground	Ground	Ground	Ground	Ground
2	Red	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
3							
4							
5	White	PWM Input 1	PPM Input 1-8	PPM Input 1-8	PPM Input	PWM Output 12	
6	Blue	PWM Input 2			PWM Output 7	PWM Output 7	
7	Yellow	PWM Input 3		Telemetry TX	PWM Output 8	PWM Output 8	Telemetry TX
8	Green	PWM Input 4		Telemetry RX	PWM Output 9	PWM Output 9	Telemetry RX
9	Orange	PWM Input 5			PWM Output 10	PWM Output 10	
10	Violet	PWM Input 6			PWM Output 11	PWM Output 11	

FlexiPort



The Revo uses the same FlexiPort as the CC3D. The port can be used as either a UART or for I2C bus connectivity. It can be connected to serial devices like the OP GPS or any I2C device like the the EagleTree Airspeed expander module, ADCs, I2C ESCs and a lot more. It can also be used to connect Spektrum DSM2/DSMX Satellite to be used as receiver, or any other custom component interfacing with I2C or a serial connection including custom extension boards. Of course, it’s also possible to run a serial Telemetry link to the GCS over the FlexiPort.

Pinout

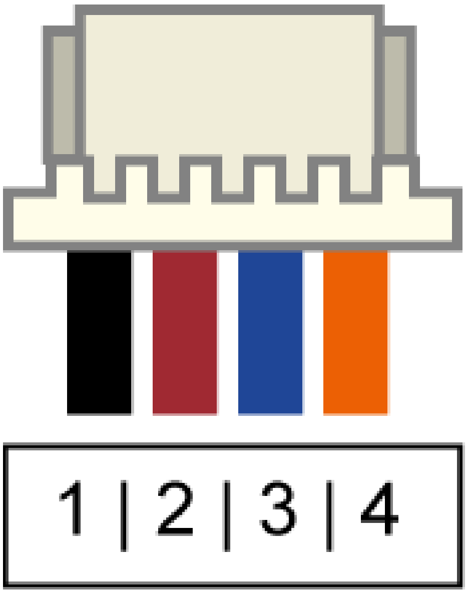
Color	JST-SH Pin	Voltage	Serial Function (GPS, Telemetry)	I2C Function	DSM
Black	1	GND	GND	GND	GND
Red	2	4.8V - 15V	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)
Blue	3	3.3V	TX	SCL	
Orange	4	3.3V (5V Tolerant)	RX	SDA	TX (Signal)

Warning: The Spektrum adapter should only be powered by 3.3V, a step down adapter must be used.

Warning: The PWR Out voltage is dependent on the CC supplied voltage. Verify that you use the correct voltage for your S.BUS receiver.

MainPort

Standard serial port/S.Bus port (same as CC3D). This can be used to connect serial devices like Telemetry, OP GPS, Futaba S.Bus receivers or Spektrum DSM2/DSMX satellites (to be used as a receiver), freeing in these cases the Flexi-IO port for other uses. These systems use a single wire to help cut down cable clutter.

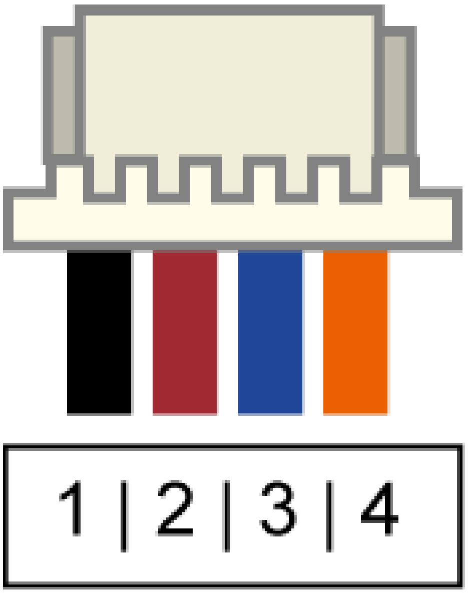


Pinout

Color	JST-SH Pin	Voltage	Serial Function (GPS, Telemetry)	DSM	S.BUS
Black	1	GND	GND	GND	GND
Red	2	4.8V - 15V	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)	PWR Out (VCC Unregulated)
Blue	3	3.3V	TX		
Orange	4	3.3V (5V Tolerant)	RX	TX (Signal)	TX (Signal)

Current / Sonar

This port can be configured to accommodate an Autopilot current sensor and a low cost Sonar sensor such as the HC-SR04. It can also be used as a general purpose input/output port or as a one or two channel analog input port.



Pinout

Color	JST-SH Pin	Voltage	Power Sensor
Black	1	GND	GND
Red	2	4.8V - 15V	PWR Out (VCC Unregulated)
Blue	3	3.3V	Current Input
Orange	4	3.3V (5V Tolerant)	Voltage Input

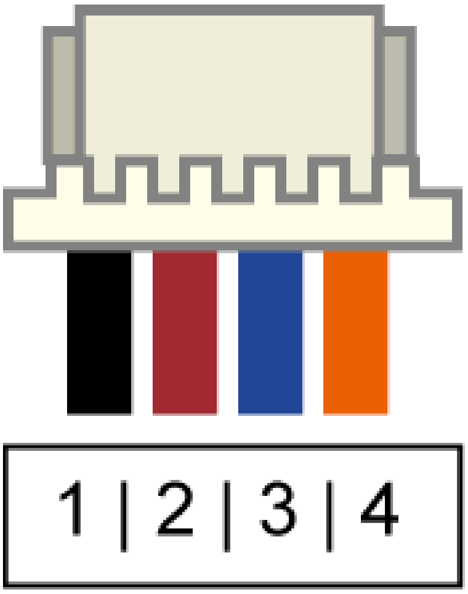
See also: Revo - Current/Voltage Sensor Setup

PWM output headers

Just like the CC & CC3D, the Revo has a bank of 6 PWM output headers. If more PWM outputs are needed - the Flexi-IO port can be configured to support up to an additional 6 PWM channels if so required. PWM port 5 can also be configured to communicate with an external analog airspeed sensor or a governor for helicopters.

SWD Port

Serial wire debug port. This allows the use of cheap boards like the STM F4 Discovery as an in-circuit debugger to ease the firmware development.



Pinout

Color	JST-SH Pin	Pin Description
Black	1	GND
Red	2	NRS
Blue	3	IO
Orange	4	CLK

Micro USB

We have decided to move away from using the mini USB port found on the existing CC and CC3D and instead use a Micro USB port. The Micro USB port has several benefits: it's physically smaller, more robust, a much more widely adopted standard, and is the same type of port found on the majority of mobile phones. In all likelihood, you'll already have one of these cables at home.

The USB port provides a USB composite device with the following functions:

- OpenPilot HID device (default GCS interface, uses PC system drivers)
- CDC virtual serial port (telemetry, debugging, serial bridge mode relaying data from/to physical serial port to the virtual one for GPS/Bluetooth module setup, etc)
- 8-channel HID joystick (passes data from all supported R/C inputs to PC flight simulators)

And all these at the same time.

OPLink Modem

The Revolution has its own OPLink Mini built right onto the board! This is not only a 10DoF flight controller with an ST32F4 processor in the same small footprint as the CC3D, but also has its own LRS modem. The modem is directly powered from the Revolution itself, so you don't need to worry about any additional power supply.

Of course, the on-board modem will have the same functionality as the OPLink Mini. Both are fully configurable from the GCS.

Operating on the 433MHz band (a 900Mhz version will be released at a later date), the modem provides a direct telemetry link between the GCS and your flight controller. And just as with the OPLink Mini, you can adjust the output RF power for compliance with any governmental RF regulations, or it can be disabled entirely.

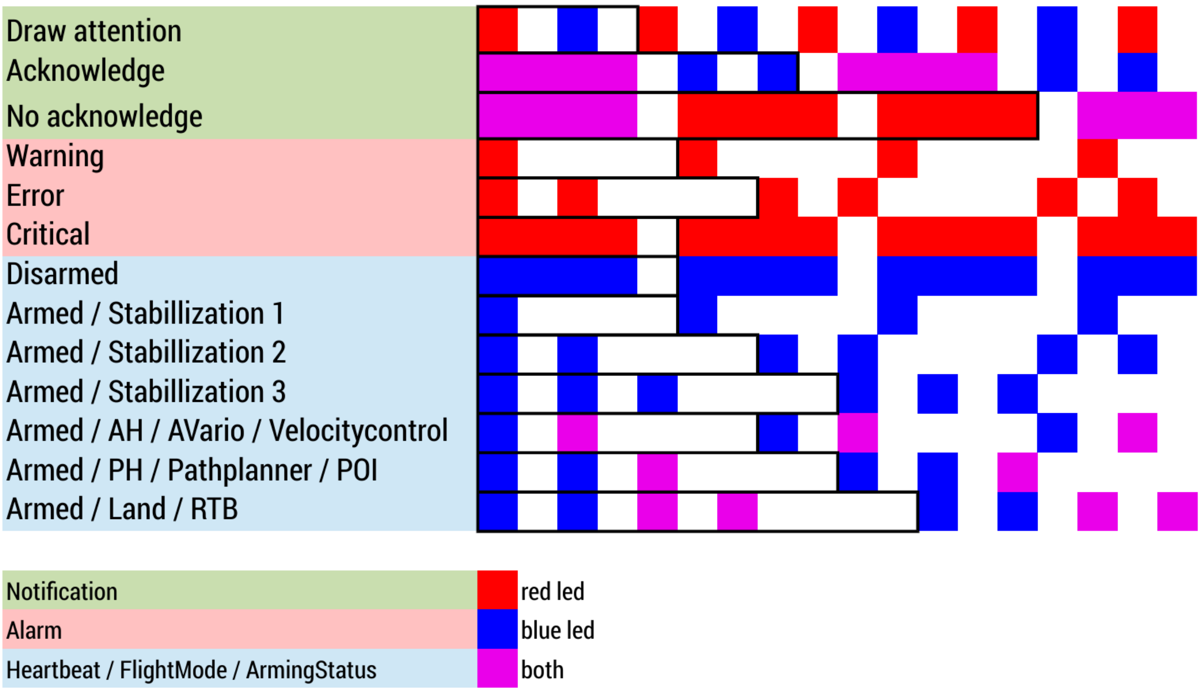
DIY Board

Schematics, PCB Layout, Gerbers, BOM: `Revolution.zip`

1.4.2 Revo - LED Indications

They will show in sequence:

- Notifications
- Alarms
- Heartbeat/FlightStatus/ArmingMode.



Todo

Video Example of Revo LED indications

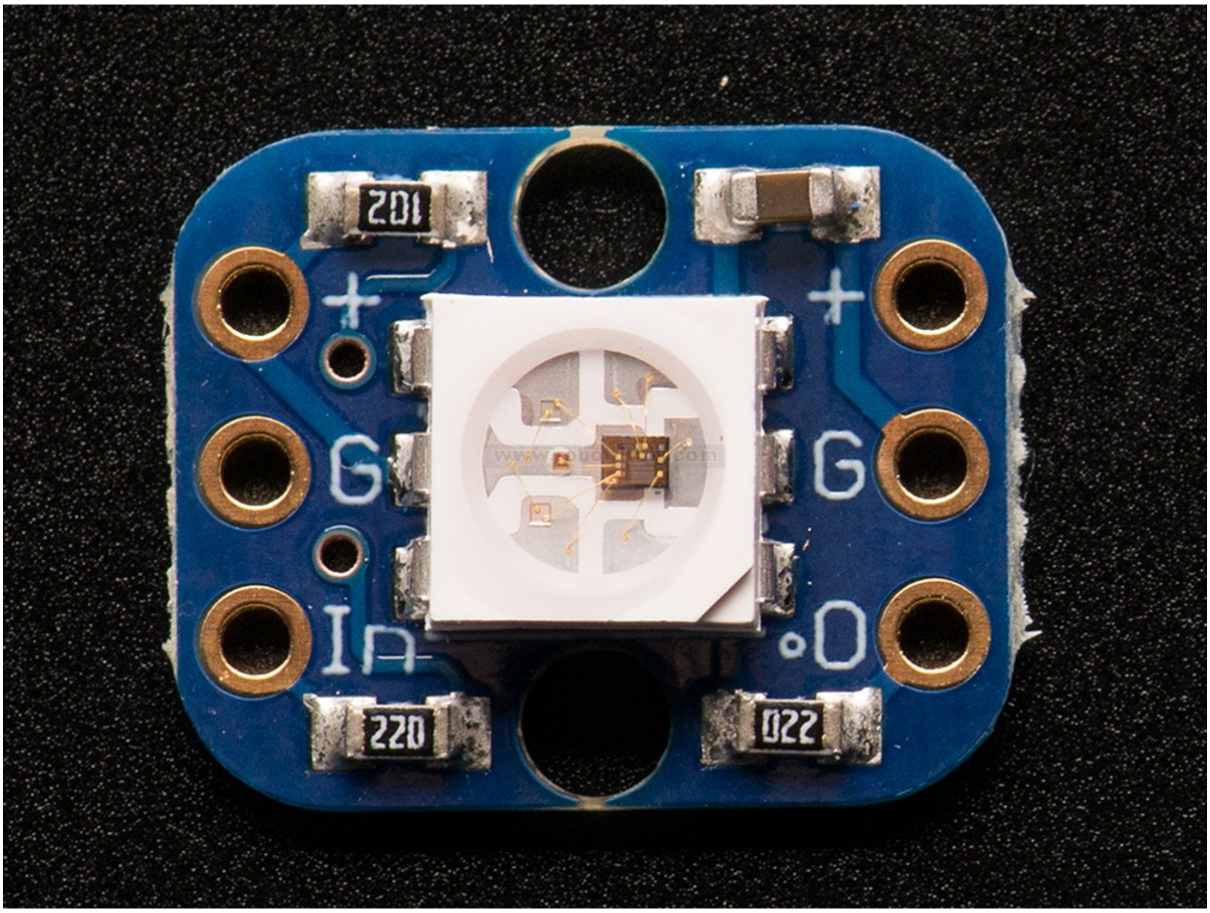
1.4.3 Revo - External LED setup

Overview

This document will describe the necessary hardware and software setup to connect an external notification LED to Revolution. WS2811, WS2812 and WS2812B multicolor LEDs are supported in both single LED or multiple LED configurations. The LED is useful for debugging and it provides visual information about flight modes and warnings during flight. The battery low voltage warning is particularly useful.

https://www.youtube.com/watch?v=S7CISYWC7MA

Hardware Connections



The OpenPilot firmware supports LEDs wired in both parallel and series configurations. A single LED is capable of displaying all of the data. Both the LED strips and the modules have voltage input (5V), ground (GND) and signal input (DI) pins, and also often corresponding output pins. For a single LED configuration, the output pins are not used; and in a multiple LED configuration, you can wire the LEDs together serially in a chain.

On the left is an example of a very common single LED module which is readily available on eBay. If you use a single LED, make sure that the breakout board has a capacitor on it; otherwise, inrush current spikes can damage the LED. The capacitor is usually a surface mounted component just like the brown one next to the LED unit in the photo. A capacitor is optional on configurations with two or more LEDs.

Revolution can command the LED(s) from various outputs. For the output signal, servo output rail pins 1-6 are supported, and Flexi-IO port pins 3 and 4 (from the left) are supported. Power for the LED is available from Flexi-IO pins 1 (GND) and 2 (5V), or from the servo output rail. After making the necessary connections, it is recommended that you mount the LED so that it is visible from below while flying.

The diagram below show the various configurations.

Todo

FIXME diagram

Software Setup

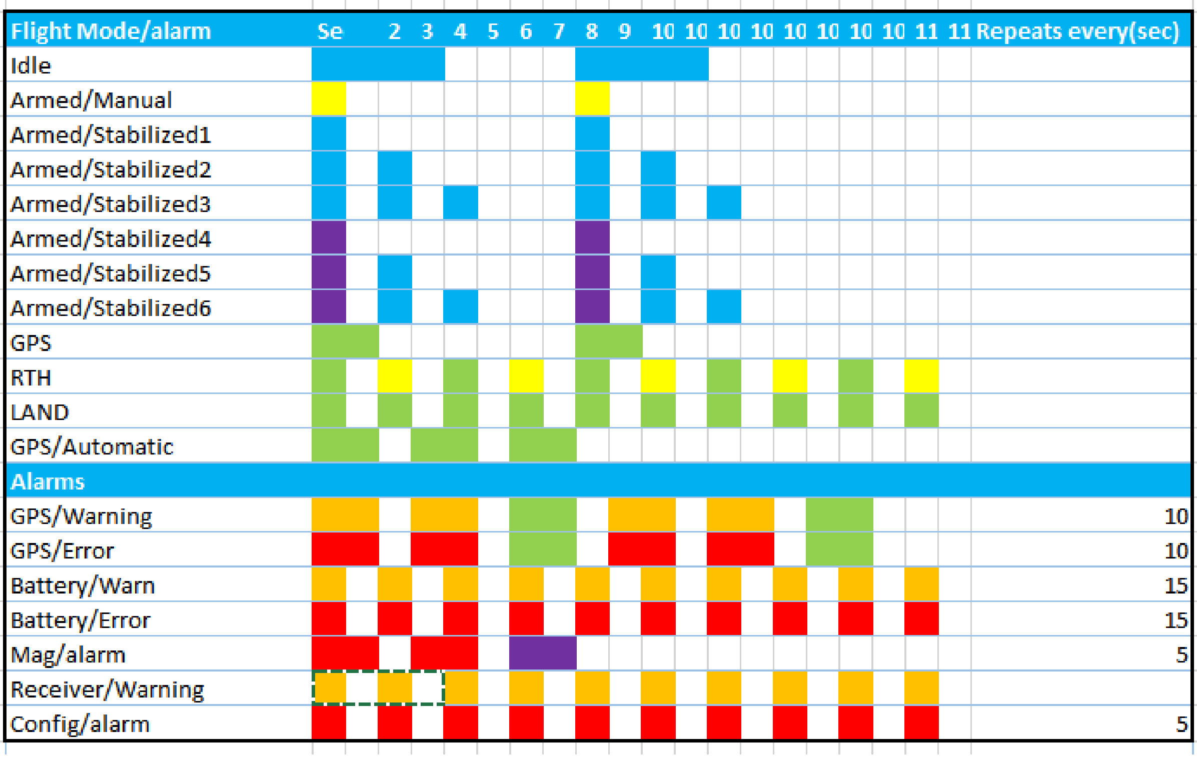
Following hardware setup, configure the controller in GCS as follows:

- Go to **System tab > HwSettings > WS2811LED_Out** and choose the pin where you connected the LED(s)
- Click the red **Save button** in the top part of the view

- Disconnect Revolution from PC
- Connect your flight battery. LED(s) should start showing system status according to the graphic in the Light Codes section below.

Light Codes

The following graphic shows different notifications Revolution will display with the LED(s).



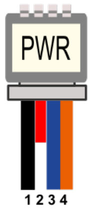
Flight Mode/alarm	Se		2	3	4	5	6	7	8	9	10	10	10	10	10	10	10	10	11	11	Repeats every(sec)
Idle	🟦	🟦	🟦	🟦					🟦	🟦	🟦	🟦									
Armed/Manual	🟨								🟨												
Armed/Stabilized1	🟦								🟦												
Armed/Stabilized2	🟦		🟦						🟦		🟦										
Armed/Stabilized3	🟦		🟦		🟦				🟦		🟦		🟦								
Armed/Stabilized4	🟪								🟪												
Armed/Stabilized5	🟪		🟦						🟪		🟦										
Armed/Stabilized6	🟪		🟦		🟦				🟪		🟦		🟦								
GPS	🟩	🟩							🟩	🟩											
RTH	🟩		🟨		🟩		🟨		🟩		🟨		🟩		🟨		🟩		🟨		
LAND	🟩		🟩		🟩		🟩		🟩		🟩		🟩		🟩		🟩		🟩		
GPS/Automatic	🟩	🟩		🟩	🟩		🟩	🟩													
Alarms																					
GPS/Warning	🟧	🟧		🟧	🟧		🟩	🟩		🟧	🟧		🟧	🟧		🟩	🟩				10
GPS/Error	🟥	🟥		🟥	🟥		🟩	🟩		🟥	🟥		🟥	🟥		🟩	🟩				10
Battery/Warn	🟧		🟧		🟧		🟧		🟧		🟧		🟧		🟧		🟧		🟧		15
Battery/Error	🟥		🟥		🟥		🟥		🟥		🟥		🟥		🟥		🟥		🟥		15
Mag/alarm	🟥	🟥		🟥	🟥		🟪	🟪													5
Receiver/Warning	🟧		🟧		🟧		🟧		🟧		🟧		🟧		🟧		🟧		🟧		
Config/alarm	🟥		🟥		🟥		🟥		🟥		🟥		🟥		🟥		🟥		🟥		5

1.4.4 Revo - Current/Voltage Sensor Setup

Overview

This page describes how to setup a voltage/current sensor using a Revolution or Revolution Nano board.

