Qualcomm Technologies, Inc.

PX4 User Guide for Qualcomm® Snapdragon Flight™

80-H9631-2 Rev. K

June 8, 2016

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	September 2015	Initial release
B	October 2015	Update title and content to refer to Snapdragon Flight
C	October 2015	Clarified how to download the PX4 code
D	October 2015	Updated the build instructions for PX4 to include a missing make file variable
E	November 2015	Updated PX4 download, build and execution instructions
F	December 2015	Updated Section 6.2 (Instructions), Section 8.1 (Snapdragon Flight pre-requisites) and Section 10.2 (Airframe setup).
G	January 2016	Updated and refactored tool installation instructions. Updated PX4 source tag.
H	March 2016	Updated procedures in Section 3.3 PX4. Updated PX4 source tag in Section 5.2 Instructions. Removed broken website link in Section 9.4 IMU calibration using QGroundControl. Corrected commands in Section 10.5 Start jMAVSim
I	-	Revision letter “I” was not used per 80-NM730-1
J	May 2016	Numerous changes throughout the document. Several chapters were removed and replaced by references to online documentation.
K	June 2016	Updated the online documentation links. Other minor changes.

Contents

Figures

Tables

1 Overview

The PX4 Autopilot Software is an open-source flight control software. It is ported to run on the Qualcomm® Hexagon™ Digital Signal Processor (DSP) in the Qualcomm Snapdragon System on Chip (SoC), and is used as a reference flight controller on Snapdragon Flight.

More information on the PX4 is available at https://pixhawk.org/start.

This document provides instructions for building and running the PX4 firmware on the Snapdragon Flight.

1.1 Flight modes

The Snapdragon Flight supports running the flight controller not just on a real flying platform, but also on a simulator setup that allows use and test of the platform without installing on a real air frame.

The supported modes are:

- Real flight mode – Allows the PX4 software stack to read real sensor values and control actual motors.
- Hardware in the Loop (HIL) mode – Provides a simulated environment allowing testing of the flight stack by replacing the real physical vehicle and sensors with a simulator of vehicle dynamics and sensor outputs. The flight stack "is not aware" that it is not on a real vehicle. This is a powerful tool for developing and testing code rapidly in a benchtop environment.

2 PX4 High-level HIL Architecture

2.1 What runs where

The flight stack runs anywhere that supports a network connection to the simulator (with sufficient bandwidth and latency to transport the sensor and actuator messages). This includes:

- A standard Linux workstation
- The on-target Linux image
- The on-target Hexagon DSP image. The right platform should be selected based on the goals of the testing. A workstation is useful for rapid testing in a tool-rich environment. Hexagon DSP image testing is the closest to the final implementation, and is useful for testing actual hardware operation.

The Simulator runs on a Windows or a Linux (x86) PC. The Ground Control (QGC) software runs on a Windows PC.

2.2 Architecture

NOTE: This section may be out-of-date.

Figure 2-1 displays a diagram of the PX4 setup.

NOTE: UDP port numbers are displayed only on the socket server and are left blank on the socket client.

NOTE: The references to "#91", "#107", "113" and "#69" in Figure 2-1 refer to the Mavlink message types defined at https://pixhawk.ethz.ch/mavlink/. The ports denoted as "X" are not used.

