Gumstix® AeroCore 2 Series User Guide

Revised March 21, 2018

www.gumstix.com

Redwood City, CA • San Jose, CA • Vancouver, BC

Gumstix® AeroCore 2 Series
User Guide

Keith Lee
Gumstix, Inc.

March 21, 2018

Contents

1 Introduction

Micro Air Vehicles **(MAVs)** have become increasingly popular both commercially and in industry. They are used for surveying, videography, surveillance, entertainment, and more. These compact drones are deceptively complex and, without advanced flight control software, they would be impossible to pilot from the ground. For example, in the case of multi-rotor copters, a high-speed feedback loop is vital to maintaining stable flight, adjusting rotors' rate of rotation to keep the drone level. A user on the ground could not manipulate a Radio Controlled **(R/C)**

The AeroCore 2 series of expansion boards from Gumstix® offer a complete solution for MAVs, with an on-board PX4 control subsystem for R/C and stay-in-the-sky functionality, plus the processing power of an attached compute module or Single Board Computer **(SBC)**.

There are currently AeroCore 2 expansion boards for the following compute devices:

- Gumstix® Overo™
- Gumstix® DuoVero ™
- DragonBoard 410C[1]
- Intel® Joule™ module

The benefit of the AeroCore 2 design over most MAV controllers is that the NuttX Real-Time Operating System **(RTOS)**, running on an on-board MicroController Unit **(MCU)**, controls the drone completely independently of the compute device, yet they can operate in concert through an on-board communications bridge. This allows for high-level feedback systems, such as obstacle avoidance or swarm behavior algorithms without compromising the response time of the core flight software.

2 Board Features

Figure 1 illustrates the hardware common to all AeroCore 2 boards. These components are generally associated with the PX4 flight control subsystem. The positioning of these elements can also vary from board to board. The designs also include Universal Asynchronous Reciever/Transmitter **(UART)** and Serial Peripheral Interface **(SPI)** communication channels between the PX4 system and the connected compute device.

Here is a full list of common features:

- 16-36V battery connector
- ARM Cortex-M4 MCU
- 9-axis Internal Measurement Unit **(IMU)**
  - 3-axis accelerometer
  - 3-axis gyroscope
  - 3-axis magnetometer
- Barometer
- 3 analog input pins
- 8 Pulse Width Modulation **(PWM)** Output pins

[1] Compliant with the 96Boards Consumer Edition specification

- 40-pin General Purpose Input/Output **(GPIO)** header
- Pre-GO Global Positioning System **(GPS)** module connector
- Digital Spectrum Modulation **(DSM)** antenna connector
- Inter-Integrated Circuit **(I^2C)**, UART, SPI, and Controller Aera Network **(CAN)** serial bus headers