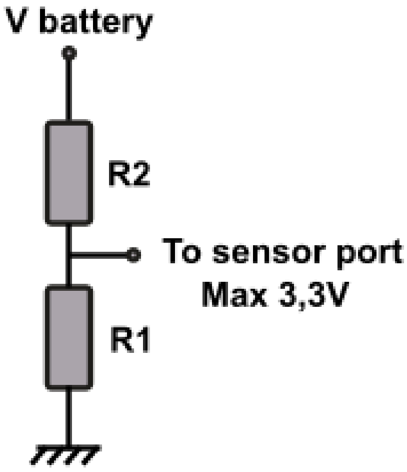
Hardware Connections

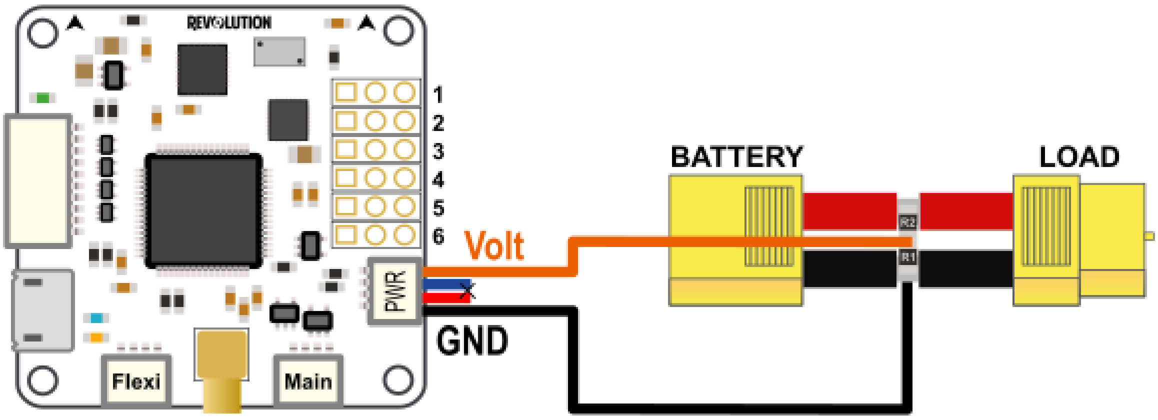


Color	Connector pin (board)	Description	AttoPilot pin (sensor)
Black	1	GND	GND
Red	2	Vcc	Not used
Blue	3	Current input	I
Orange	4	Voltage input	V

Basic voltage sensor

A basic voltage divider can be used, two resistors connected between **ground** and **plus** from battery.





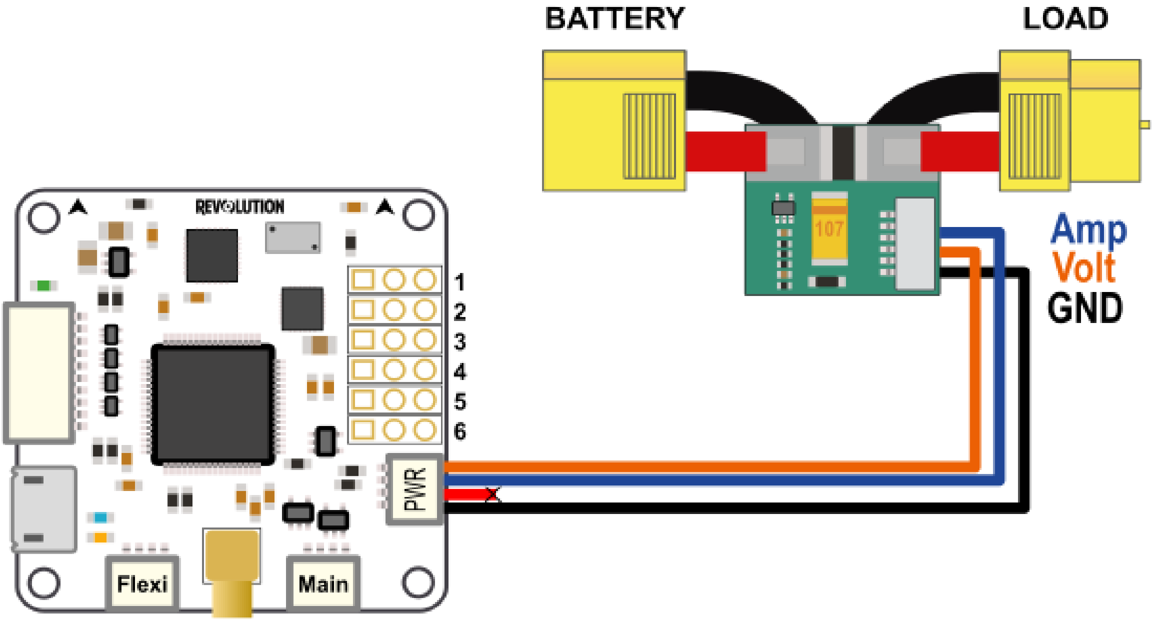
For a 4S battery (16.8Volts) the following values can be used:

- R1: 2,2 KOhms
- R2: 10 KOhms

With Vbattery = 16.8V, Vout = (16.8 * 2.2) / (10 + 2.2) = 3,03V

See also http://en.wikipedia.org/wiki/Voltage_divider

Current / Voltage sensor



Here is a list of common Attopilot sensors that can be used:

Module Type	Sensitivity	VoltageFactor	CurrentFactor	VoltageZero	CurrentZero
Attopilot 13.6V/45A	242.3mV/V, 73.20mV/A	4.127	13.661	0	0
Attopilot 50V/90A [1]	63.69mV/V, 36.60mV/A	15.701	27.322	0	0
Attopilot 50V/180A	63.69mV/V, 18.30mV/A	15.701	54.645	0	0

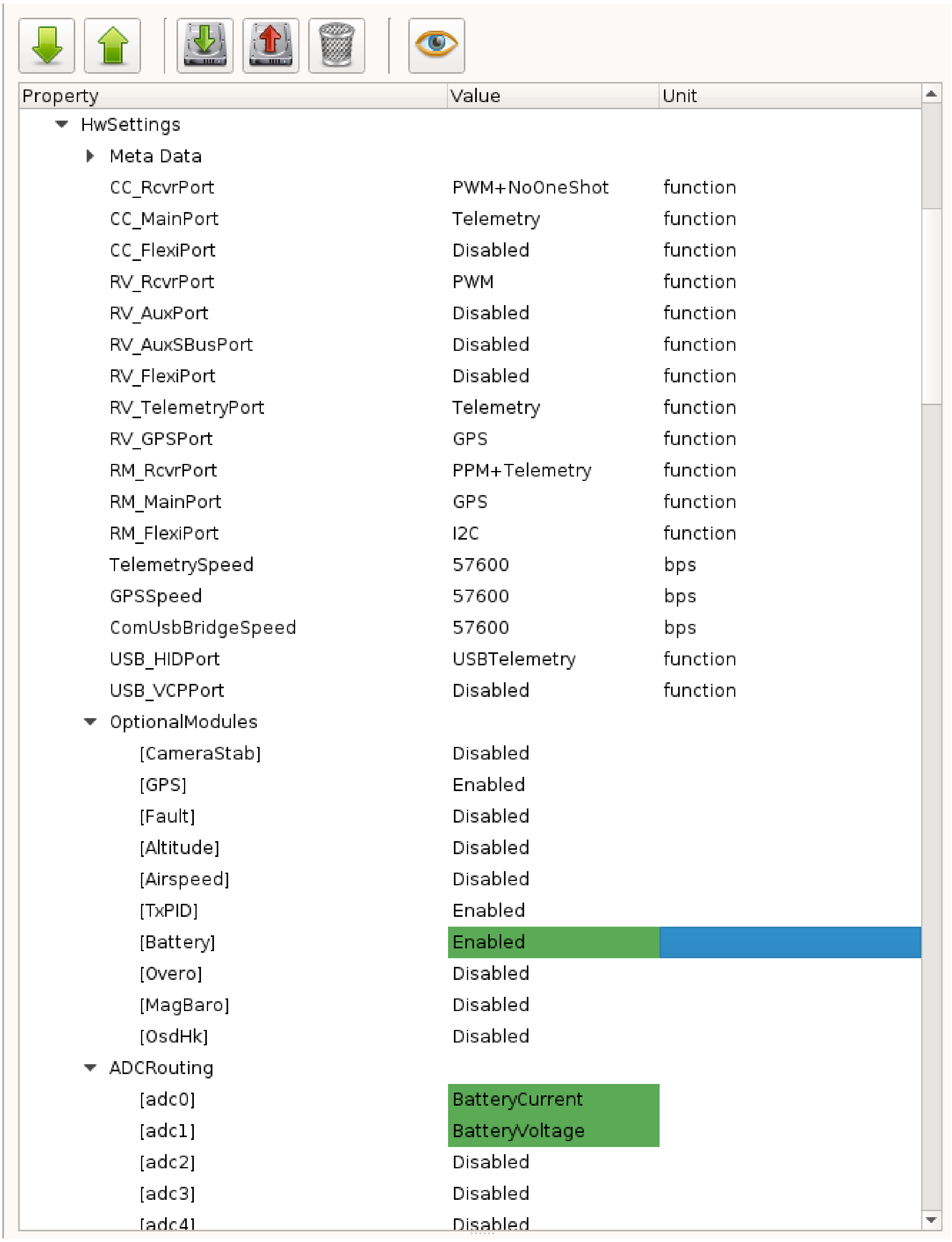
For custom sensor see the *Calculate SensorCalibration values* section.

Hardware settings

The first thing to do is to configure the hardware settings:

1. Connect the board to your computer and got to the **System tab** in your GCS
2. Browse the first part (Settings) and find the **HwSettings** UAVObject
3. Enable the Battery module: **OptionalModules > Battery > Enable**
4. Set the analog input pins for voltage and current:
   - **ADCRouting > adc0 > BatteryCurrent**
   - **ADCRouting > adc1 > BatteryVoltage**
5. Save changes, click **Upload** button
6. Reboot your board: disconnect **all power sources** and reconnect to your computer.

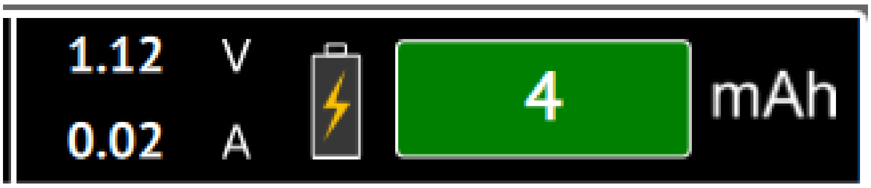
[1] Same as RCTIMER 90A current sensor



Property	Value	Unit
HwSettings		
Meta Data		
CC_RcvrPort	PWM+NoOneShot	function
CC_MainPort	Telemetry	function
CC_FlexiPort	Disabled	function
RV_RcvrPort	PWM	function
RV_AuxPort	Disabled	function
RV_AuxSBusPort	Disabled	function
RV_FlexiPort	Disabled	function
RV_TelemetryPort	Telemetry	function
RV_GPSPort	GPS	function
RM_RcvrPort	PPM+Telemetry	function
RM_MainPort	GPS	function
RM_FlexiPort	I2C	function
TelemetrySpeed	57600	bps
GPSSpeed	57600	bps
ComUsbBridgeSpeed	57600	bps
USB_HIDPort	USBTelemetry	function
USB_VCPPort	Disabled	function
OptionalModules		
[CameraStab]	Disabled	
[GPS]	Enabled	
[Fault]	Disabled	
[Altitude]	Disabled	
[Airspeed]	Disabled	
[TxPID]	Enabled	
[Battery]	Enabled	
[Overo]	Disabled	
[MagBaro]	Disabled	
[OsdHk]	Disabled	
ADCRouting		
[adc0]	BatteryCurrent	
[adc1]	BatteryVoltage	
[adc2]	Disabled	
[adc3]	Disabled	
[adc4]	Disabled	

Calibrate sensor

Now connect the battery to the battery sensor, if changes are done and battery module enabled you can see the display on PFD:



But values displayed are not accurate, we need to calibrate sensor!

Calculate SensorCalibration values

To configure the battery monitor, you need to calculate the **SensorCalibration** values.

For example, the documentation for Attopilot current sensor specifies the following information for each sensor type:

Sensor type	Voltage sensitivity	Current sensitivity
50V/180A	63.69mV/Volt	18.30mV / Amp
50V/90A	63.69mV/Volt	36.60mV / Amp
13.6V/45A	242.3mV/Volt	73.20mV / Amp

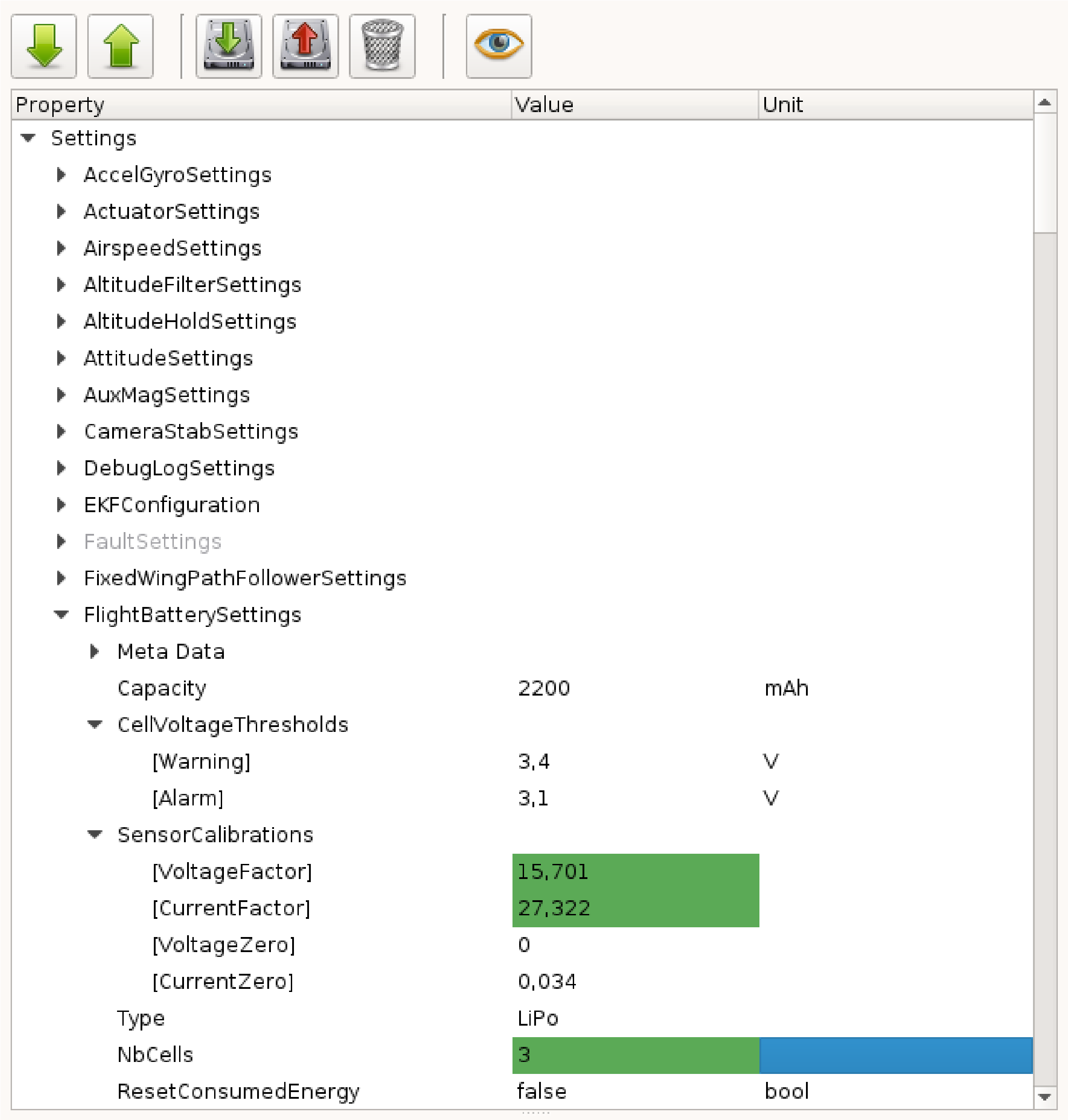
In order to use 50V/90A you'll have to do the following:

- Convert everything to Volt, so Volt/Volt and Volt/Amp:
  - 63.69mV/V / 1000 = 0.06369 V/V
  - 36.60mV/A / 1000 = 0.0366 V/A
- Then calculate the values to insert in **Voltage** and **Current Factors**:
  - **VoltageFactor** = 1 / 0.06369 = 15.701
  - **CurrentFactor** = 1 / 0.0366 = 27.322

Those factors are start values that can be adjusted later from readings, generally sensor's sensitivity are not so accurate.

Setting FlightBatterySettings Properties

1. Connect the board to your computer and go to the **System tab** in your GCS
2. Browse the first part (Settings) and find the **FlightBatterySettings** UAVObject
3. Set the **VoltageFactor** and **CurrentFactor** as in the previous section.
4. Set the cell **Capacity** if you want a warning to be issued when there are less than 2 minutes of flight time available; otherwise, set to 0.
5. Set **NbCells** with the number of cells in series (2 for 2S, 3 for 3S, 3 for 3S2P)
6. Set the **Warning** and **Alarm** thresholds if you would like a warning or critical alarm to be issued when going under the values that you specify.
7. Save changes, click **Upload** button.



Property	Value	Unit
▼ Settings		
▶ AccelGyroSettings		
▶ ActuatorSettings		
▶ AirspeedSettings		
▶ AltitudeFilterSettings		
▶ AltitudeHoldSettings		
▶ AttitudeSettings		
▶ AuxMagSettings		
▶ CameraStabSettings		
▶ DebugLogSettings		
▶ EKFConfiguration		
▶ FaultSettings		
▶ FixedWingPathFollowerSettings		
▼ FlightBatterySettings		
▶ Meta Data		
Capacity	2200	mAh
▼ CellVoltageThresholds		
[Warning]	3,4	V
[Alarm]	3,1	V
▼ SensorCalibrations		
[VoltageFactor]	15,701	
[CurrentFactor]	27,322	
[VoltageZero]	0	
[CurrentZero]	0,034	
Type	LiPo	
NbCells	3	
ResetConsumedEnergy	false	bool

Note: Sometimes sensor are not accurate with small readings, especially for current. A **CurrentZero** setting can be set: without any load change this value to obtain a 0 ampers display.

1.4.5 Revo - Sensor Calibration

Introduction

Todo

screenshot

This panel will allow you to undertake a Manual Calibration of the Senors on the Revolution board.

There is six (6) different Calibrations that can be preformed.