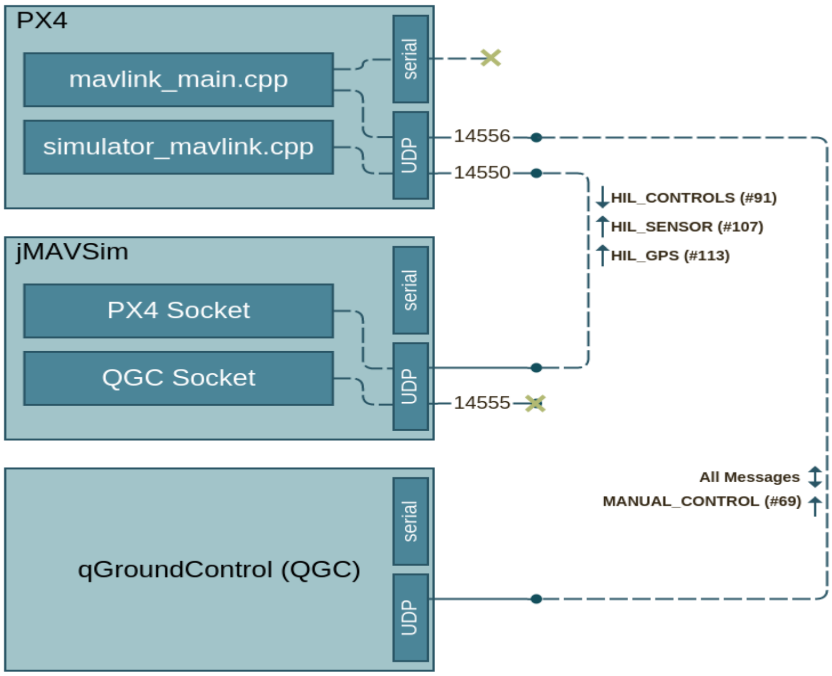


Figure 2-1 PX4 High-level HIL architecture

2.3 Requirements

2.3.1 PX4 build host requirements

- Ubuntu Linux 12.04 or 14.04 (other Ubuntu versions might work, but have not been tested)
- Snapdragon Flight SDK (platform qrlSDK released by Qualcomm Technologies, Inc. (QTI))

2.3.2 HIL mode requirements

- jMAVSim (current supported simulator)
  - The setup described requires PX4 and jMAVSim to be installed and running.
- QGC (supported method of providing manual control commands)

3 Environment Setup

This section provides detailed instructions to build jMAVSim on Linux or Windows, to download and install QGC on Windows, and to download and set up the tools needed to build PX4 software.

3.1 jMAVSim (needed for HIL mode only)

3.1.1 Platform requirements

- Linux/Windows PC with java-1.7.x or greater
- git 1.7 or greater

3.1.2 Windows setup

- Install Java 1.7.x (Java SE Development Kit 7u79) from http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

NOTE: The Java JDK is required, not the "JRE" version. This has been tested only with Java 1.7.

- Install ANT tools from http://ant.apache.org/bindownload.cgi
- See Apache Ant for instructions: http://www.nczonline.net/blog/2012/04/12/how-to-install-apache-ant-on-windows/

3.1.2.1 Environment setup

1. Add environment variables:
   - ANT_HOME: Location where the ANT tools are installed
   - JAVA_HOME: Location of Java installation
2. Add ANT and Java SDK paths to the PATH environment variable

   Example paths for Java and ANT 1.9.6:
   - ANT path: C:\ant\apache-ant-1.9.6\bin
   - Java path: C:\Program Files\Java\jdk1.7.0_79\bin

3.1.3 Linux setup

1. Install jdk1.7 on the Linux host:
   ```
   1 > sudo apt-get install openjdk-7-jdk
   ```
2. Install ant (if not already installed).
3. Add the following environment variable:
   ```
   1 > export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
   ```

3.1.4 Build instructions

In a clean directory, run:

```
1 > git clone https://github.com/mcharleb/jMAVSim.git
2 > cd jMAVSim
3 > git submodule init
4 > git submodule update
5 > ant
```

3.2 QGroundControl (QGC)

3.2.1 Platform requirements

- Windows 7 PC
- Logitech Gamepad F310 joystick controller OR
- Spektrum Transmitter with AeroSIM RC dongle (recommended)

3.2.2 Download/install instructions

Download QGC from http://qgroundcontrol.org/downloads and install QGC using the Windows executable; v2.7.1 is recommended.

3.3 Snapdragon Flight prerequisites

Install the platform software on target. This needs to be done only once.

3.4 PX4

3.4.1 Target platform requirements

Linux on the Snapdragon Flight with a working IP interface.

3.4.2 Build host PC requirements

- Ubuntu Linux 12.04 or 14.04.
- QTI Snapdragon Flight SDK (qrlSDk)

3.4.3 Download and build prerequisites

1. Install the qrlSDK package that is part of the Snapdragon Flight platform.
   a. Obtain the qrlSDK.tgz file. Contact your customer support team for details.
   b. Untar the file at the desired location, for example: `/opt/tmp`.
   c. Install the file at the desired location, for example: `/opt/qrlSDK`.

```
> mkdir /opt/tmp
> cd /opt/tmp
> tar zxf <top of build tree>/oe-core/build/
  tmp-eglibc/deploy/sdk/qrlSDK.tgz
> ls README
> ./qrlSDKInstaller.sh -d /opt/qrlSDK
```

2. Follow the instructions on this page to install clang, gcc, cmake, Hexagon SDK and Hexagon tools, and set up the environment variables:
   https://github.com/ATLFlight/ATLFlightDocs/blob/master/GettingStarted.md
3. Install Python 2.7, if it is not already installed.
4. Install the required Debian packages:

```
sudo apt-get install python-pip python-empy
```

5. Install the required python package.

```
sudo pip install catkin_pkg
```

4 PX4 Download and Build Instructions

Follow the instructions at https://github.com/ATLFlight/ATLFlightDocs/blob/master/PX4.md to build and install the binaries and configuration files for flight and HIL modes.

5 PX4 in Real Flight Mode

Follow the instructions at https://github.com/ATLFlight/ATLFlightDocs/blob/master/PX4.md to build and install the binaries and configuration files and to start the flight software on the vehicle in flight mode.

5.1 Using QGroundControl

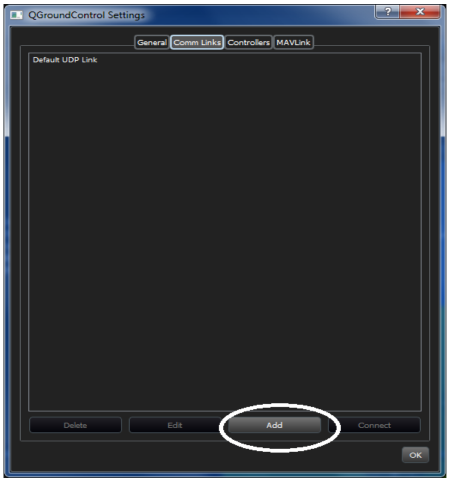
NOTE: This procedure and images in this chapter apply to QGroundControl 2.7.1. The UI and settings may appear different in later versions.

5.1.1 Start QGroundControl

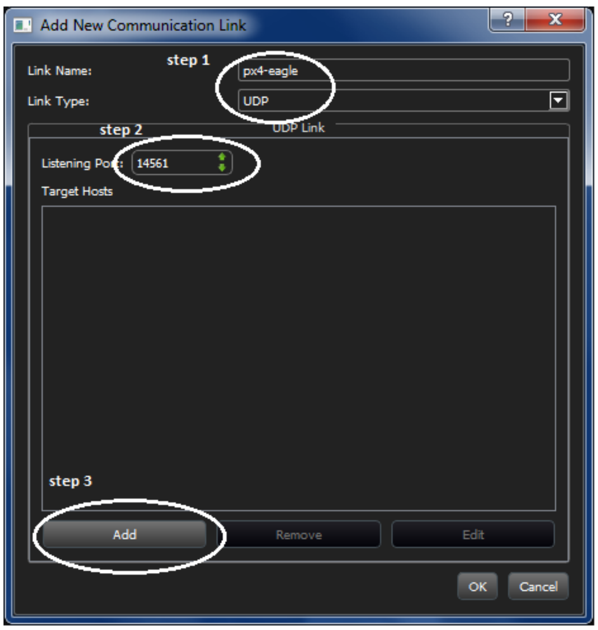
1. Launch the QGroundControl (QGC) application.

2. Set up the communication to Snapdragon Flight:

   a. Select **File** > **Settings** > **CommLinks**.



b. Click **Add.**



c. Enter a **Link Name** of your choice.

d. Under Link Type, select **UDP.**

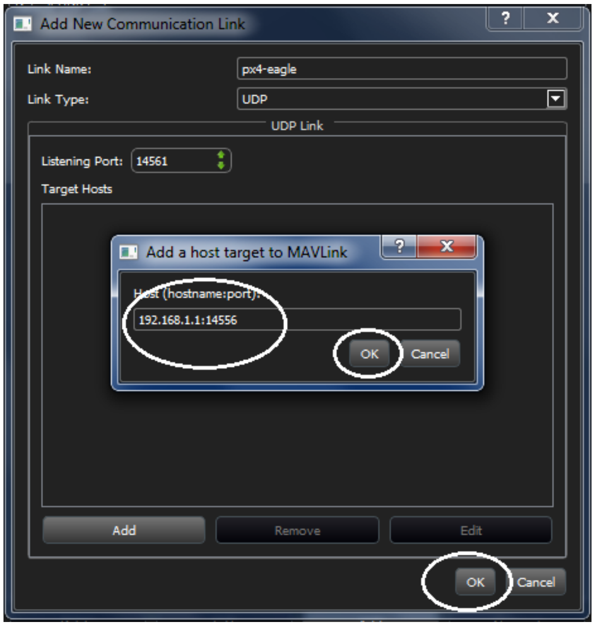
e. Set the listening port to an unused port, for example, 14561.

3. Click **Add**.

   a. Enter the **IP address and port** of the PX4 Mavlink app: <IPADDR>:14556

      - Where <IPADDR> = IP address of the Snapdragon Flight board

   b. Click **OK**.