Steps:

- Setting Home Location
- Thermal Calibration
- Accelerometer Bias Calibration
- Magnetometer Calibration
- Board Level Calibration
- Gyro Bias Calibration

How to Video

by Corvus Corax

https://www.youtube.com/watch?v=dn9IDw2D1qw

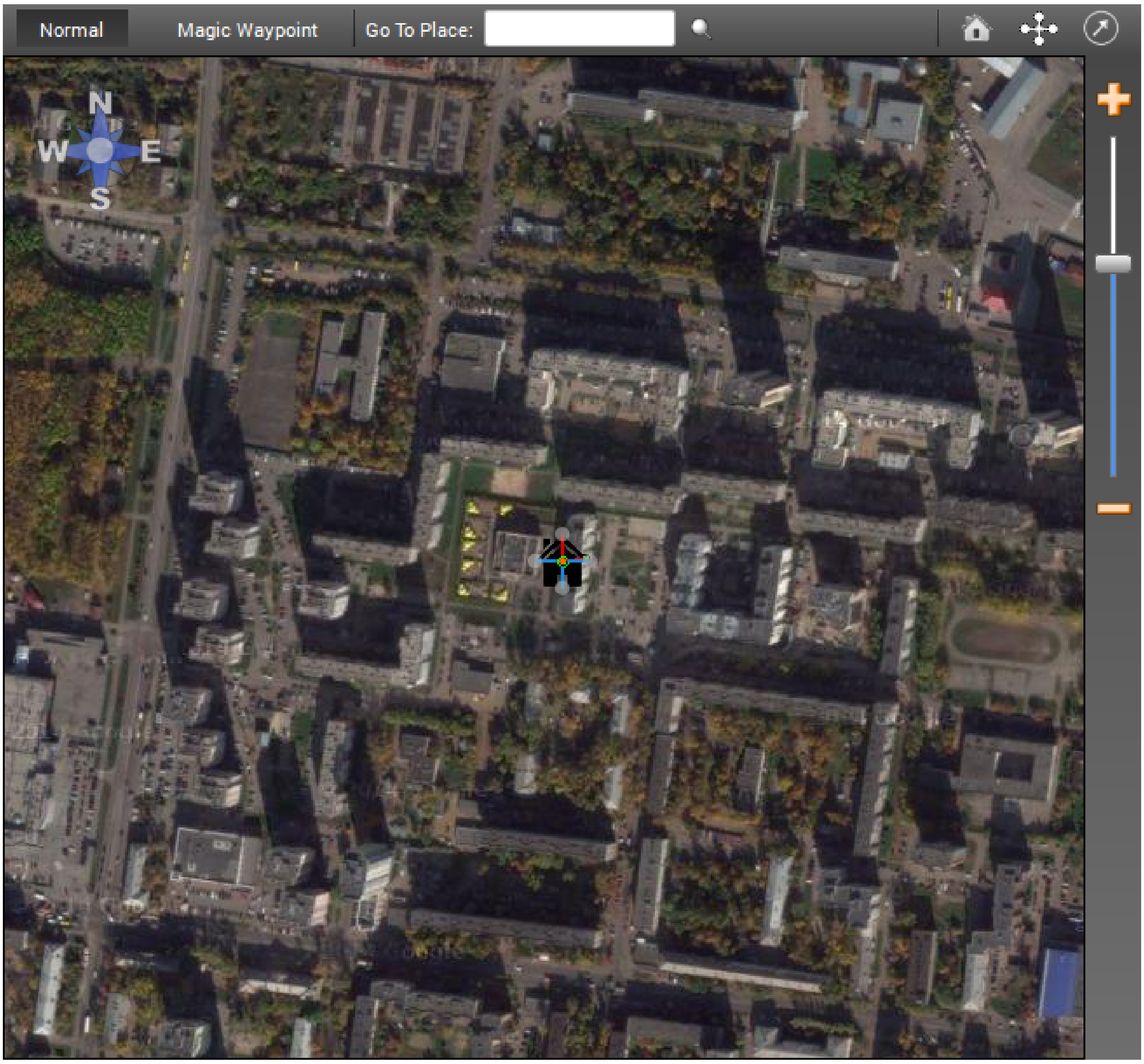
Setting Home Location

Warning: You must set your home location before performing the Sensor Calibrations

There is two ways of setting your home location:

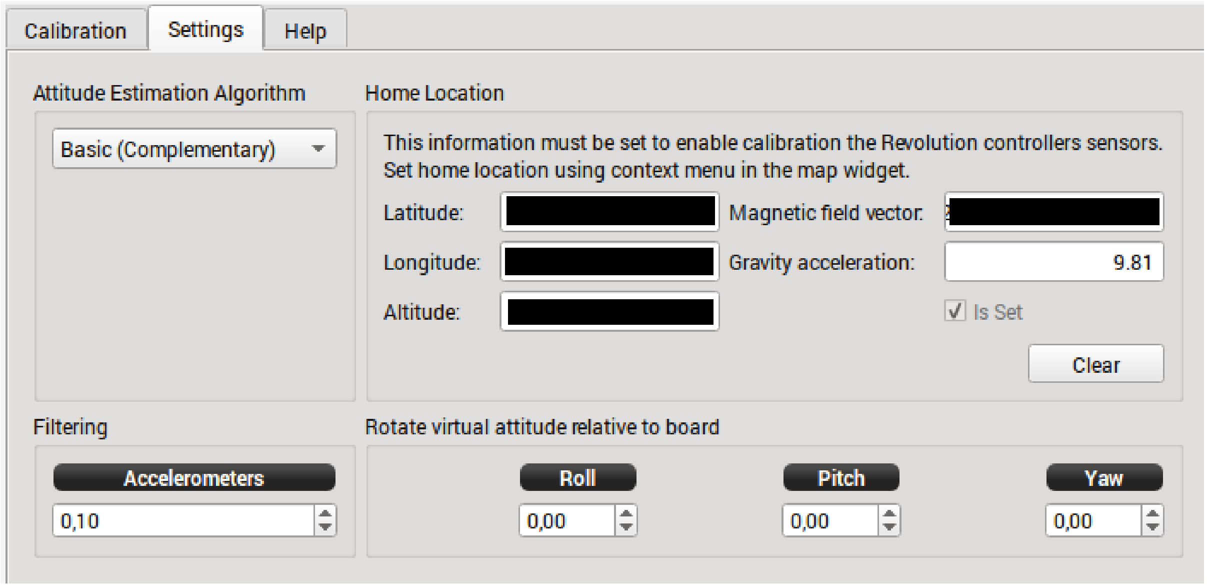
1. Using GPS Module
2. Manually Select

- Go to **Flight Data Tab**
- Either wait for GPS to goto your home location (using GPS Module) or manually zoom map up to your location (Manual Select)
- Once found, right click on the map and click on **Set the Home Location**



- Enter the Altitude of your home location in metres, click **OK**

Note: You can check the home location coordinates by going the the **Configuration Tab** - **Attitude/Settings tab**



Attitude Estimation Algorithm

Basic (Complementary)

Home Location

This information must be set to enable calibration the Revolution controllers sensors. Set home location using context menu in the map widget.

Latitude: [redacted]

Longitude: [redacted]

Altitude: [redacted]

Magnetic field vector: [redacted]

Gravity acceleration: 9.81

☑ Is Set

Clear

Filtering

Accelerometers: 0,10

Rotate virtual attitude relative to board

Roll: 0,00

Pitch: 0,00

Yaw: 0,00

Thermal Calibration

Note: The procedure below outlines the way to thermally calibrate the Revo to meet large temperature changes, it is not essential to put in the freezer as the min and max temp ranges of the calibration only need to be below and above the ambit temps of your location. (approx +- 10/15 deg C above and below)

However if the temperature rises or falls above or below the min/max temps you calibrated at, you will need to redo this calibration, so it is best to use some form of cooling to lower the min start temp and raise above max ambit temps of your location.

Also note this calibration is not needed, but will improve the performance of your board.

1. Plug your USB cable into your Revo. Start the OP GCS and ensure that you have disabled the OPLink modem (i.e. **Max Power** = 0). Then **Save** and **Disconnect**. Remove USB cable from computer but leave Revo plugged in.
2. Put your Revo in a plastic bag and seal it reasonably well, but with the USB cable exiting the baggie so it can be plugged in the computer.

Note: The bag is because we are going to freeze Revo and it will get condensation on it if we don't do this.

For added temperature range (higher temperatures) you might build a very simple hot box out of a low wattage incandescent light and a shoe box. You also might consider just wrapping Revo up in a good insulator, like a warm lightweight modern jacket. You could also set it out in the hot sun in the summer if you can plug it into USB at the same time.

3. Put Revo/baggie/cable in the freezer for 20 minutes or so.
4. Take it out, put it in the hot box or wrapped up or in the sun, or not... Immediately start the thermal calibration.
   - Don't move it at all while calibrating is best (gyros and baro are being calibrated).
   - Don't allow your home heater or air conditioner fan to run while doing this is best (air pressure changes).
   - Don't open or close any doors in the house is best (air pressure changes).
   - Don't do this on a windy day (chimney, etc.).
   - Don't do this on a stormy day (rapid changes in barometer).
   - Don't play 1812 Overture or let your sister dance in the next room is best (:)).

Note: All these "is bests" aren't critical though. I understand that the large number of samples taken masks a few bad samples.

5. Watch the thermal calibration and unplug the light bulb when it gets within a few degrees of your desired high temperature (i.e. 10-15 deg above what hottest day in your location).
6. If it finishes thermal calibration before reaching your desired temperature, you can start over or use what you have.
7. Save your thermal calibration with the **Save** button in the lower right.

Note: You can check the temperature min-max (temp_calibrated_extent). Go to the **System Tab** then **Settings/AccelGyroSettings/temp_calibrated_extent**. You can then see the min & max calibrated temps (below is screen shot of uncalibrated board showing 0 for both min & max, once calibrated you will see your boards min & max temps)

Todo

screenshot

Accelerometer Calibration

> **Warning:** During this calibration the board MUST remain perfectly still during each position calibration. Even vibrations on a table or someone walking past could upset the calibration.

1. On the **Calibration Tab**, click start "**Accelerometer calibration**".
2. Place the board as shown in the picture on screen and click **Save Position**.

...

10. If the calibration failed, you will see above, you will need to restart the accelerometer calibration over.

1.5 Multirotor Hardware Setup

The two videos below demonstrate how to set up a basic quadcopter as your vehicle.

The videos are now a little outdated but remain a valuable reference. The GCS has had numerous upgrades since they were prepared.

Video demo of Dave setting up a quad

Video demo of Dacat setting up his quad

1.6 Multirotor Vehicle Setup Wizard

The Vehicle Setup Wizard allows users to step through and setup all the initial settings including the TX.

When the GCS is started, the Vehicle Setup Wizard can be accessed by clicking on the "green" Vehicle Setup Wizard button.

Ensure board is connected through USB port before starting Wizard.

Depending on the flight controller board, some screens may or may not be displayed as you progress through the wizard.

1.7 First Flight

1.7.1 Pre-flight checks

There are a few things that you'll want to check before flying. Especially in the case of a multi-rotor. If you have swapped your motor outputs or your servo output is reversed, then the multi-rotor may flip or spin directly upon lift off.

> **Warning: REMOVE PROPELLERS!**
> You should always remove your propellers prior to starting tests or connecting your battery for the first time.

Arming (Turn your board on)

At start-up the system is disarmed. When disarmed, the motor(s) will not function, but the servos should work (*If installed, e.g. on Tri-Copter*). Before flying, the system needs to be armed with a specific stick input. The inputthat is needed for arming (and disarming) is configurable. For safety reasons, the default configuration is such that the system will not arm under any condition. Hence, it's mandatory that you configure the arming in the GCS input configuration.

Check your system alarms

The flight controller will not arm itself if any system status alarms are present. They can be checked by connecting to the flight controller board with OpenPilot Ground Control Station, and looking at system status indication under the primary flight display. All status objects should be green. PATH can remain yellow. For mode information on alarms, read the next page.

Check your stick inputs

With props removed, after you have armed your board, the blue LED should start flashing rapidly indicating that the board is armed. If you have specified that the "motors should spin at neutral output" in the output configuration, the motors should start spinning right now. Otherwise you can now apply some throttle and get the motors spinning.

Verify that the correct motors follow your stick input. E.g. rolling left should increase the throttle on your right motor(s) and vice versa. If the wrong motors are spinning faster, you need to recheck your configuration.

Rate mode

Note that some motors could start spinning faster if you have already selected a stability mode on your transmitter with the Flight Mode switch. You should do your first tests in rate mode only.

Check your stabilization

While the motors are spinning, you can move the aircraft in the pitch or roll direction. The flight controller should react promptly and accelerate the correct motors to counteract the movement. Please make sure that the correct motors are spinning up. If the opposite motors spin up, then you would flip your multi-rotor immediately upon lift off. In this case, you need to recheck your configuration.

Important: Check fail-safe!

Arm the aircraft and run your motors, now turn off the transmitter and confirm that all motors stop.

1.7.2 Your first flight (multicopters and helicopters)

You've done your pre-flight checks, your transmitter and flight batteries are all fully charged, and you are confident that the vehicle has been configured correctly - it's time to fly!

But, it's also time to take a deep breath and honestly evaluate your flying skills - if you are a novice then you should consider buying a flight simulator and developing some basic skills - crashes in a simulator cost nothing, so you will rapidly justify the investment for this software.

If you are feeling confident, take the model to a large, clear and deserted open space - you don't want any trees or overhead wires nearby, you definitely don't want any people or dogs in the area, and you shouldn't be too close to a road. A grass field is better for first flights, rather than concrete or tarmac. Ideally there will be no wind, but a very slight breeze shouldn't be a problem.

Important: You may be disconnecting and connecting power to the model several times during these first steps - you must wait at least 30 seconds before reconnecting power, otherwise the model may behave unpredictably.

Remember to keep the model still for about 10 seconds after each connection, while the gyros are calibrated.

The following steps have been written on the assumption that ‘Zero gyros while arming aircraft’ has been selected on the Attitude Configuration page of GCS (the recommended setting). This can help with the stabilization of the model.

Note: Remember to keep the model still for several seconds after arming, while the gyros are zeroed.

1. If you have brought anybody with you to witness the first flight, make sure that they know where to stand, how to behave, and what to do if the vehicle goes out of control. If there are any children, make sure that they are under close supervision.
2. Put the throttle stick to minimum and switch on the transmitter. If your transmitter has a throttle lock facility, set this to ‘locked’. Put the transmitter on the ground next to where you plan to stand while flying the vehicle.
3. Place the model on the ground about 10 paces away with the tail towards the transmitter. If there is a breeze, put the model either upwind or downwind of the transmitter - you don’t want a cross-wind.
4. Connect the power to the vehicle and **do not move it for about 10 seconds while the gyros are calibrated**. Return to your transmitter.
5. Take a good look around for safety’s sake, then unlock the throttle and arm your board - the blue LED on the board will now flash rapidly and the rotors will spin if you’ve selected ‘Motors spin at neutral output when armed’. **IMPORTANT: you must wait a few seconds after arming while the gyros are zeroed** (see note above). The aircraft may be unstable if this isn’t done. Use this time to have another good look around you.
6. Steadily increase the throttle until the vehicle is about to lift off the ground - any tendency to flip or spin will be apparent at this time. Close the throttle immediately if the vehicle does anything unexpected, and then take a look at the problem-finding guide.
7. If everything looks OK. then close the throttle, disarm the vehicle, take a deep breath and have yet another look round the field - it’s time for lift-off!
8. Arm the vehicle, wait for a few seconds, then open the throttle confidently until the aircraft lifts off the ground. Try to hover about 1 or 2 metres off the ground, while keeping in one position.

You are now flying! Obviously this bit is not as easy as it sounds and much practice is required.

The important thing to remember is to close the throttle if the aircraft goes out of control - you will crash at some time, and cutting the power will minimize the damage.

Get into the habit of disarming the board when landing for more than a few seconds or when approaching the vehicle to handle it, and don’t forget the short wait after re-arming - again, use this period to look around the field before flying.

Disarm the board and set the throttle lock when you have finished flying, then put down the transmitter a little way from the model. Disconnect the power from the aircraft, then turn off your transmitter.

Now that you have proven that the aircraft will fly, you might like to try the training exercise videos for helicopters found on this website page. Many of these are also suitable for multi-rotors.

Todo

FIXME: Trainig videos

1.7.3 Optimizing values

Apart from tuning the stabilization settings, there are some values which advanced users may want to change pretty soon. The default values will fly your aircraft perfectly fine, but would limit some users in their flying style.

Please find below a few settings which can easily be cranked up for more experienced users. Note: these are available in each of the three Settings Banks.

Full stick angle in Attitude mode

Location	Stabilization panel, Responsiveness, Attitude mode response (deg)
Standard value	55°
Tuned+ value	65° or more

Specifies how many degrees the vehicle will bank on a full stick deflection when in attitude mode.

If you fly your multi-rotor in heavy wind, you may find that low values are not sufficient to counteract the wind fast enough. The default value is fairly good for a beginner.

Full stick response in Rate mode

Location	Stabilization panel, Responsiveness, Rate mode response (deg/s)
Standard value	220°/s
Tuned+ value	360°/s or more

Specifies how many degrees per second a full stick deflection commands in all modes **except Attitude mode**.

If you want to do flips with your multi-rotor you should increase this setting. Flips that take too long to complete can result in too much loss of altitude for the beginner. To get some idea of how fast you want to flip, imagine the flip taking one second to complete - that would equate to 360°/s.

Note: This is the value to control rotation rate when using Rattitude mode.

Full stick response limit in any mode

Location	Stabilization panel, Responsiveness, Max rate limit (deg/s)
Standard value	300°/s
Tuned+ value	360°/s or more

Specifies the maximum rotation rate in degrees per second commanded by a full stick deflection on the associated axis **in any mode**.

Make sure this is the same or higher than Rate mode response above.

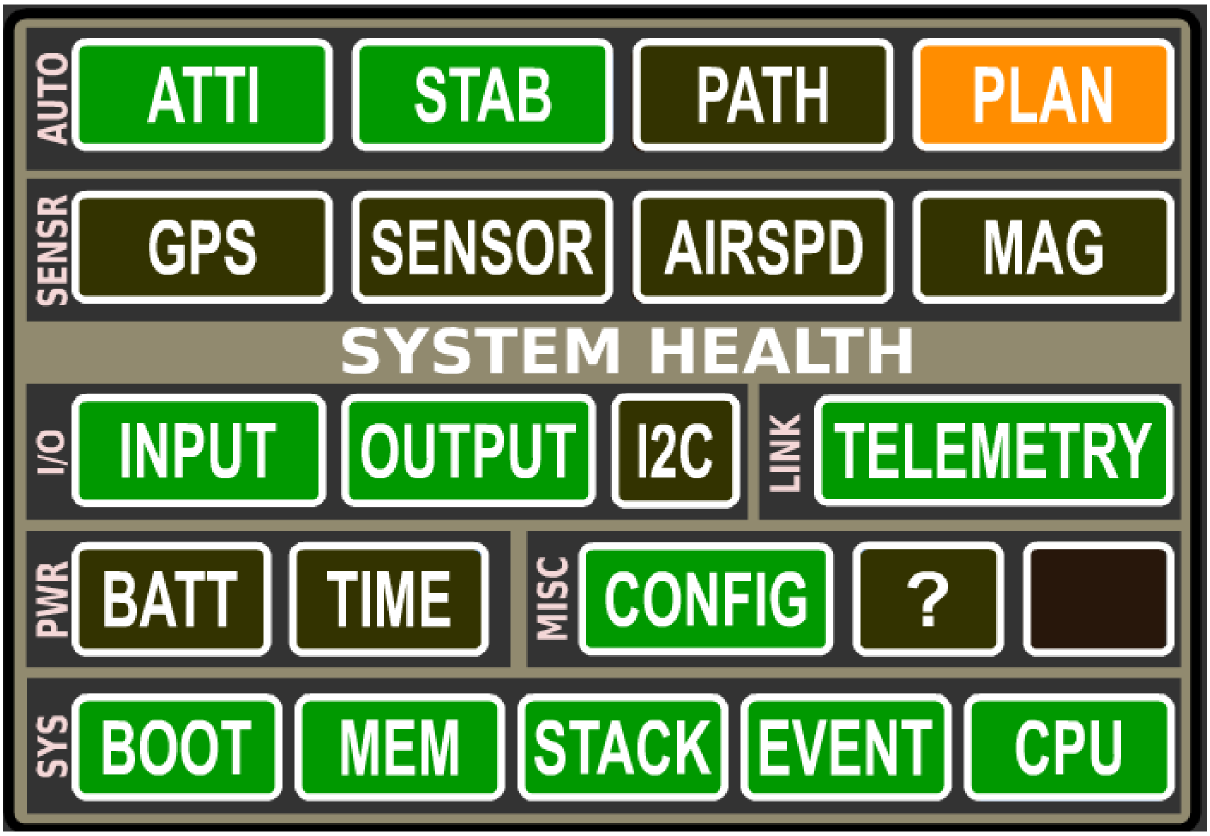
MaxAxisLock

Location	Stabilization panel, Expert Tab, Axis Lock
Standard value	5°
Tuned+ value	15°

The maximum number of degrees that the control accumulates error. The default setting is changed to 15° in newer firmware & should be a good value for multi-rotors.