5.1.2 Joystick setup

- Spektrum Controller (DX6i) with AeroSIM RC dongle (recommended).

NOTE: This procedure may vary based on the profile settings of the Spektrum transmitter. Refer to the Spektrum documentation for managing/setting up user profiles.

NOTE: If using the Spektrum Transmitter, make sure that it is not bound to the Spektrum Receiver on the UAV. **Not doing so sometimes produces deterministic results**.

1. Obtain the AeroSIM RC dongle: http://www.aerosimrc.com/en/home.htm
2. Connect the AeroSIM RC USB to the host that will run QGC.
3. Connect the other end of the USB cable into the trainer port of the Spektrum transmitter.
4. Start QGC.
5. Connect to the UAV based on the configuration done in Start QGroundControl.
6. Navigate to File > Settings > Controllers.

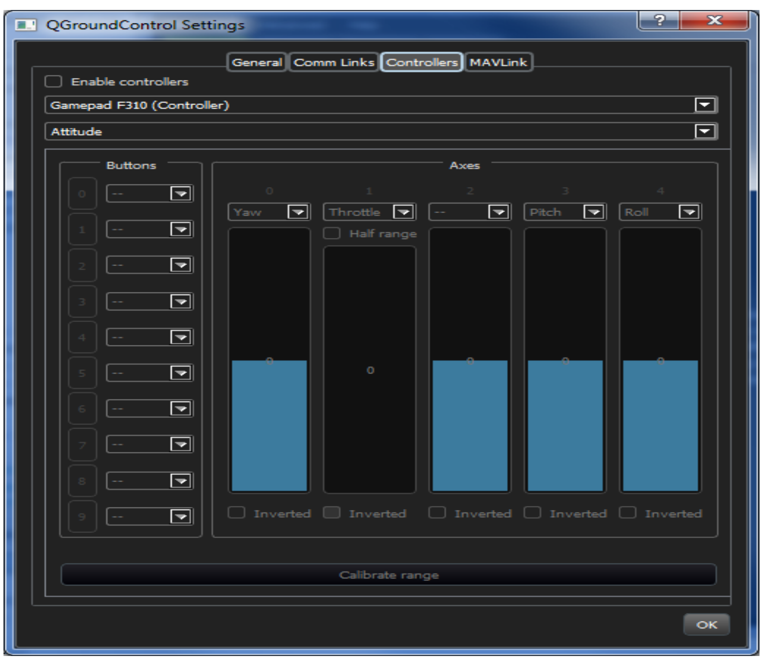
7. Select the box next to **Enable controllers**.
8. Select **AeroSIM RC v2**.
9. Select **Manual**.
10. Set the axes/channel mapping to:
    - 0 > Throttle
    - 1 > Roll
    - 2 > --
    - 3 > Yaw
    - 4 > Pitch
    - 5 > --
11. Check **Inverted** for the Roll axis.
12. Check **Inverted** for the Yaw axis.
13. Check **Inverted** for the Pitch axis.

14. Click **Calibrate range**.
15. Move the right joystick through its full range of motion (in a complete circle).
16. Move the left joystick through its full range of motion (in a complete circle).
17. Click **End calibration**.

18. Click **OK**.

5.1.3 Logitech Gamepad F310 joystick controller

1. Connect the Logitech Gamepad F310 joystick to the host that runs QGC.
2. Start QGC.
3. Navigate to File > Settings > Controllers.

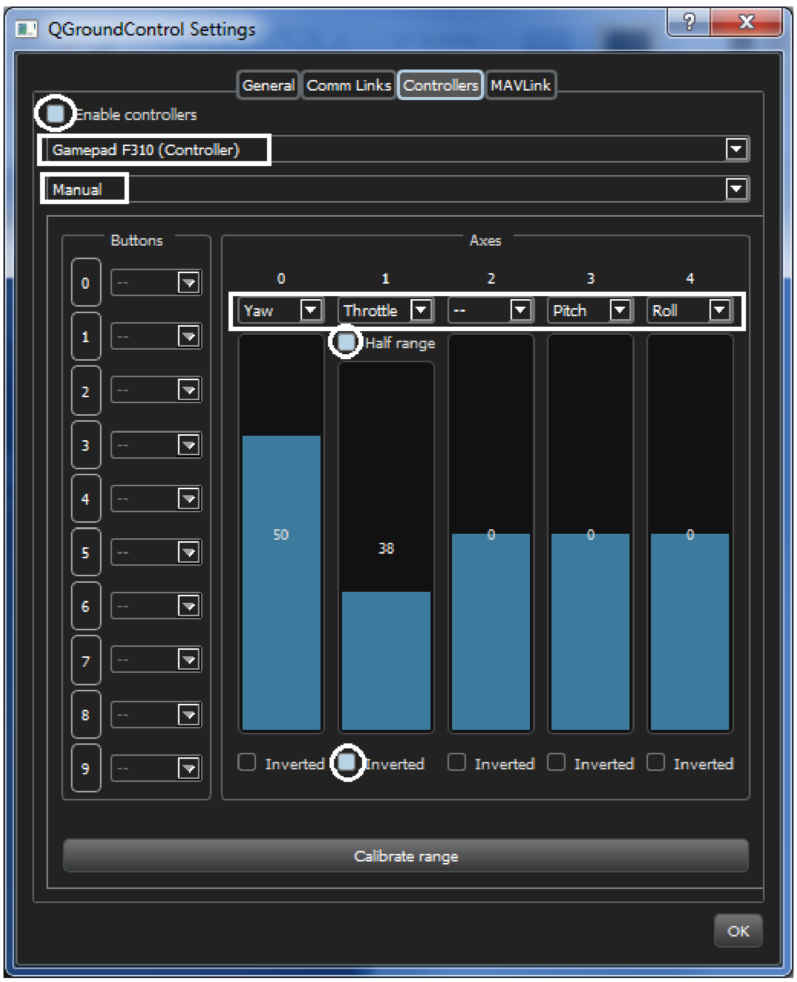


4. Check the box next to **Enable controllers**.
5. Select **Gamepad 310**.
6. Select **Manual**.
7. Set the axes/channel mapping to:
   - 0:Yaw
   - 1 > Throttle
   - 2 > --
   - 3 > Pitch
   - 4 > Roll

   a. Check **Inverted** for the throttle axis.

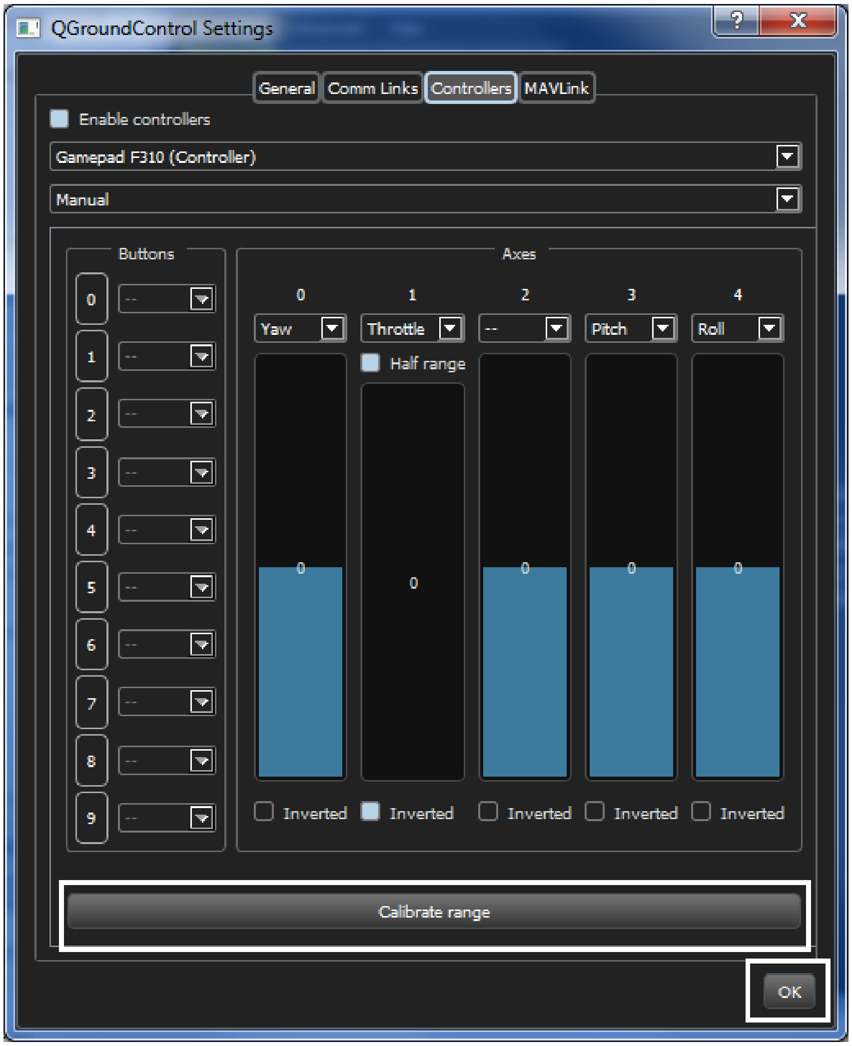
   b. Check **Inverted** for the pitch axis.

   c. Check **Half range** for the throttle axis.