1.8 Alarms - What they mean and how to fix them

1.8.1 Status widget overview



The OpenPilot/LibrePilot flight control firmware has a built-in status system that gives you an overview about what's happening with the board. The System Health Status widget in the Ground Control Station can be used to diagnose various problems, and to check which modules of the firmware are running.

Arming

The flight control board cannot be armed until crossed-out alarms and red alarms have been solved. All objects, except for the special "Plan", should be green for optimal flight performance. The information corresponds to the UAVObject **DataObjects > SystemAlarms > Alarm**.

1.8.2 Status object explanations

Auto (Autonomous)

Atti

Shows the status of the board's attitude data. If all is well with gyroscope and accelerometer, it turns green after gyroscope calibration upon power up, or if you are using "**GPS Navigation (INS13GPSOutdoor)**" stabilization mode, when the Inertial Navigation System's **Extended Kalman Filter (EKF)** has fired up. EKF is a sophisticated sensor fusion algorithm that takes data from relevant sensors and creates a best possible estimation of the board's angle, velocity and position.

- ATTI Attitude data not available, waiting for gyroscope calibration. Don't move the vehicle while gyros are being calibrated upon board power-up. There is no data coming in from the sensors, which usually indicates faulty onboard sensors on CC3D and unknown home location on Revo. For Revo this is normal when you have not had GPS fix yet on a new build. The home location will be automatically set when GPS gets enough satellites and good fix. The sensors can be damaged in a bad crash. Otherwise, contact board seller.

- ATTI Data is received from the sensors, but attitude information is not available. This usually happens when the EKF is not running yet. Make sure that GPS and MAG alarms are green, and that all calibrations have been done properly. It sometimes helps to move the vehicle around a bit to give EKF a better view of what's going on with sensors.
- ATTI EKF is running, but the state estimation is not optimal. Good calibration and moving the vehicle a bit helps this situation.
- ATTI The system has a clear view of the state of the vehicle.

"The Bug"

There is a very specific condition that can arise within the system that can sometimes occur after the EKF has been initialised, and the vehicle has been left stationary for an extended period of time.

Fortunately this condition is easily detected and the flight firmware mitigates the event. However, the solution is extremely complicated and beyond the capability of most people, and until such stage as the developers have a chance to address the matter, it's occurrence is indicated as follows:

- A large red X appears over the PFD.
- A Yellow ATTI alarm with STAB green

This event has been nicknamed "The Bug", and it is still possible to arm and fly the vehicle in this condition.

Should you wish to clear the indication, one can reset the Revolution flight controller, or alternatively, change the Attitude Estimation Algorithm to **Basic(Complementary)**, and then change back to **GPS Navigation(INS13)**. This action will cause the EKF to reinitialise and the indications will be cleared.

Stab

Shows whether the board is capable of stabilizing flight. This status goes very much hand in hand with Atti.

- STAB Waiting for gyroscope calibration. Don't move the vehicle while gyros are being calibrated upon board power-up.
- STAB The stabilization module cannot stabilize flight. See red Atti status for explanation.
- STAB Can be a brief alarm when one gyroscope update is missng.
- STAB Stabilization of flight can be performed.

Path

Shows whether the Revolution board is capable of autonomous path following. Autonomous flight requires a GPS and a stabilization algorithm set up as GPS navigation.

- PATH The flight controller has not been configured to do autonomous flying. Autonomous flight is only possible with stabilization mode "GPS Navigation".
- PATH The system has been configured to initialize Path Follower module, but it can't be used at the moment. It happens usually because EKF is not running, see Atti red explanation.
- PATH All is good with the Path Follower module. You can use GPS flight modes.

Plan

Shows the status of an autonomous flight plan that can be uploaded to Revolution using the Ground Control Station. The status of Path remains yellow until a proper plan has been uploaded, and turns green if all is good with the plan. A valid plan can be activated with a path follower flight mode.

- PLAN The flight controller has not been configured to do autonomous flying. Autonomous flight is only possible with stabilization mode "GPS Navigation".
- PLAN Path has been uploaded, but data is invalid and cannot be used for autonomous missions.
- PLAN No path plan has been uploaded, but the system is ready to receive a plan. This is okay if you don't intend to do autonomous missions right now.
- PLAN A valid and usable flight plan has been uploaded and stored on Revolution.

Sensor

GPS

Shows the status of the GPS that can be connected to an OpenPilot flight controller. GPS is required for autonomous missions and more sophisticated flight modes.

- GPS A GPS has not been configured to be used.
- GPS The GPS has been configured, but no valid data is coming in. This is normal if flight battery is not connected, because GPS only gets power from external sources, not USB. Check the baud rate and the used protocol of your GPS. Double check serial connection, TX/RX need to be crossed between board and GPS.
- GPS Serial communication is fine but the GPS has no valid fix. Wait for GPS to gather satellites, and preferably have your vehicle in an open area.
- GPS The GPS has a fix and navigation can be used. However, the position quality is very low (the indication <7 satellites and/or PDOP > 3.5m). A blue LED will flash on the OP v8 and v9 GPS.
- GPS The GPS has a valid 3D fix.

Initial GPS setup information

When powering up the GPS for the first time, it might take over 30 minutes for the GPS to download almanac information from the satellites and acquire a good lock. Be patient, and have a clear view of the sky.

Sensor

Shows the status of the sensor handler module.

- SENSOR Sensor module is not being used in current configuration.
- SENSOR Sensor communications are up and ok.

Airspd

Shows the status of an optional air speed sensor that can be used with fixed wing aircraft.

- AIRSPD Airspeed sensor has not been configured to be used.
- AIRSPD Valid data is coming in from the airspeed sensor.

Mag

Shows the status of Revolution's magnetometer, or the status of an auxiliary magnetometer on board the OpenPilot GPSv9 if the GPS is configured to be used.

- MAG Magnetometer is not being used in current configuration, or auxiliary magnetometer is not feeding data. If using GPSv9, power up the board externally. *Home location has to be set* to enable magnetometer.
- MAG Data is coming from the magnetometer, but the readings are off by over 15%. This can be caused by various reasons:
  - Magnetometer has not been calibrated with current vehicle (or after recent modifications to vehicle)
  - There are high currents in wires, interfering with the magnetometer. Twist wires and route them away from magnetometer.
  - Calibration was properly done outside, away from metallic objects, but the vehicle is now inside in a different magnetic environment. This behavior is normal.
- MAG Magnetometer readings are off by over 5%.
- MAG Magnetometer is working properly and the quality of the measurements is good.

I/O (Input / Output)

Input

Input module handles the data that is coming from your receiver.

- INPUT R/C input has not been configured. Use **Input tab** or **Transmitter Setup Wizard** to configure your radio channel inputs.
- INPUT No R/C input data. Power up receiver with the flight battery.
- INPUT Valid R/C input data is coming in.

Output

Output module takes motor speed and servo position data from stabilization algorithms, and feeds it into output channels.

- OUTPUT Channel outputs have not been configured. Use **Vehicle Setup Wizard** to configure them automatically.
- OUTPUT Outputs are configured and can be updated.

I2C

I2C is a bus that connects onboard or auxiliary sensors and handles the data transmissions. I2C is designed for communications internal to a PCB, and does not work well via wire connections. It is okay to use for LED controls and similar functions, but is absolutely not recommended for flight-critical sensor connections.

- I2C I2C module is not being used.
- I2C I2C module is in error state.
- I2C I2C communications are up and working properly.

Link

Telemetry

Shows the status of Telemetry communications module

- TELEMETRY Telemetry module has encountered an error. Set up only one telemetry output port.
- TELEMETRY Telemetry data communications are working properly.

Pwr (Power)

Batt

Battery status shows whether you have enough voltage in the battery to fly. Set limits for this in FlightBatt (CHECK this) settings. It requires a battery voltage sensor to work. Battery monitoring module can be enabled in system settings' optional modules.

- BATT Battery monitoring module is not enabled.
- BATT Not enough battery voltage to safely take off. Default limit is 3.1V/Cell.
- BATT Battery voltage is low, but flying is possible. Default limit is 3.4V/Cell.
- BATT Battery voltage is ok.

Time

Shows whether you have enough energy in the battery left for flying, and requires a battery voltage and current sensor to work. Currently has a bug when not using a current sensor; set the battery capacity to 0. This disables the estimated flight time counter and associated alarms.

- TIME Battery monitoring module is not enabled, see above Batt explanation.
- TIME Battery energy is low, flying cannot be performed safely.
- TIME Low amount of energy in the battery, flying is still possible.
- TIME Good amount of energy left in the battery to fly.

Misc

Config

Shows whether your flight controller board has been properly set up.

- CONFIG Board configuration problem. If you have set up GPS modes (GPS Assist, PosHold, RTB) to one flight mode, then make sure that "GPS Navigation (INS13)" fusion algorithm is selected.

  CC3D can't use GPS Navigation (INS13) and do not support GPS Assisted modes.

  Select **GPS Navigation (INS13)** in **Config > Attitude tab > Parameters > Attitude Estimation Algorithm**.
- CONFIG Board configuration ok.

Sys (System)

Boot

Shows that a board reboot is required, or fail-safe settings have been loaded upon boot.

- BOOT Boot alarm can be caused by various reasons:
  - No valid telemetry option selected, so board will boot with default USB telemetry
  - Board init failed due to driver, module or RAM issues, and the board has been booted up in fail-safe state
  - Board has been put to safe mode by the user
  - Board needs a reboot after hardware configuration changes
- BOOT Flight controller booted up properly.

Mem

Displays the status of remaining memory (RAM) that are used by processes internal to the flight controller.

- MEM Very low RAM left, flying cannot be done safely. Less than 40 bytes for CC/CC3D or 500 bytes for other boards.
- MEM Low amount of RAM left, flying can be done but don't enable more software modules. This is common with older flight controllers such as CopterControl. Less than 200 bytes for CC/CC3D or 1000 bytes for other boards.
- MEM Sufficient amount of RAM left for system to operate and expand.

Stack

Shows the status of the microcontroller's stack, which is a place where low-level functions store data.

- STACK Stack status ok.

Event

Shows the status of event system. A very heavy load can cause the event system to be overloaded.

- EVENT Event system error or overloaded. This can be caused by a bug or too high telemetry update rates when OPLink has low baud, for example.
- EVENT Event system at high stress. See above.
- EVENT Event system ok.

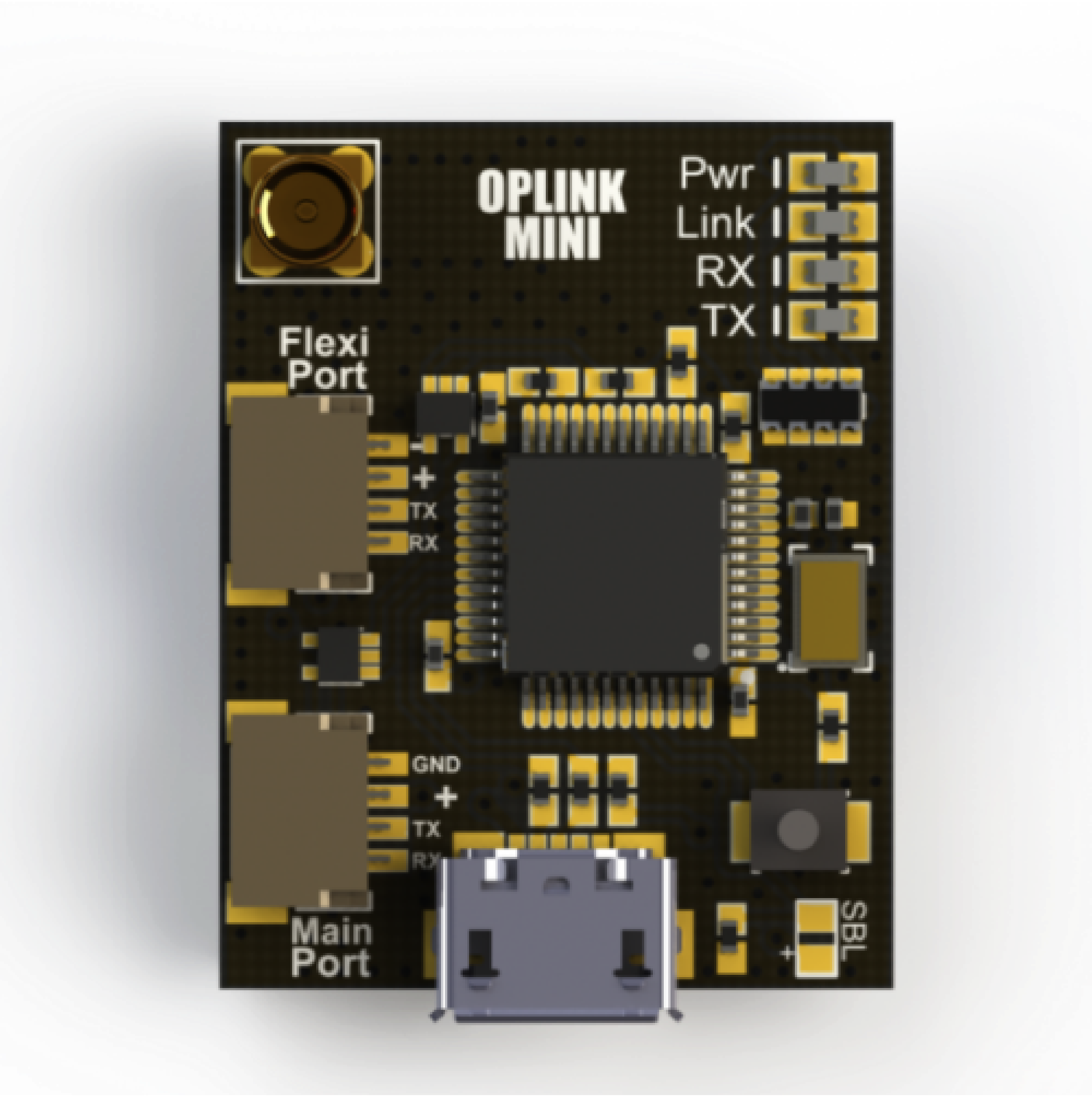
CPU

Indicates CPU load.

- CPU CPU load is very high, flight cannot be performed safely.
- CPU CPU load is high, but flight can be performed. Don't enable more software modules like TPS or board rotation. Should only occurs for CC/CC3D.
- CPU CPU load is at an acceptable level, and flying is safe.

1.9 OPLink and Bluetooth Setup

1.9.1 Overview



The OPLink is a radio transceiver backed by an ARM32 powered digital packet processor specifically designed for the OpenPilot project, it was originally developed under the codename: PipXtreme and there are many artifacts that still reflect this. It is a two-way radio system allowing real time telemetry information for the Ground Control Station, Wireless Configuration and even radio control from your transmitter over a single communications link.

You will need two OPLink radios to establish a connection between the vehicle and ground station. Note that the Revolution board has an OPLink built into the flight controller. The same firmware version must be running in Revo and OPLM devices *(i.e. if you upgrade one board, you will need to upgrade all others that are 'bound' together)*.

The OPLM units operate at 433Mhz. This means that there will be a **frequency conflict** if you operate the vehicle using a UHF 433Mhz system. In basic terms, this conflict will lead to a loss of vehicle control and / or a crash. Use transmitters in other parts of the spectrum *(i.e. 2.4G, 27Mhz, 36Mhz)* for vehicle control if you intend to enable the OPLink on your Revo for transmission of telemetry data. If you wish to use a 433Mhz transmitter for vehicle control, then use an OPLM which will by default, enable the transmission of telemetry.

Check local laws and regulations regarding radio licensing, see our `license` article for more information.

The device that is intended to undertake the most of the transmissions should be allocated as the Co-ordinator. If you wish to use the OPLM for vehicle control, then the Co-ordinator should be the OPLM device in your transmitter module at the ground station. In this case, the Revo or another OPLink will be acting as a slave receiver but will also transmit telemetry information to your ground station which is comprised of another OPLM connected to a computer that is running the OP GCS Flight Data page. Alternatively, if you wish to simply receive telemetry data at the ground station, then the Revo should be configured as the Co-ordinator. Configuration instructions are provided below and linked-to pages.

First time use with OPLM

When you first try to use your OPLM with the GCS, it will not be automatically picked up by the GCS. You first need to go into the 'Firmware' tab and click 'Upgrade & Erase' while the OPLink is **disconnected**. Then plug in OPLink when the upgrade process asks for it. Once this has been done, your OPLM board will show up at the bottom of your GCS as a connected device. **The configuration page icon at the bottom of the page icon list may be hidden without scrolling down, depending on the size of your computer screen.**

OPLink Mini

- 100mW Standalone Radio Modem
- 3 IO Ports: Micro-B USB, Mainport & Flexiport
- MMCX Antenna Connector.
- Weight: 4g
- Dimensions: 20mm x 29mm
- Input Voltage: +5v

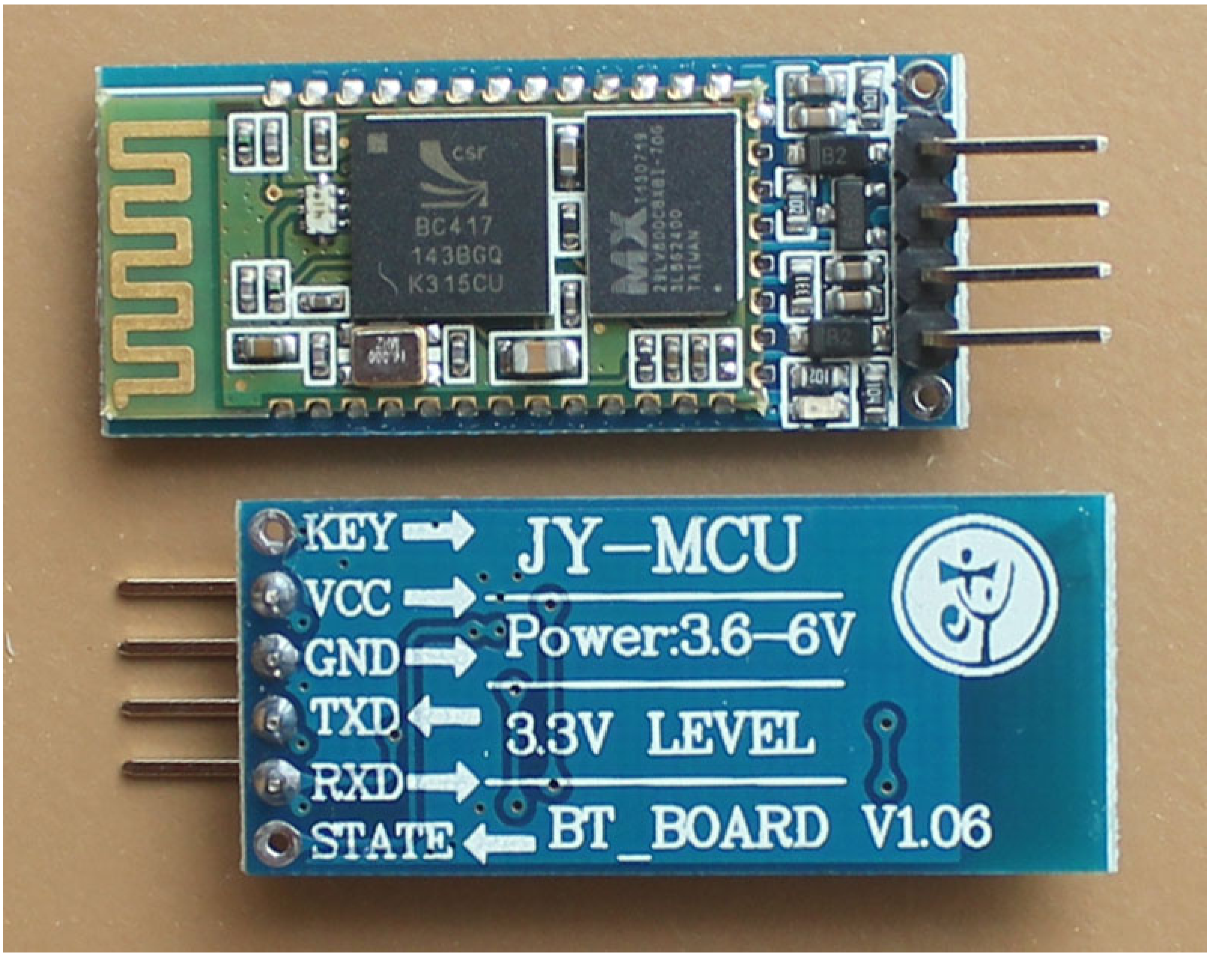
OPLink Revo

- 100mW integrated radio modem included the OpenPilot Revolution platform.
- MMCX Antenna Connector.
- Directly integrated, no external connections or power needed.
- Dipole whip antenna
- OPLM CC - CC3D - Atom Hardware Setup
- OPLM Binding and Basic Telemetry
- OPLM Vehicle Control Link

1.9.2 DIY Board

Schematics, PCB Layout, Gerbers, BOM: `OPLink Mini.zip`

1.9.3 Serial Bluetooth RF Transceiver Module RS232



This small size Bluetooth TTL transceiver module allows your target device to both send or receive the TTL data via Bluetooth technology without connecting a serial cable to your computer. It's easy to use and completely encapsulated. It can be used to establish a short telemetry link directly to the vehicle, or a telemetry link between a computer and a transmitter, that has an OPLink module connected to it.