8. Click **Calibrate range**.
9. Move the right joystick through its full range of motion (in a complete circle).

10. Move the left joystick full left, then full right.
11. Move the left joystick full forward (but not full backward).
12. Click **End calibration**.
13. Click **OK**.



5.1.4 Other controllers

To configure other transmitters (assuming the driver for the transmitter is installed properly):

1. Connect the transmitter and start the QGC application.
2. Click **Connect** to connect the UAV.
3. Navigate to File > Settings > Controllers.
4. Click the box next to **Enable controllers**.
5. Select **the transmitter**.
6. Select **Manual**.
7. Set the **axes/channel mapping**.

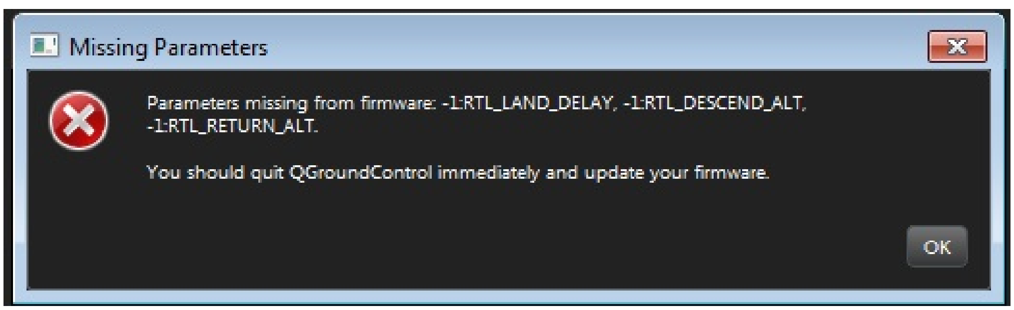
8. Click **Calibrate ranges**.
9. Move the throttle the full range (top and bottom).
   a. Move the roll the full range (left to right).
   b. Move the pitch the full range (top to bottom).
   c. Move the yaw the full range (left to right).
   d. Click **OK**.
10. Label the column that changes when the throttle is moved to **throttle**. Repeat the same for roll, pitch, and yaw.
11. Check for correct polarity
    - Throttle: Up is max, bottom is low.
    - Roll: Left is negative, right is positive.
    - Pitch: Bottom is positive, top is negative.
    - Yaw: Left is negative, right is positive.

    If the polarities do not match as specified, invert the appropriate channels, and click **OK**.
12. Click **Connect**, which is located at the upper right corner, to connect the flight stack.
    a. Select the connection created in step 1.

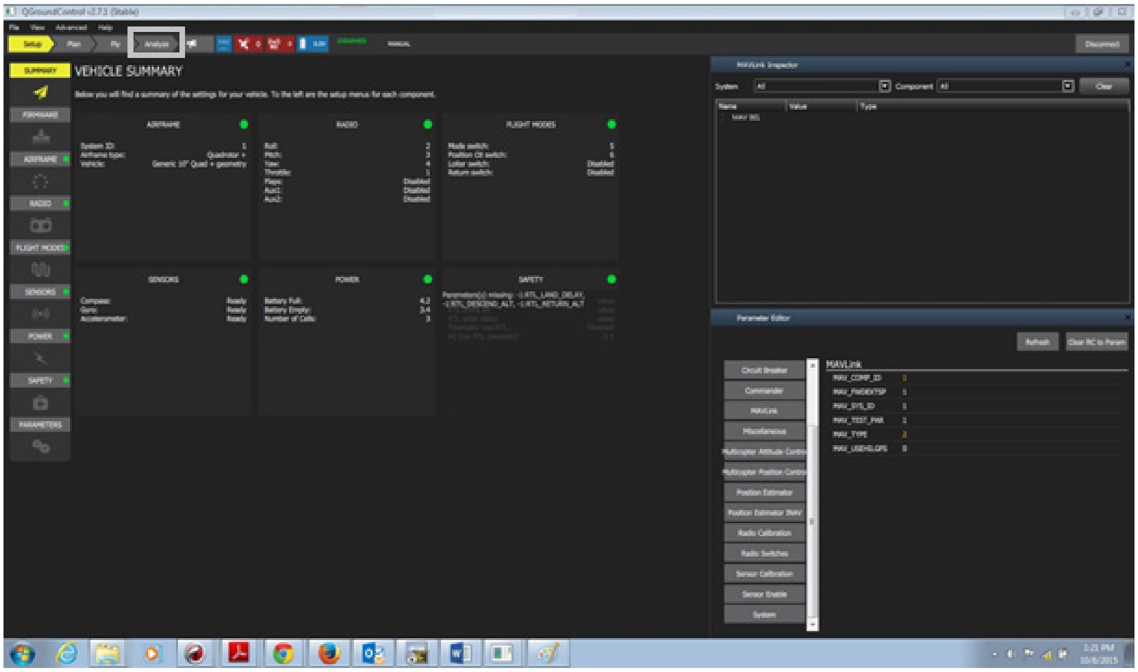
    The following error appears when connecting to Snapdragon Flight. These errors are normal.

    b. Click **OK** for all.

    An error message, like the one below, might appear when Snapdragon Flight is connected. Click **OK**.



13. After the error message closes, the page displays. Click **Analyze** in the top left corner.



14. The Analyze page displays a final snapshot after a connection is made to Snapdragon Flight.

When you are connected to the flight stack, you can see incoming Mavlink packets using the MAVLink Inspector (from Advanced > Tool Widgets). The recommended widgets are:

- MAVlink Inspector
- Primary Flight Display
- Parameter Editor

5.2 Airframe setup

NOTE: If you use the Microheli 200 QX configuration files as described in this webpage, you may skip this section.

Complete this procedure only once.

1. Create the mainapp.config and p4.config files.
2. Start the PX4 flight software on target.
3. Open QGroundControl and connect to the target according to Section 5.1.1.
4. Click **Setup** > **AIRFRAME**. Specify your airframe type and apply the changes.
5. The flight software on the target now exits and QGroundControl disconnects.
6. Restart the flight software on target
7. Reconnect QGroundControl to the target.

Steps 1, 2, and 6 are described here:
https://github.com/ATLFlight/ATLFlightDocs/blob/master/PX4.md

5.3 Set sensor board rotation parameter

Depending on your hardware platform and how it is oriented on your airframe, you might need to update the **SENS_BOARD_ROT** parameter prior to performing IMU calibration or an actual flight. For more information, see https://pixhawk.org/sensor_orientation.

The procedure is as follows:

1. Select the **Setup** > **Sensors** tab.
2. Open Advanced > Tool Widgets > Parameter Editor.
3. Click the **Sensor Calibration** group.
4. Click the **SENS_BOARD_ROT** entry.
5. Edit the parameter value.
   - In the Qualcomm Snapdragon Flight reference design, the value is 0 for the new Snapdragon Flight P2 target platform.
   - If your design uses a different orientation between the board / IMU sensor and the airframe, see https://pixhawk.org/sensor_orientation for more information on computing this and the related parameters.
6. Click **Save**.
7. Verify that the updated value is correct by physically moving the airframe (pitch, roll and yaw motions) and observing the flight display panel.

NOTE: If you use the Microheli 200 QX configuration files as described in this webpage, you must manually update the following line in the px4-calib.config and px4-flight.config files:

```
param set SENS_BOARD_ROT <n>
```

where <n> is the value from Step 5 above.

5.4 IMU calibration using QGroundControl

Before a real flight, the following IMU calibration procedure must be performed at least once to calibrate the gyroscope, accelerometer, and magnetometer sensors. The setup and calibration procedures are illustrated in this video: https://youtu.be/91VGmdSlbo4. Also refer to this page http://qgroundcontrol.org/setup_view from step 4 onward.

1. Set the sensor board rotation parameter as described in Section 5.2.
2. Select the **Setup** > **Sensors** tab.
3. Calibrate the gyro, accelerometer and magnetometer by clicking the buttons for each of these sensors and following the on-screen prompts and visualizations to position the drone.
   a. Start gyroscope calibration (drone is on a flat surface at rest).
   b. Start accelerometer calibration (drone is oriented along each axis per onscreen instructions and visualizations).
   c. Start magnetometer calibration (drone is oriented along different axes and rotated per onscreen instructions and visualizations).
4. Create the mainapp.config and px4.config files for real flight according to this webpage.
5. For the parameters to take effect, reboot the board and reconnect QGroundControl.

5.5 PID tuning using QGroundControl

NOTE: If you use the Microheli 200 QX configuration files as described in this webpage, you may skip this section.