- Chipset **CSR BC417143**
- Bluetooth version **V2.0+EDR (Enhanced Data Rate)**
- Output power **Class II (±2.5mW range ±10m)**
- Flash **8Mbit**
- Power Supply **3.3V (5VDC via breakout board)**
- Interface **UART**

This configuration guide covers the displayed serial Bluetooth module. It's available in many different internet shops (e.g. Ebay, Goodluckbuy, BlueSkyRC). The boards generally ship for around 10$ - 20$. Although the hardware is the same, several software or firmware revisions exist.

Different input voltage versions

Important: There are different versions available; 3VDC or 5VDC. You want the 5VDC version in order to connect directly to CopterControl. This module has an on-board voltage regulator.

The voltage input must be clearly mentioned. Typically, these boards can handle a power supply between 3.6VDC ~ 6.0VDC when an on-board voltage regulator is available.

Check for the input & output voltage which is mostly mentioned on the module diagram.

- Bluetooth Setup for Telemetry

1.9.4 OPLM CC - CC3D - Atom Hardware Setup

Summary

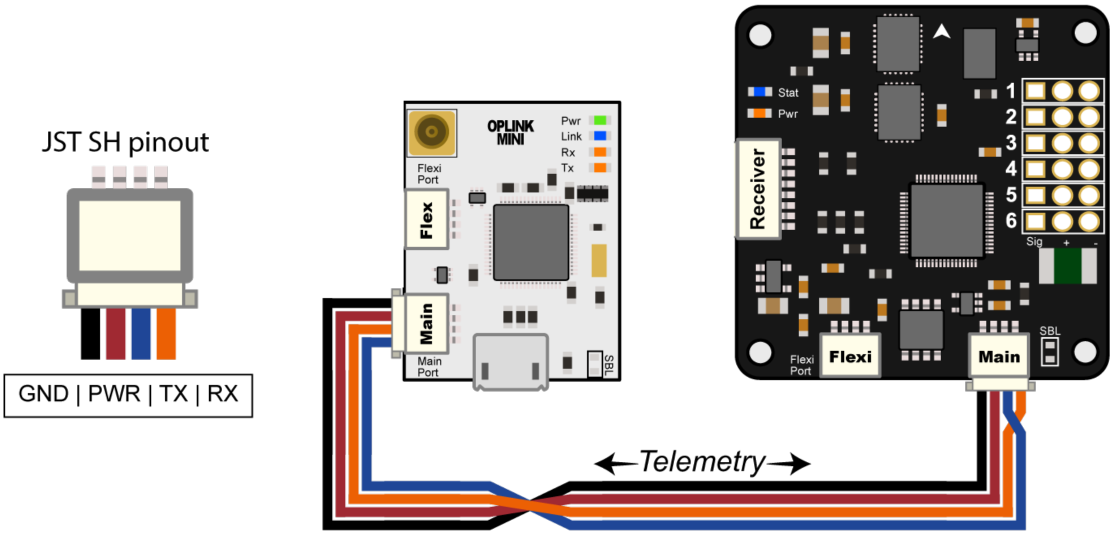
This page will walk you through how to set up the hardware connections that are required for CopterControl, CC3D and Atom OPLink operation. All OpenPilot boards support connection with the OPLink module. The OPLink connection can be used for telemetry and/or vehicle control.

The OPLink module will get it's power from the flight controller's external power, so unlike Revolution, the modem will not get powered on by just plugging in the USB to the flight controller. Make sure that you have the same OpenPilot firmware versions on both the flight controller and OPLink modem. All the examples below are shown with a CC3D flight controller board, but they can be applied to CopterControl and Atom, also.

Hardware connections

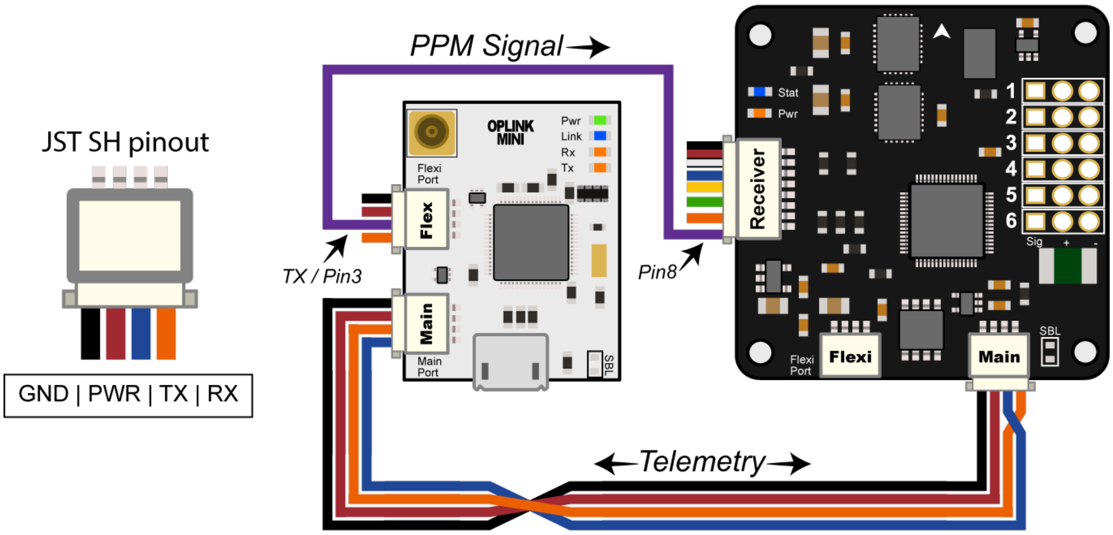
Telemetry only

The OPLink will be set up to talk bidirectional telemetry with the flight controller. You can use a regular OpenPilot 4-pin JST SH cable to connect the two boards. It is recommended to use Main port for both boards, but you can also use other ports to connect telemetry. Just remember the changes in the Configuration steps below. Connect the boards like in the diagram:



Telemetry and vehicle control

If you also want to use OPLink modem for vehicle control, you'll need to add one additional wire from the TX pin of the OPLink to the PPM input pin on the flight controller. In this diagram the wire goes to PIN 8 on the controller, because that is the pin that allows OneShot125 and PWM Sync with CC3D. Revolution PPM remains on PIN 3.

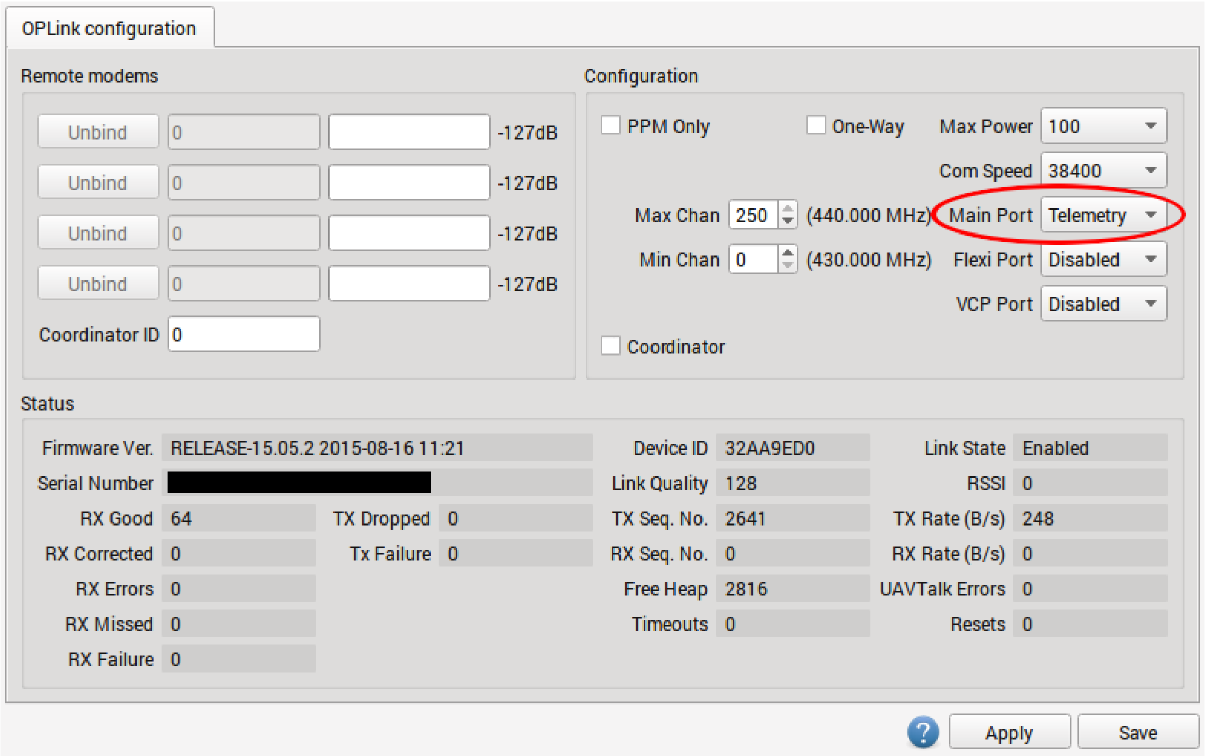


Configuration

Change the flight controller and OPLink Mini configuration to reflect the wiring changes that have been made. This is where it all becomes pretty rational. After these configuration steps, you are ready to proceed with OPLM Binding and Basic Telemetry.

Flight-side OPLink modem configuration

- Connect to the OPLink mini modem with USB.
- From the Configuration tab, Navigate to the **OPLink page** in the GCS.
- From the **Main Port** drop down menu, select **Telemetry**.
- Click **Save**, wait a few seconds for the telemetry gadget (the meter at the bottom of the GCS) to calm down, and disconnect from the modem.



OPLink configuration

Remote modems

Unbind	0		-127dB
Unbind	0		-127dB
Unbind	0		-127dB
Unbind	0		-127dB

Coordinator ID 0

Configuration

□ PPM Only □ One-Way Max Power 100

Com Speed 38400

Max Chan 250 (440.000 MHz) Main Port Telemetry

Min Chan 0 (430.000 MHz) Flexi Port Disabled

VCP Port Disabled

□ Coordinator

Status

Firmware Ver.	RELEASE-15.05.2 2015-08-16 11:21			Device ID	32AA9ED0	Link State	Enabled
Serial Number	[redacted]			Link Quality	128	RSSI	0
RX Good	64	TX Dropped	0	TX Seq. No.	2641	TX Rate (B/s)	248
RX Corrected	0	Tx Failure	0	RX Seq. No.	0	RX Rate (B/s)	0
RX Errors	0			Free Heap	2816	UAVTalk Errors	0
RX Missed	0			Timeouts	0	Resets	0
RX Failure	0						

Flight controller configuration

- Connect to the flight controller with USB.
- From the Configuration tab, navigate to the **Hardware page** in the GCS.
- From the **Main Port** drop down menu, select **Telemetry**
- From the **Telemetry Speed** drop down menu, select **38400**
- Click **Save**, wait a few seconds for the telemetry gadget (the meter at the bottom of the GCS) to calm down, and then disconnect from the flight controller.

1.9.5 OPLM Binding and Basic Telemetry

Summary

OPLink radios can be used in several different ways: for telemetry connection only, for telemetry + vehicle control, and for vehicle control only. The common thread for all these uses is *binding*, which also leads to basic telemetry data being transmitted over the link automatically.

This procedure will bind an OPLink mini to another OPLink, either to the integrated OPLink of the Revolution, or to another independent OPLink module. If you are binding two stand-alone OPLink modules together, it is recommended to first go through the OPLM CC - CC3D - Atom Hardware Setup page in order to prepare hardware for the binding operation. For Revolution, port setup or hardware changes are not necessary.

We will use the ground OPLink as a link coordinator, and the flight OPLink as a link slave. This enables full telemetry capabilities and makes it possible to use the link for vehicle control. **If you are going to use OPLink for vehicle control, do this tutorial first, then proceed to the vehicle control tutorial**. If you only intend to use telemetry, you can also make the flight OPLink the coordinator, but this does not have any additional benefits.

Antennas

> **Caution:** From this point on, you should always have antennas connected to all OPLink modules. Neglecting this will cause permanent damage to the RF electronics.

Prerequisites

- Make sure that all modules are up to date by using the GCS Firmware Update & Erase feature, found in the Firmware tab. OPLink must be disconnected before clicking the update button. The OPLink that is integrated into Revolution will be updated when you update the flight controller firmware.
- If you are connecting a ground OPLink to another stand-alone module, set up the hardware first by following the OPLM CC - CC3D - Atom Hardware Setup page.
- To protect your hardware from short circuits, consider mounting the OPLink module inside a case or at least wrap it with heat shrink. **DO NOT** mount a bare board so that it can touch carbon fiber frame without insulating it first (carbon fiber is conductive).
- OpenPilot firmware can only support one Telemetry link at a certain time, so disable Telemetry going to OSD or Bluetooth module etc. **DO NOT** disable USBTelemetry.
- *Check, and* **re-check** *that there is an antenna connected to both OPLink modules!*

Binding

Coordinator side

1. Connect to the OPLink module that you are going to use as Coordinator, usually the ground module.
2. Go to the OPLink page in the GCS, available on the Configuration tab's left bar. It is only visible if an OPLink or Revolution is connected.
3. Set the following recommended settings:
   (a) "Max Power" **100** (or maximum for your country)

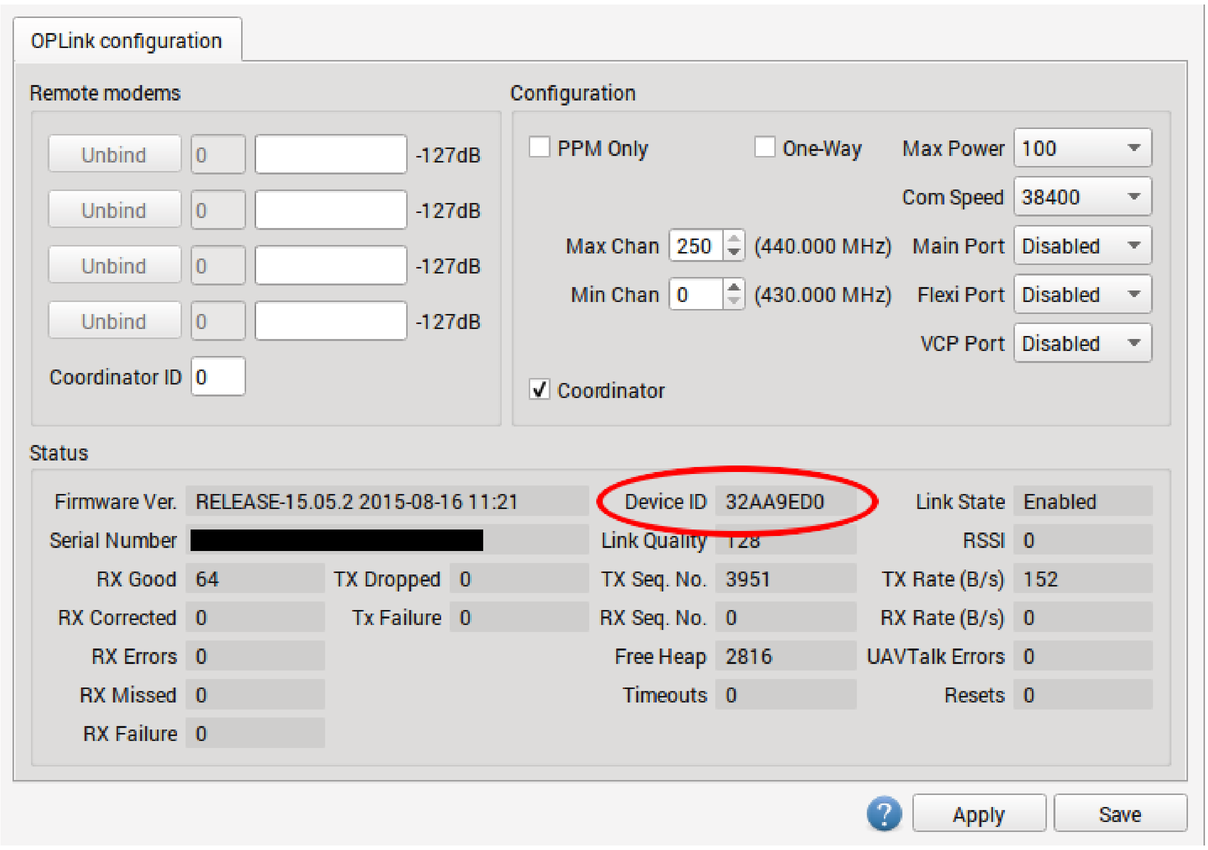
   (b) "Com Speed" **38400**

   (c) "Max chan" **250**

   (d) "Min chan" **0**

   (e) Tick the **Coordinator** checkbox

   (f) Click **Save**, wait a few seconds for the telemetry gadget (the meter at the bottom of the GCS) to calm down.
4. Write down the Device ID. You are going to use it later.
5. Disconnect from the coordinating module.



OPLink configuration

Remote modems

Unbind	0		-127dB
Unbind	0		-127dB
Unbind	0		-127dB
Unbind	0		-127dB

Coordinator ID 0

Configuration

□ PPM Only □ One-Way Max Power 100

Com Speed 38400

Max Chan 250 (440.000 MHz) Main Port Disabled

Min Chan 0 (430.000 MHz) Flexi Port Disabled

VCP Port Disabled

☑ Coordinator

Status

Firmware Ver.	RELEASE-15.05.2 2015-08-16 11:21			Device ID	32AA9ED0	Link State	Enabled
Serial Number	[redacted]			Link Quality	128	RSSI	0
RX Good	64	TX Dropped	0	TX Seq. No.	3951	TX Rate (B/s)	152
RX Corrected	0	Tx Failure	0	RX Seq. No.	0	RX Rate (B/s)	0
RX Errors	0			Free Heap	2816	UAVTalk Errors	0
RX Missed	0			Timeouts	0	Resets	0
RX Failure	0						

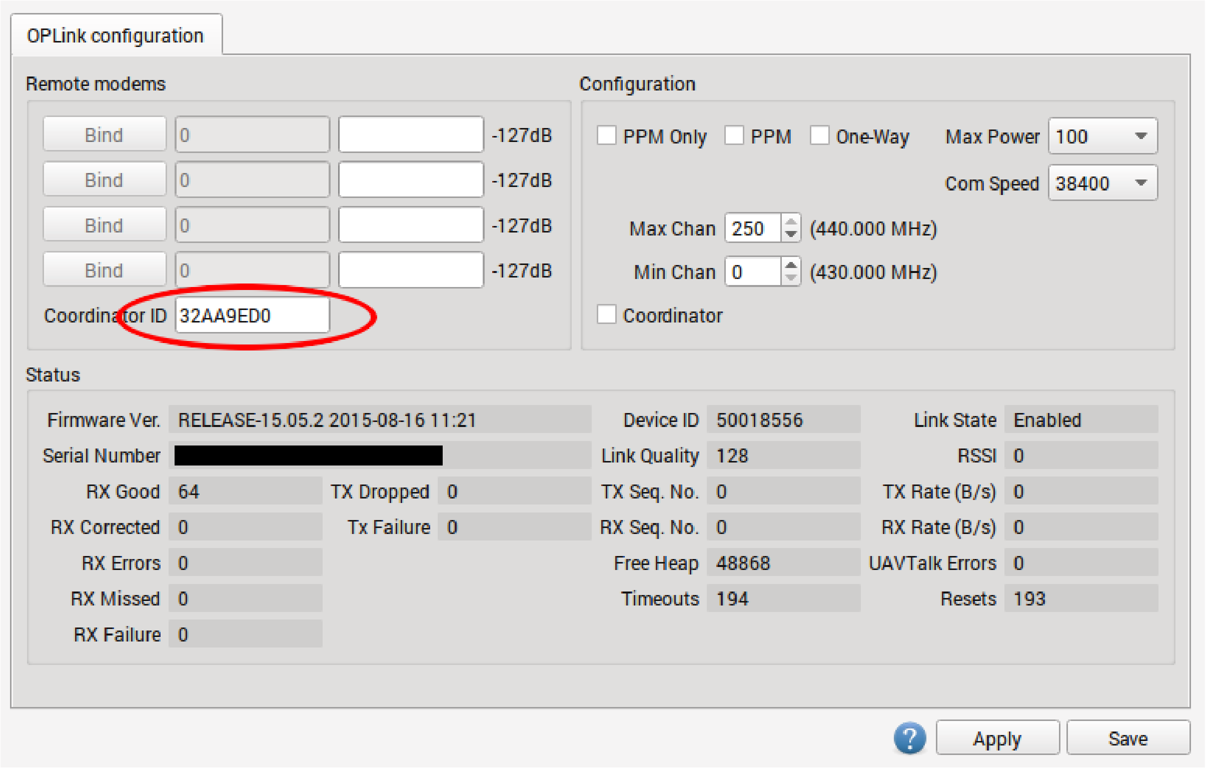
The OPLink Save button will display a green "tick" when settings are saved, but it can disappear soon after settings are saved. This is normal behavior.

Using custom frequencies

If your country allows only a fraction of the available frequency band to be used, you can adjust the operational OPLink channel range to reflect that. The GCS indicates minimum and maximum used frequency when you change min and max channels. Both OPLinks must have the same min and max channel pair to bind successfully.

Slave side

1. Connect to the another OPLink device that is going to be the connection slave
2. On the OPLink page in the GCS, set exactly the same **Max Power**, **Com Speed**, **Max chan** and **Min chan** as you did for the Coordinator. **DO NOT** tick the "Coordinator" checkbox.
3. Enter the Device ID that you wrote down from Coordinator into the **Coordinator ID** text box.
4. Click **Save**, wait a few seconds for the telemetry gadget (the meter at the bottom of the GCS) to calm down.
5. Disconnect from the slave module. The bind is complete.



OPLink configuration

Remote modems

Bind 0 -127dB

Bind 0 -127dB

Bind 0 -127dB

Bind 0 -127dB

Coordinator ID 32AA9ED0

Configuration

□ PPM Only □ PPM □ One-Way Max Power 100

Com Speed 38400

Max Chan 250 (440.000 MHz)

Min Chan 0 (430.000 MHz)

□ Coordinator

Status

Firmware Ver. RELEASE-15.05.2 2015-08-16 11:21 | Device ID 50018556 | Link State Enabled

Serial Number [redacted] | Link Quality 128 | RSSI 0

RX Good 64 | TX Dropped 0 | TX Seq. No. 0 | TX Rate (B/s) 0

RX Corrected 0 | Tx Failure 0 | RX Seq. No. 0 | RX Rate (B/s) 0

RX Errors 0 | Free Heap 48868 | UAVTalk Errors 0

RX Missed 0 | Timeouts 194 | Resets 193

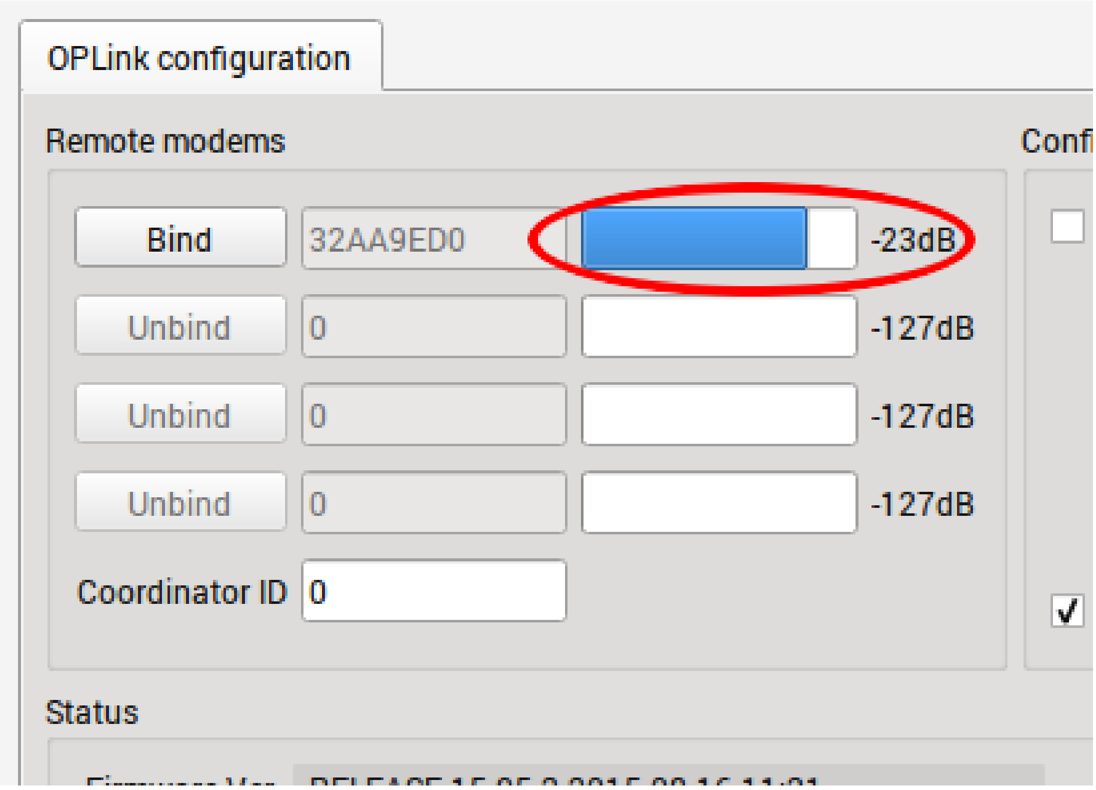
RX Failure 0

Testing the link for the first time

At this point, the OPLink modules have a successful bind and no more configuration steps are needed. The next step is to actually test and verify that the connection between the OPLink devices works properly. All OpenPilot devices should be powered off and disconnected from your PC before testing. This is because the modules need a proper power cycle to apply the changes we made, and because only one telemetry connection can be supported at a time. This means that if you have USB connected to the flight controller, only the USB connection gets telemetry, and the OPLink does not.

1. Power on your vehicle using external power (i.e., a flight battery).
2. Connect to the ground module with USB and observe GCS behavior as communication link comes online. The link usually takes between 10 and 30 seconds to establish, because there are a lot of settings objects that need to be downloaded through the link. You can observe the download process in System tab.
3. Go to the **OPLink Configuration** page, and you should see a blue bar in the **Remote modems** area, which indicates signal strength.

Congratulations! Your OPLink telemetry connection is now ready for use. If you wish to use the connection for vehicle control also, proceed to the OPLM Vehicle Control Link page.



32AA9ED0 -23dB

0 -127dB

0 -127dB

0 -127dB

Coordinator ID 0

Advanced topics

Binding to multiple vehicles

You can use a single coordinator OPLink module to bind to multiple vehicles that act as signal slaves. They cannot be powered on simultaneously, but this allows you to get telemetry from many vehicles with just one ground module and no configuration changes. It even supports controlling multiple vehicles. This is established by repeating the slave side binding procedure for every additional OPLink.

Multiple Slaves

> **Caution:** Use caution that only one vehicle at a time is powered with a flight battery, or you may have a second vehicle power up unexpectedly when you arm!

Using higher COM speed

OPLink supports higher COM speeds than 38400. If necessary, you can safely bump it up to 57600. After that, the range of the link decreases significantly. This makes it possible to transmit larger amounts of data and raise the update periods of various data objects (manually, in data object meta data). With faster data rate, ridiculous object update speeds can be achieved, but at the cost of range.

Using Bluetooth module with OPLink

It is possible to connect a Bluetooth module to OPLink, and establish a completely wireless telemetry link. In this setup, telemetry data is first transmitted between OPLink modules, and then from ground OPLink module to Bluetooth module, and finally a computer or a smartphone with integrated Bluetooth chip. Setup instructions can be found in the Bluetooth Setup for Telemetry page.

One way operation

If for some reason you want to only transmit with the other module, and receive with the other one, you can tick **"One way"** in both OPLink module settings. It will enable a special mode where only the coordinator module can transmit packets. The slave module receives packets normally.

Vehicle control link only

For those extra long or safety critical flights OPLink can be brought into a special mode where nothing but PPM vehicle control signal is transmitted from coordinator to slave. This enables a custom communication speed and protocol in OPLink. You can enable it by ticking **"PPM Only"** at both OPLink module configs. The control link has to be set up normally using OPLM Vehicle Control Link page instructions.

1.9.6 OPLM Vehicle Control Link

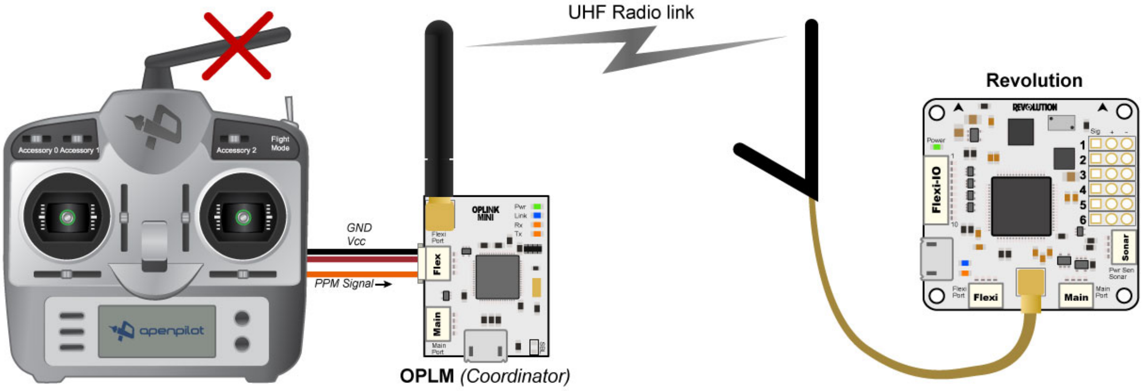
Summary

These instructions will walk you through on how to use your existing OPLink telemetry connection for vehicle control. The instructions consist of hardware and software side. Set up the hardware side first and then do the necessary hardware configuration changes. If you have not done binding yet for the OPLink modules, do that first using OPLM Binding and Basic Telemetry page instructions, and return to this tutorial after that.

For Revolution, no hardware configuration is required at the flight controller side. For CC, CC3D and Atom, a PPM wire has to be connected from the OPLink module to the flight controller. The instructions for how to prepare those flight controllers for control link can be found on OPLM CC - CC3D - Atom Hardware Setup page. If you do not have the PPM wire connected, do it now, and then return to this page.

OPLink wiring

You will need to supply power and control data from the transmitter to the OPLink module. The OPLink Mini module requires three wires for normal operation with a transmitter.



- Ground
- Voltage (recommended 5V - 8.4V)
- PPM Out from the transmitter

The voltage wire, Vcc in the picture, is a bit tricky. A regular transmitter usually uses 3S LiPo for power and it outputs that 3S voltage to the trainer port and radio module connections. That voltage is too high to be directly used. You should use a linear regulator or UBEC to bring the voltage down to 5 volts.

The PPM wire can be directly connected and does not need anything in between the OPLink module and transmitter. You can cut a regular OpenPilot 4-pin serial cable in half and use that to connect the wires to your OPLink module. The blue wire in the 4-pin cable is not used for this application, so you can remove it.

OPLink PPM input

OPLink module is programmed to accept PPM sum between 4 and 8 channels. Recommended PPM frame length is 22,5 milliseconds.

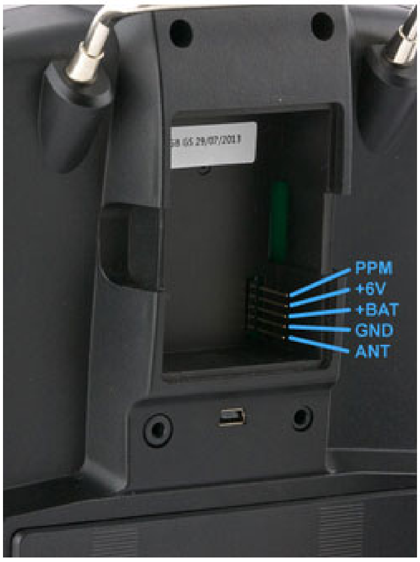
OPLink input voltage

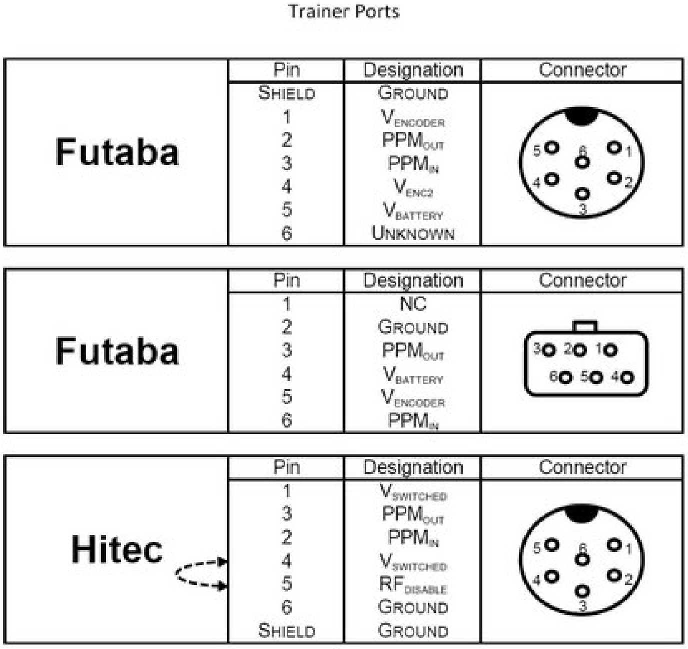
Do not input higher than 8.4 volts (2S) to the OPLink module! Higher voltage will cause permanent damage to the electronics.

Pinouts for connecting OPLink to the transmitter

Here are two pinout diagrams that might help to connect the OPLink module to your transmitter. The JR type module bay outputs PPM, which can be connected directly to your OPLink module. A 5V UBEC is needed between +BAT and GND. You can connect the UBEC output to OPLink Vcc and GND. The +6V wire can be used to power the OPLink directly if your radio has it, but most radios don't.

The Futaba and Hitec trainer ports have similar pins that you can use, called GROUND, PPMout and Vbattery in the diagram. A nice trick is to cut a trainer cable in half and use that to make a neat connection to the transmitter. If you cut a trainer cable, you'll have to identify the correct wires with a multimeter.

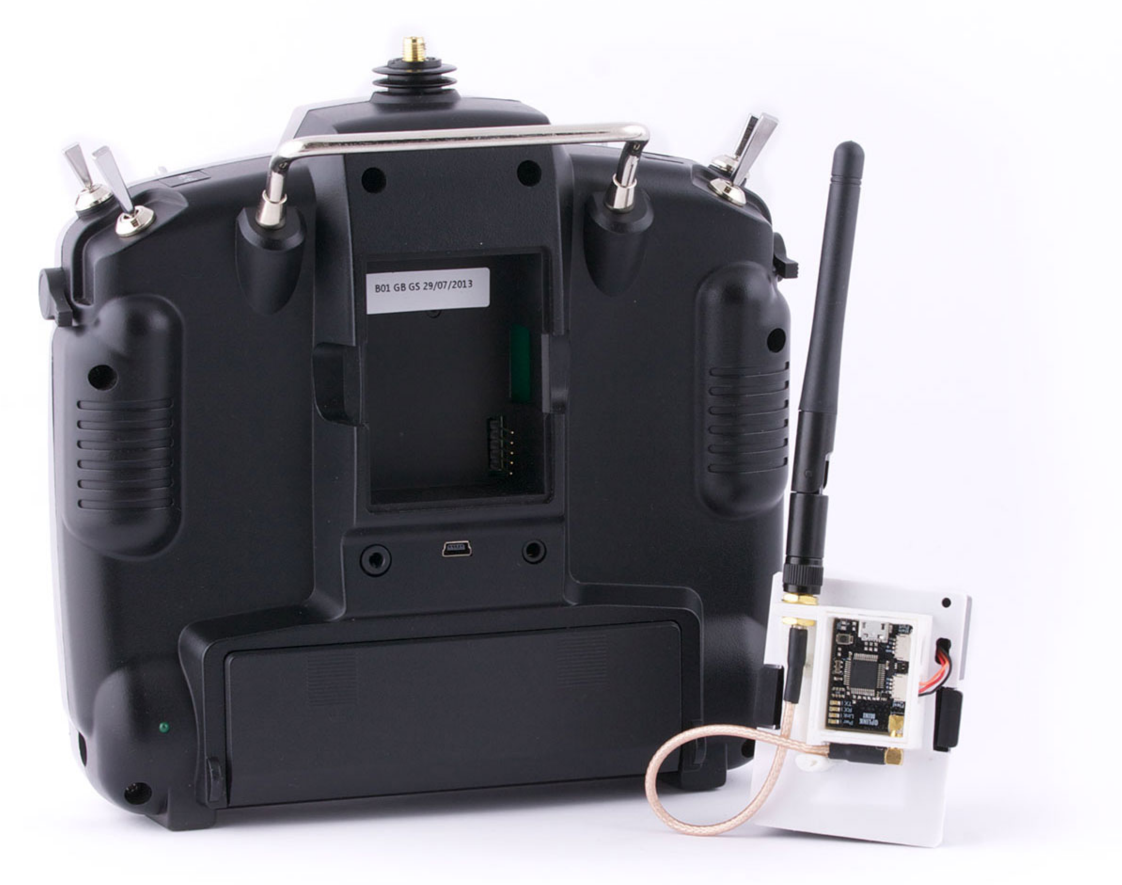




Trainer Ports

	Pin	Designation	Connector
Futaba	SHIELD	GROUND	
	1	$V_{ENCODER}$	
	2	PPM_{OUT}	
	3	PPM_{IN}	
	4	V_{ENC2}	
	5	$V_{BATTERY}$	
	6	UNKNOWN	
	Pin	Designation	Connector
Futaba	1	NC	
	2	GROUND	
	3	PPM_{OUT}	
	4	$V_{BATTERY}$	
	5	$V_{ENCODER}$	
	6	PPM_{IN}	
	Pin	Designation	Connector
Hitec	1	$V_{SWITCHED}$	
	3	PPM_{OUT}	
	2	PPM_{IN}	
	4	$V_{SWITCHED}$	
	5	$RF_{DISABLE}$	
	6	GROUND	
	SHIELD	GROUND	

Mounting the OPLink module to the transmitter



There are many ways to attach the OPLink mini module to your transmitter. The options listed here are just ideas and possible inspiration for the job.