Use the following guide to tune the flight controller PID parameters for your platform: https://pixhawk.org/users/multirotor_pid_tuning

Refer to the following page for instructions on how to set PID (and other) parameters during the tuning process: https://pixhawk.org/users/parameter_guide

Table 5-1 lists the defaults for the Microheli 200QX.

Table 5-1 Defaults for the Microheli 200QX

	P	RATE_P	RATE_D
MC_YAW	8	0.12	0.001
MC_PITCH	7	0.08	0.001
MC_ROLL	7	0.08	0.001

5.6 Controlling PX4 flight

Once the system has powered up, the transmitter joysticks and switches may be used to control and operate the UAV (including arm, disarm, pitch, yaw, roll, and altitude). QGC can display instruments such as artificial horizon and maps.

For details on operating the UAV using the transmitter in various modes of operation, see https://pixhawk.org/peripherals/radio-control/start and http://px4.io/docs/px4-autopilot/flying.

6 PX4 in HIL Mode

Follow the instructions at https://github.com/ATLFlight/ATLFlightDocs/blob/master/PX4.md to build and install the binaries and configuration files and to start the flight software on the vehicle in HIL mode.

6.1 Start QGC

See Section 5.1 for the procedure to set up and start QGroundControl.

6.2 Start jMAVSim

NOTE: In the directory where jMAVSim is installed, run the following command:

Windows command (recommended)

```
java -cp lib\*;out\\production\\jmavsim.jar me.drton.jmavsim.Simulator -udp
<IPADDR>:14560
```

Linux command:

```
java -cp lib\*:out/production/jmavsim.jar me.drton.jmavsim.Simulator -udp
<IPADDR>:14560
```

1. Replace `\<IPADDR\>` with the IP address of the machine/target running PX4 (Snapdragon Flight). Run **ifconfig** on the same machine/target to obtain the IP address.

6.3 Control PX4 flight in HIL mode

The joystick is now ready to fly the simulated vehicle. Figure 6-1 displays the jMAVSim world visualization with an FPV view. QGC can display instruments such as artificial horizon and maps if GPS simulation is enabled.

NOTE: Controller arming/disarming is required in HIL mode, and the QGC displays the current arming status (as shown in Section 5.1.4, Step 13).

Figure 6-1 jMAVSim world visualization

7 Debugging and Frequently Asked Questions

7.1 QGC issues

QGC is a third-party tool. Issues that we have seen are as follows:

- Reconnecting to the Snapdragon Flight causes the Controllers to be disabled.
- Reconnecting to the Snapdragon Flight sometimes reverses the control to **Attitude**. This should be reset to **Manual**.
- Controller mappings are lost. Controller settings and calibration needs to be reestablished, per the **Starting QGroundControl** instructions in Section 5.1.1.
- The symptom of this issue is that one cannot control the PX4 stack using the joysticks.

NOTE: Refer to the QGC tool documentation, located at http://qgroundcontrol.org/users/start for more information.

7.2 Updating the parameters for PX4 running on adsp

To change the startup commands for asdp, update the px4.config file located at /usr/share/data/adsp to make the necessary changes.

NOTE: The aDSP flight software always looks for the px4.config file under /usr/share/data/adsp folder. **Using any other name will not work.**

7.3 Controlling UAV using QGC and Spektrum

1. If using the Spektrum transmitter to control the UAV over WiFi via QGC, ensure that it is not bound to the receiver on the UAV.
   - If this is done and the transmitter is turned on, the UAV will receive commands from both the transmitter and the QGC when the transmitter is connected to QGC.
2. Follow this order to configure/calibrate the Spektrum transmitter using QGC:
   a. Connect the Spektrum transmitter to a laptop running QGC via the AeroSIM RC dongle.
   b. Start QGC.
   c. Connect to UAV.
   d. Navigate to File > Settings > Controller to calibrate the joysticks.

3. When using the Spektrum transmitter, make sure it is in the **Off** position before starting QGC.
4. Make sure you *do not* turn the transmitter on and then off after it is connected to QGC via the dongle. Doing so allows the RC commands to go over the RC link as well as via the AeroSim dongle.

7.4 Control mixing

The software supports the standard PX4 control mixing feature as described at: https://pixhawk.org/dev/mixing.

A mixer configuration file named "mixer_config.txt" must be created per the instructions in the above webpage and placed in /usr/share/data/adsp on target. If the file is not present, the default settings work for most quadcopter X airframes.

NOTE:

1. Alternative airframe geometries such as 'v', 'Y', etc., have not yet been tested.
2. Currently only 4-rotor designs are supported (hard-coded in software). Designs with different numbers of rotors are not yet supported.