- You can install the OPLink module inside an existing JR or Futaba transmitter case
- 3D print a JR module: http://www.thingiverse.com/thing:585245
- Heat shrink the module and attach it to the transmitter with velcro
- Install the OPLink and UBEC inside the module bay without a case
- Invent a new method and contact a forum moderator to add your idea here!

1.9.7 Bluetooth Setup for Telemetry

Summary

These instructions will walk you through how to use a serial Bluetooth module, such as HC-05 and HC-06, with OpenPilot hardware to establish a telemetry connection. There are two ways to connect the module; either directly to the flight controller, or to an OPLink ground module. Connecting to the flight controller makes it possible to configure and calibrate the flight controller near a PC or a smartphone, but as a Bluetooth connection does not travel long distances, the link is usually lost during flight. OPLink solves this problem, and communicates telemetry data from the vehicle to ground. Then Bluetooth connection can be used to connect a PC or a smartphone to the OPLink ground module. This is especially useful when using OPLink for vehicle control + telemetry.

No matter which hardware configuration you choose, the Bluetooth module's configuration procedure is basically the same. HC-05 and HC-06 Bluetooth modules are factory configured to 9600 baud rate, 8 data bit, 1 stop bit and no parity serial connection. The baud rate must be increased to 38400 to match with that of the OPLink.

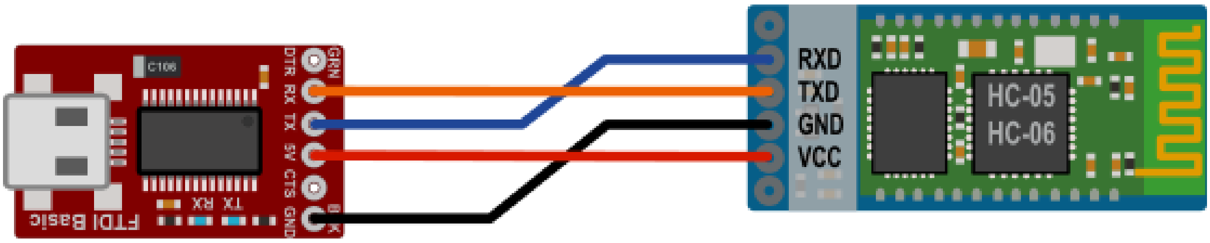
Alternatively, 57600 baud can be used if you are connecting the Bluetooth module directly to the flight controller, or if you use 57600 baud with OPLink.

Configuring the Bluetooth module

Hardware connections

To change the baud rate of the Bluetooth module, you must connect to the module with a serial link. You can either use an FTDI USB to serial adapter, or your OpenPilot flight controller to connect to the Bluetooth module. Both establish the same goal. If you have an FTDI adapter it is recommended to use that, because it does not require any configuration steps like the flight controller does for Virtual COM port ComBridge operation.

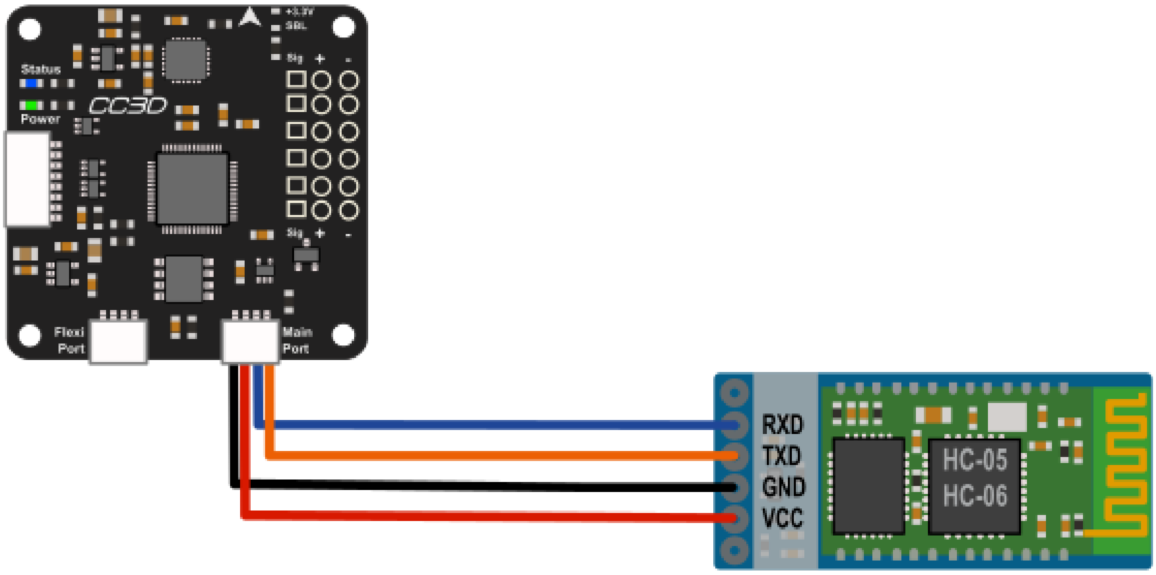
HW method 1: Using an FTDI adapter



Most R/C hobbyists are familiar with FTDI USB to serial adapters. This is probably the easiest method to connect to the Bluetooth module. An FTDI module can easily be found on eBay.

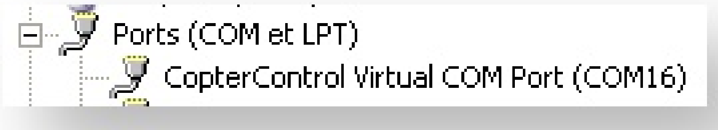
1. Connect the wires according to the diagram.
2. Connect USB cable to the FTDI adapter.
3. Install any necessary FTDI drivers on your PC, or wait for your operating system to install them.
4. COM port becomes available automatically, and can be seen in the Windows Device Manager. Proceed to the next step below: **Software steps**.

HW method 2: Using the flight controller Virtual COM port



OpenPilot flight controllers can work as a USB-to-serial adapter with a feature called Virtual COM port (VCP). The configuration for VCP ComBridge is fairly straightforward. The VCP ComBridge feature works with all OpenPilot flight controllers.

1. Connect the wires according to the diagram.
2. Connect USB cable to the flight controller.
3. Set the following settings:
   - USB VCP Function **ComBridge**
   - Main Port **ComBridge**
   - Speed **9600**
4. Click **Save**, and wait a few seconds for the changes to apply.
5. Reboot the flight controller, and the Virtual COM port should appear in the Windows Device Manager.



Ports (COM et LPT)
CopterControl Virtual COM Port (COM16)

6. Proceed to the next step below: Software steps.

Note: To power up the Bluetooth chip for configuration, the flight controller has to be powered up externally, with a flight battery and UBEC, for example.

Software steps

In this section, the Bluetooth module will be configured to correct the baud rate. You can also customize the Bluetooth unit's name. The Bluetooth module expects to be called AT commands in the COM port, which you can either write in a command line, or automatically send with a Windows tool that former OpenPilot developer PT_Dreamer wrote. Connect your Bluetooth module to your PC with either the FTDI or VCP ComBridge. VCP ComBridge usage requires external power to the flight controller.

If you have not decided on the baud rate you want to use for the Bluetooth module, now is the time. Recommended settings are 38400 for usage with OPLink and 57600 for usage directly with your flight controller. Remember the setting, it is a good idea to use a marker pen and write it on the Bluetooth module.

Configuration method 1: Windows configuration software

1. Download Bluetooth configuration software by PT_Dreamer from the attachment below.
2. Extract the files and launch the configuration program by clicking on the .exe file.
3. Select **COM port** and baud **9600**, and click **Connect**.
4. Select the appropriate **Speed** (baud rate; i.e., 38400 or 57600), **Name** and **Pin** code.
5. Click **Write values**.
6. Configuration is done.
7. Undo the USB VCP ComBridge changes if you used a flight controller for the connection.

Configuration software download: `/files/BT_Cong_Qt4.zip`

Configuration method 2: Command line

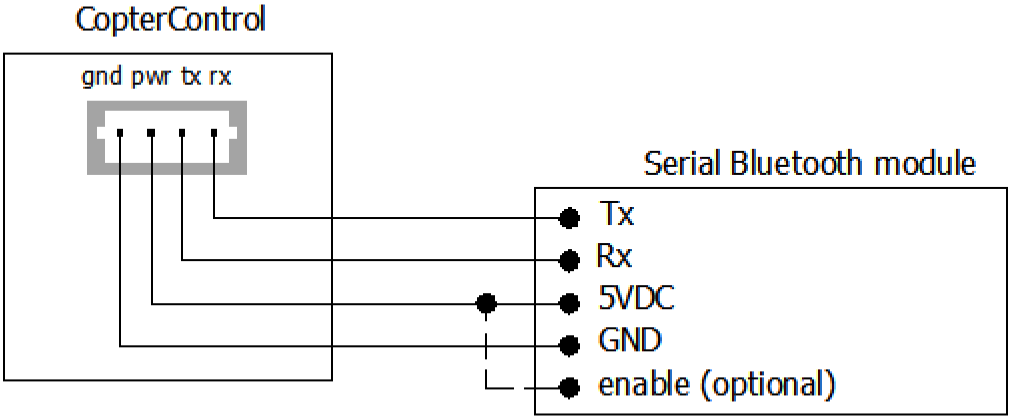
While the configuration software above is by far the easiest method on a Windows system, Mac and Linux users will have to use command line to configure the Bluetooth module. There are many terminal applications you can choose from. One option for Windows and Linux is PuTTY, and for Mac CoolTerm. No matter which terminal application you use, connect to the COM port that the FTDI or VCP ComBridge is in with 9600 baud rate, 8 data bit, 1 stop bit & no parity serial connection.

When serial connection is open in the terminal application, proceed with the following steps, all CAPS:

1. Send "**AT**", the module should answer OK.
2. Send "**AT+NAMEOpenPilot**", this sets the module's name to "OpenPilot".
3. Send "**AT+BAUD6**" for 38400 baud rate OR "**AT+BAUD7**" for 57600 baud rate.
4. Close the terminal application.
5. Configuration is done.
6. Undo the USB VCP ComBridge changes if you used a flight controller for the connection.

Connecting the Bluetooth module to OpenPilot devices

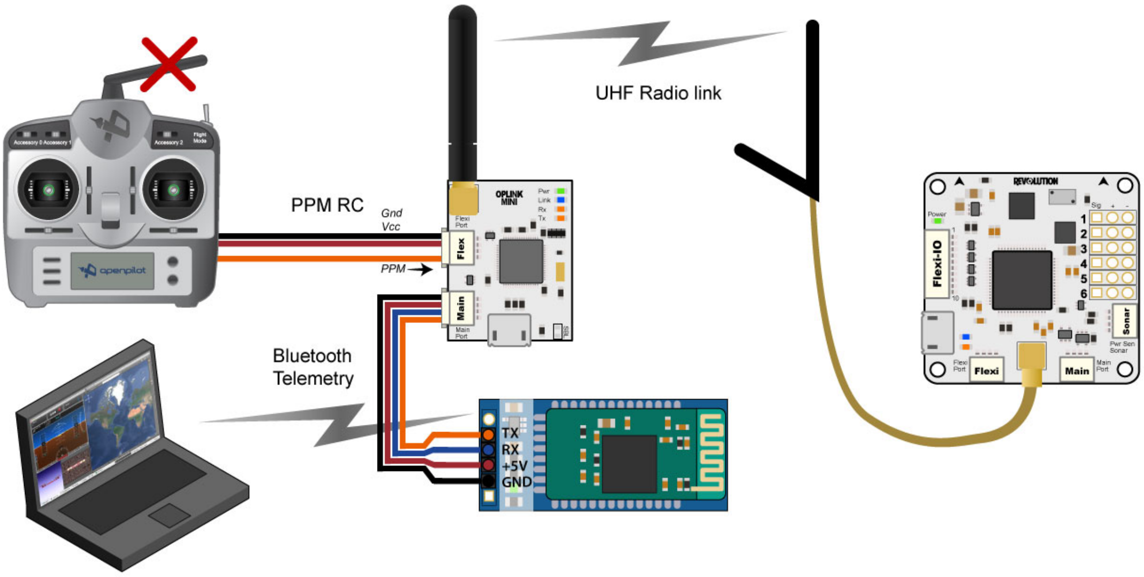
Connecting directly to the flight controller



For direct connection from the flight controller to the Bluetooth module, the wiring is exactly the same as USB VCP ComBridge wiring. The flight controller has to be configured to output telemetry data at the correct baud rate.

1. Connect the wiring according to the diagram above.
2. Connect the USB to your flight controller.
3. Navigate to the **Hardware** tab on the **Configuration** page in the OpenPilot GCS.
4. Select **Telemetry** in the port where your Bluetooth module is connected (usually **Main Port**).
5. Select **57600** (or another speed that your Bluetooth module is configured to) as **Speed**.
6. Click **Save** and wait a few seconds for changes to apply.
7. Disconnect from the flight controller. You are now ready to test the link.

Connecting to OPLink ground module



For completely wireless telemetry with OPLink and Bluetooth module, the wiring has to be connected according to the diagram above. OPLink vehicle control is not necessary for Bluetooth operation, but that is the most common use scenario. Completely wireless telemetry also works with CC/CC3D/Atom, as long as the flight side OPLink wiring has been configured using the OPLM CC - CC3D - Atom Hardware Setup instructions, and you have a successful bind. When you have done the wiring, proceed with the following steps. The baud rate of the Bluetooth module has to be the same as the OPLink radio baud rate. Remember to power up the OPLink via Flexi or Main port, because powering it from USB port redirects Telemetry temporarily to USB.

1. Connect your ground OPLink module to your PC with USB .
2. Navigate to the **OPLink** tab on the **Configuration** page in the OpenPilot GCS.
3. Select **Telemetry** in **Main Port** of the ground OPLink module.
4. Click **Save**, and wait a few seconds for the changes to apply.
5. Disconnect from the OPLink ground module. You are now ready to test the link.

Todo

FIXME: screenshot

Connecting to the flight controller via Bluetooth telemetry

Your Bluetooth telemetry link should now be ready to test.

Important: Power up all the related devices such as the flight controller and possible OPLink modules using external power and **not** USB. This is important, because if any OpenPilot device is connected via USB, other Telemetry connections will be temporarily disabled until the USB is unplugged.

Connect to the Bluetooth module using your computer's Bluetooth utilities. Use a new Pin if you changed it, or the usual default of 1234.

1. Connect to the Bluetooth module using your computer's Bluetooth utilities. Use a new Pin if you changed it, or the usual default of 1234.

2. Launch the OpenPilot GCS and select a serial COM port from the connections drop-down menu in the bottom bar. Then click **Connect**. You might have to try multiple ports to find the right one. This is the COM port that your PC's internal Bluetooth chip automatically creates when connection to your Bluetooth module is established.
3. **Success!** In the example screenshot above, the GCS is talking to an OpenPilot Revolution via OPLink <-> Bluetooth connection that is available on Serial COM13 port. The COM port number will most likely be different for you.

Todo

FIXME: screenshots

Connecting

The OpenPilot GCS will not automatically connect to Bluetooth devices. The connection has to be opened manually by selecting a COM port, and clicking **Connect**.

(Virtual Com Bridge information was based on an OP Forum post by Zucky)

Images was taken from LibrePilot Wiki

Introduction

Warning: OPENPILOT WAS DISCONTINUED!
Use LibrePilot instead.

We've designed this section and the LibrePilot/OpenPilot Documentation as a whole to flow like a story, guiding you through all the steps necessary for successful setup and flight.

New users will use this section to learn the basics of the system and progress to their first flight, while experienced users can refresh their memory when setting up a new board, or gain insight into more detailed functions and configurations covered in the advanced sections of the manual.

As a new user, try to take things one step at a time, take advantage of video tutorials where available, and follow the manual in sequential order. We've provided links to the previous page, next page, and Table of Contents whenever possible to simplify navigation as you progress through the manual.

Let's get started!

CHAPTER 2

Appendicies

2.1 Bootloader Update

2.1.1 Introduction

The bootloader is a small piece of software resident on the flight controller which is started as soon as the board is powered. It performs hardware checks and loads and executes the firmware. It also handles the USB port to communicate with GCS to support bootloader and firmware updates.

You can check the version of the bootloader installed on any OP board to make sure it is the most up-to-date version and upgrade the bootloader software if required.

> **Warning:** Upgrading the bootloader can be tricky. If something goes wrong, you may render the flight controller inoperable. Happily, almost all bad mistakes can be rectified via the methods described on the "How to" sub-pages to this one.

2.1.2 How to Check the Bootloader Version

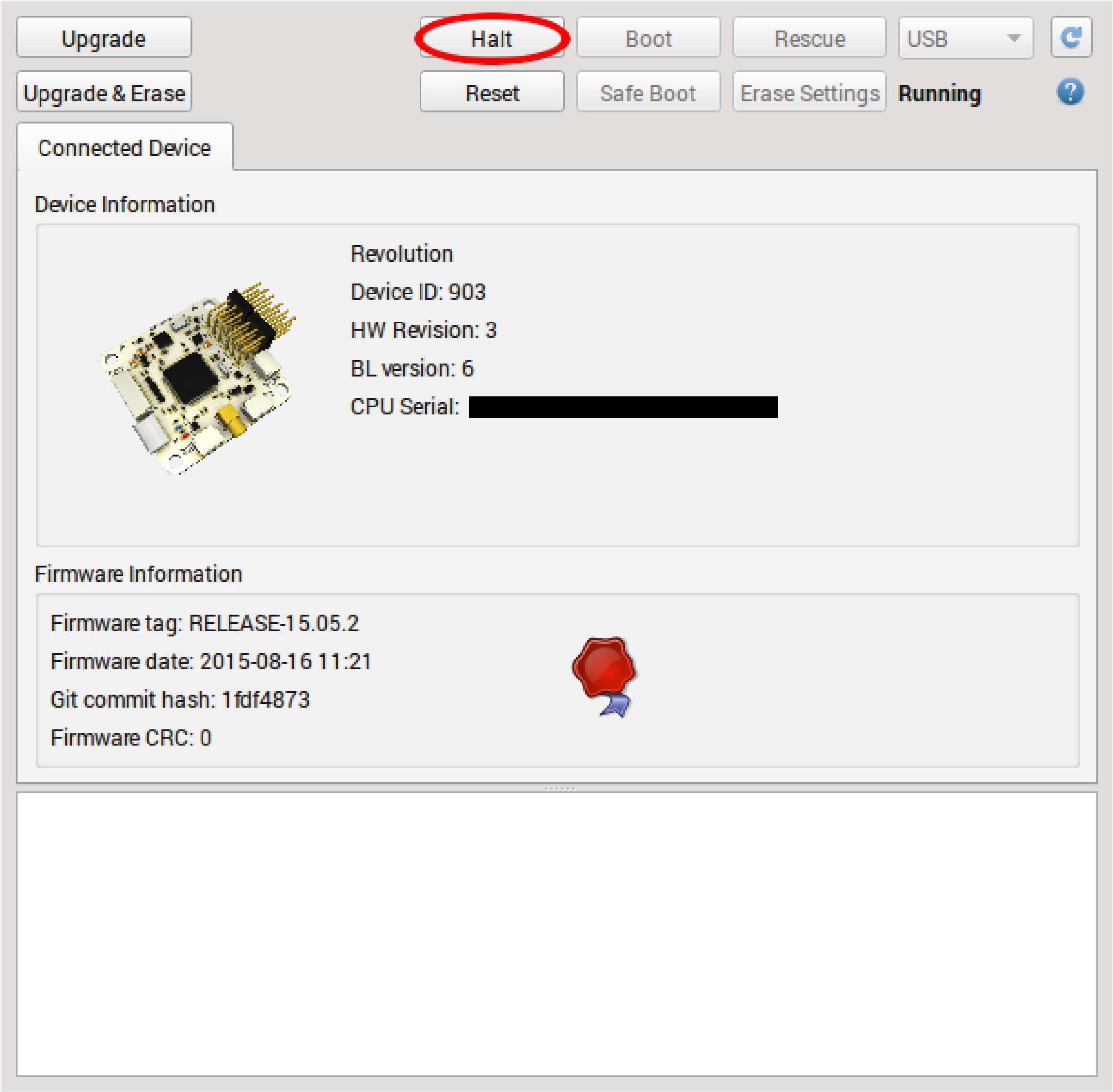
The bootloader version is shown when the flight controller is in boot mode. You can enter boot mode by:

- using either the **Halt** option or,
- by using **Rescue**.

Either method will reveal the bootloader version on your board and both methods are explained below.

Halt Option

Connect the flight controller to GCS via USB and open the Firmware page. On the Firmware page, click **Halt**.



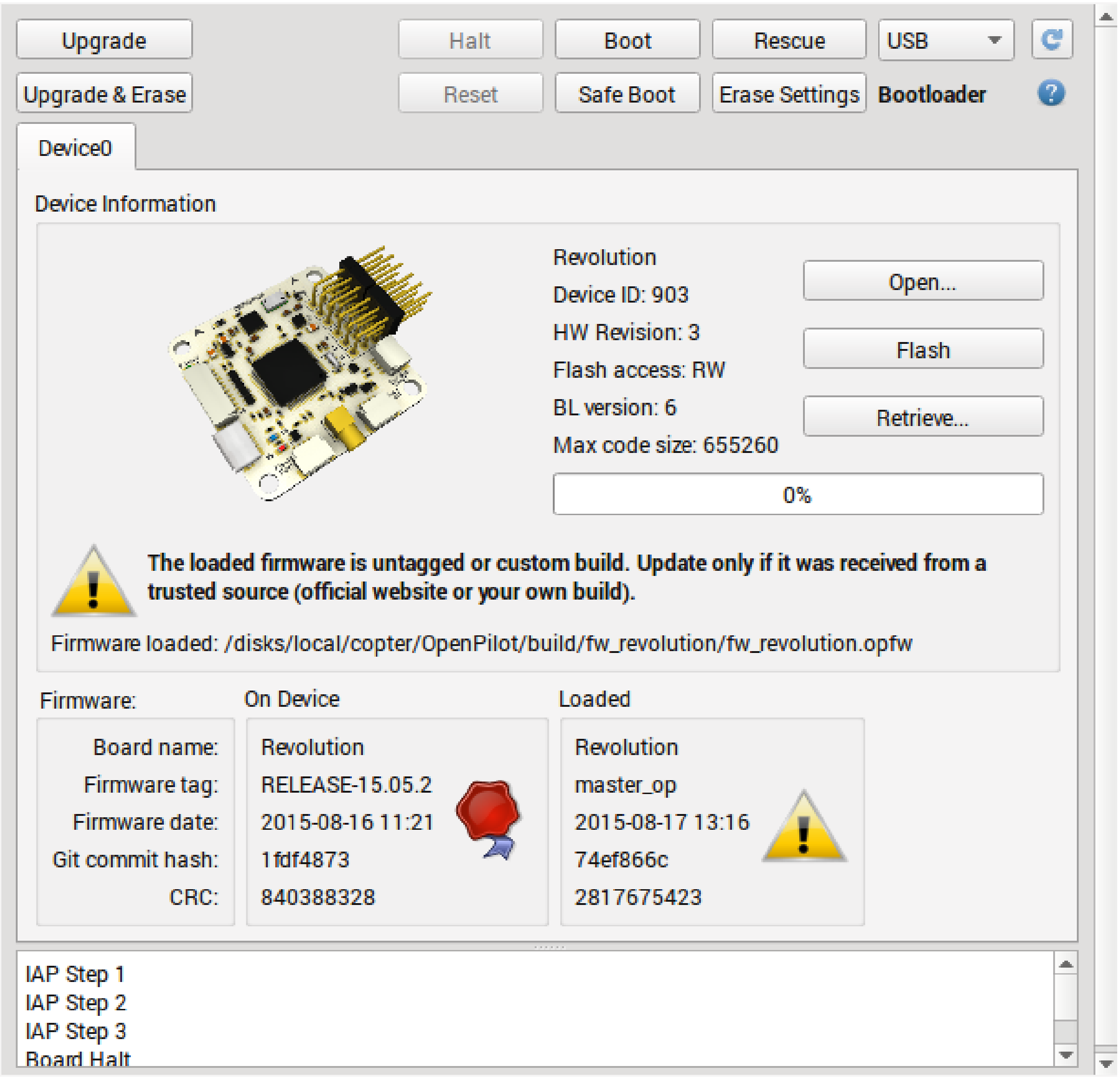
Device Information

Revolution
Device ID: 903
HW Revision: 3
BL version: 6
CPU Serial: [redacted]

Firmware Information

Firmware tag: RELEASE-15.05.2
Firmware date: 2015-08-16 11:21
Git commit hash: 1fdf4873
Firmware CRC: 0

Once the board is halted, the bootloader (BL) version of the board is shown.



Device Information

Revolution
Device ID: 903
HW Revision: 3
Flash access: RW
BL version: 6
Max code size: 655260

The loaded firmware is untagged or custom build. Update only if it was received from a trusted source (official website or your own build).

Firmware loaded: /disks/local/copter/OpenPilot/build/fw_revolution/fw_revolution.opfw

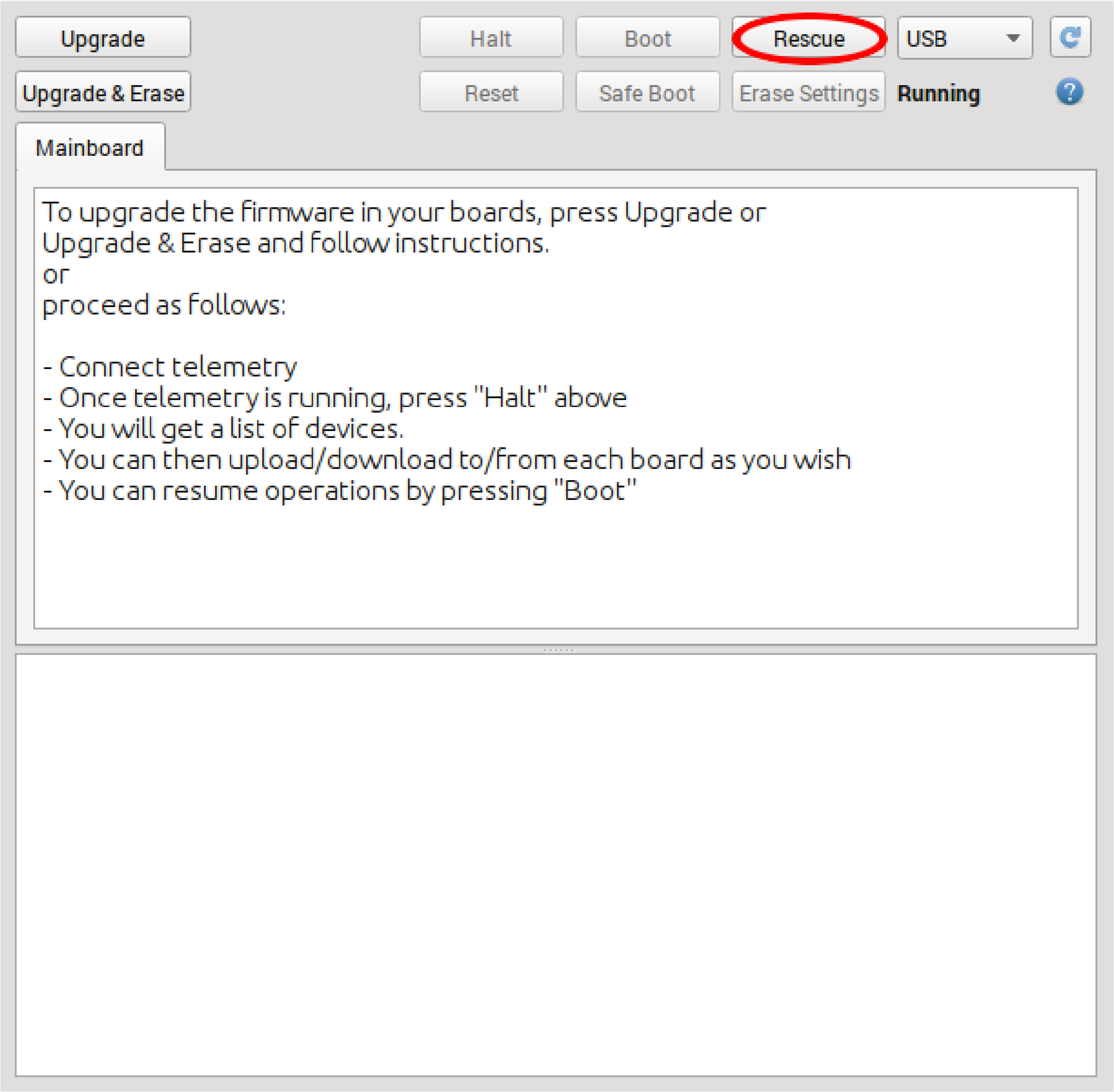
Firmware:	On Device	Loaded
Board name:	Revolution	Revolution
Firmware tag:	RELEASE-15.05.2	master_op
Firmware date:	2015-08-16 11:21	2015-08-17 13:16
Git commit hash:	1fdf4873	74ef866c
CRC:	840388328	2817675423

```
IAP Step 1
IAP Step 2
IAP Step 3
Board Halt
```

Click **Boot** to restart the flight controller once you have noted the bootloader version.

Rescue Method

Ensure that the flight controller is **not** connected to GCS via USB. Click **Rescue** and connect the flight controller via USB when prompted.



To upgrade the firmware in your boards, press Upgrade or
Upgrade & Erase and follow instructions.
or
proceed as follows:

- Connect telemetry
- Once telemetry is running, press "Halt" above
- You will get a list of devices.
- You can then upload/download to/from each board as you wish
- You can resume operations by pressing "Boot"

When the board is connected, its bootloader (BL) version is shown. Click **Boot** to restart the flight controller once you have noted the bootloader version.

2.1.3 Downloads

Bootloader Versions

- **Version 1** - First bootloader version which was loaded onto all CopterControl boards.
- **Version 2** - Safe boot capabilities were added to prevent the user from being locked out of the board due to a bad hardware configuration.
- **Version 3** - Added different USB serial numbers for bootloader and firmware. This helps Windows separate the two functions of the board by making it think there are two different boards connected. Twitching servo movement during the board start has been eliminated.
- **Version 4** - Added support for internal settings erasure so that no more special firmware is required; simply enter boot mode and click Erase Settings (CC/CC3D/Atom and OPLM)
- **Version 5** - Added better F4 processor support (Revo and OSD)

The bootloaders for CopterControl (CC), CopterControl3D (CC3D), Atom, Revolution (Revo), OPLink mini (OPLM) and On Screen Display (OSD) are available for download below.

> **Caution:** All OP boards are different and require the appropriate bootloader to be installed, please ensure you download and use the correct version listed below.

Also note that you should have the appropriate GCS installed (in most cases, the latest version) on your PC when you flash the new bootloader so that the followup Auto Update you perform will automatically install the correct version of firmware that is embedded in the version of the GCS on your PC.

The PCB for CC or CC3D boards should have “CopterControl” or “CC3D” printed clearly on the board. This indicates which bootloader needs to be flashed to the board.

Version 3 bootloaders can only be used with GCS versions 12.10.2 or older. Version 4 bootloaders can only be used with GCS versions 13.06.01 and newer.

Bootloader Updater Files

Board	BL Version	Updater Bootloader	Note
CC	3	`bu_coptercontrol-20120630_5a1efef.opfw`	For use with GCS 12.10.2 and lower
CC3D	3	`bu_CC3D-20120620_f44b9d3.opfw`	For use with GCS 12.10.2 and lower
CC	4	CC V4 `bu_cc.opfw`	For use with GCS 13.06.01 and higher
CC3D/Atom	4	CC3D V4 `bu_cc3d.opfw`	For use with GCS 13.06.01 and higher
Revolution	5	Revo V5 `bu_revolution.opfw`	For use with GCS 13.06.01 and higher
Revolution	6	Revo V6 `bu_revolution_6.opfw`	For use with LibrePilot
Revolution Nano	6	Revolution Nano V6 `bu_revonano.opfw`	For use with LibrePilot
OPLMini	4	OPLink Mini `bu_oplinkmini.opfw`	For use with GCS 13.06.01 and higher
OSD	5	OSD `bu_osd.opfw`	For use with GCS 13.06.01 and higher

2.1.4 How to Upgrade the Bootloader and Erase Settings

If it is necessary to upload the bootloader, **strictly** follow these instructions:

- Download the appropriate bootloader (ie CC3D - BL4 or Revo - BL5) and save it to your hard drive where you can find it again
- Using the **Firmware** workspace in GCS, and with the board disconnected from USB, click **Rescue** and follow the onscreen instruction to connect the board
- After the board is detected, click **Open** and select the BootloaderUpdater (BU) file from where you saved it on your hard drive
- Click **Flash** to flash it to your board
- After the flashing is complete, press **Boot** and wait until the blue LED is on, then flashes, and finally goes off (normally ±15 seconds).
- Wait 10 seconds more.
- Disconnect the board from USB.
- Click **Rescue**, connect the board, click **Erase Settings** and wait for the erasure to complete.

- Disconnect the board from USB.
- Click **Upgrade** and follow the onscreen instruction to connect the board to automatically install firmware.

2.1.5 LED Behavior

- A slowly blinking blue LED indicates that the board is booted and running the firmware; this is the normal operating mode.
- Bootloader mode. A slow fading in and out of the blue LED with the green LED on indicates that the board is in bootloader mode.
- A rapidly blinking blue LED during a bootloader update indicates an error state. An invalid bootloader image was likely detected and the update hasn't been performed.

2.1.6 FAQs

What's the difference between firmware, bootloader (BL) and bootloader updater (BU)?

The **firmware (FW)** is the application to be loaded by the bootloader after the board has been powered up and initialized. The firmware is regularly updated and newer firmwares typically include new features and bug fixes. The firmware and GCS version must match in order to be able to configure the board.

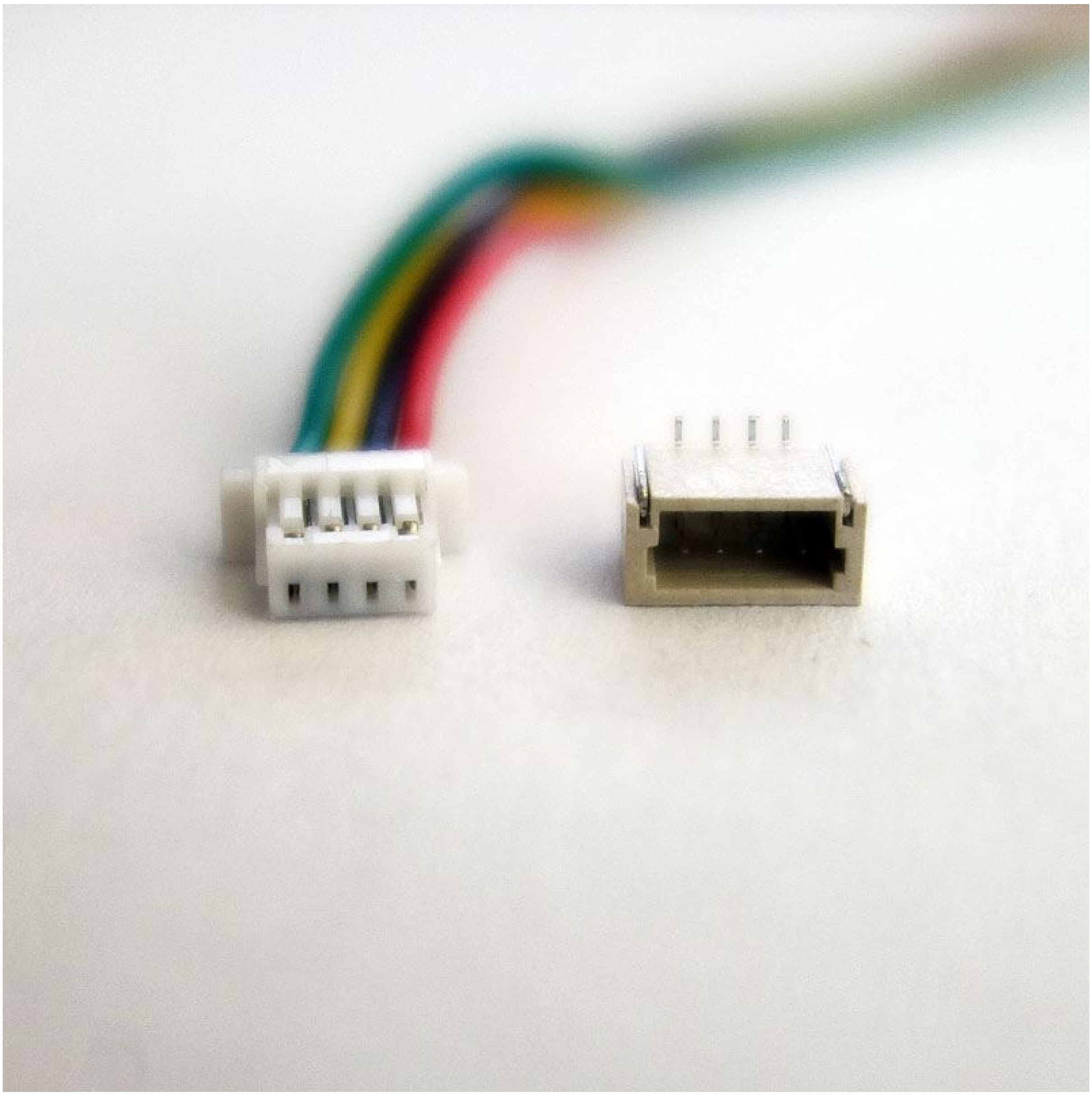
The **bootloader (BL)** is a small piece of software which is started as soon as the board is powered up. Every board ships with a bootloader preloaded and is not normally required to be upgraded by the user.

The **bootloader updater (BU)** is a special firmware which is loaded by the current bootloader and replaces the old bootloader with the new bootloader which it contains. This approach is required because the bootloader can't erase and overwrite itself.

The bootloader version isn't updated after the update.

There is a built in check that prevents the user from updating the bootloader with an incompatible version. If the blue LED blinks rapidly and continuously during the upgrade process, the updater is in an error state. Reboot the board and repeat the process using the correct bootloader updater.

2.2 JST-SH Connectors



2.2.1 Description

JST-SH is a 1.0 mm pitch/ disconnectable crimp style connector used on several of the OpenPilot boards. It is used for it small size and reliable connection. Do not confuse it with JST-ZH or JST-XH that are different connectors and not used in the OpenPilot project.

2.2.2 Uses

The Revolution is using the JST-SH 4-pin for telemetry, GPS, UART and other IO ports.

CopterControl is using the JST-SH 4-pin for telemetry and the FlexiPort. It also uses a JST-SH 8-pin on the ReceiverPort for PWM or PPM input, and optionally for PPM input and servo output.

2.2.3 Datasheets

`SH datasheet`

CHAPTER 3

Indices and tables

- genindex
- search