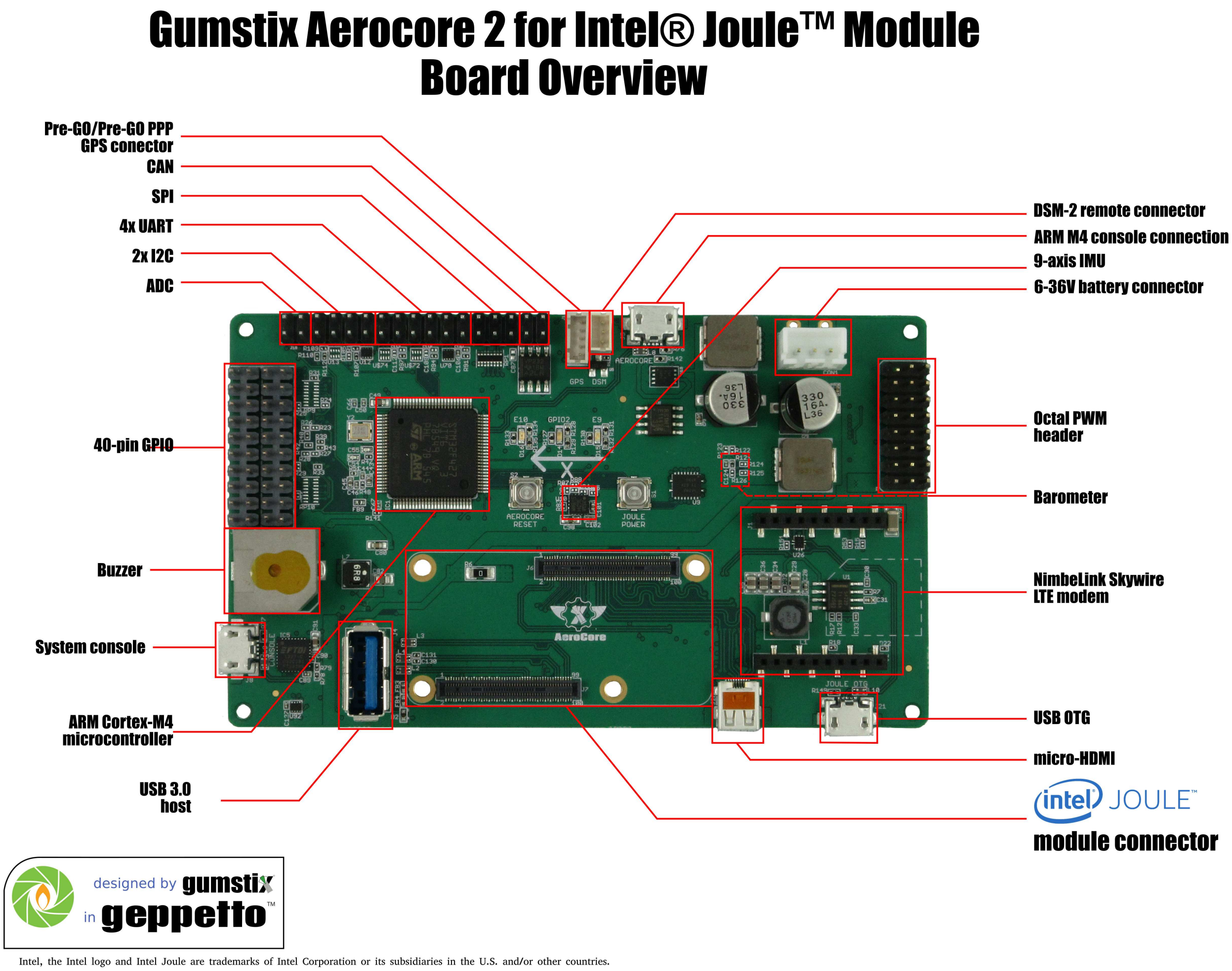

Figure 1: Common hardware components of AeroCore 2 expansion boards

AeroCore 2 boards also vary slightly depending on the Computer on Module **(COM)** or compute device they support. The various differences are listed in Table 1.

Overo	DuoVero	DragonBoard 410C	Intel® Joule™ module
Right-angle USB-A 2.0 port	Right-angle USB-A 2.0 port		Right-angle USB-A 3.0 port
		NimbeLink Skywire LTE modem connector	NimbeLink Skywire LTE modem connector
		15-pin MIPI CSI-2 camera connector	microHDMI port
Overo COM connector	DuoVero COM connector	96Boards CE Mezzanine connector	Intel® Joule™ module connector

Table 1: Board Differences by COM Model

3 AeroCore 2 Setup

The AeroCore 2 series of expansion board is easily integrated into an MAV. Gumstix®'s custom PX4 firmware is flashed onto the ARM Cortex-M4's on-board memory, the vehicle's hardware is wired to the board and the sensors and motors are calibrated. Below are instructions, examples and guidelines for getting started with the AeroCore 2.

3.1 Flashing AeroCore

The AeroCore 2 firmware provides the NuttX real-time operating system and PX4 software to the ARM Cortex-M4 microcontroller on the AeroCore 2 . There are two methods for flashing the PX4 firmware: via QGroundControl **(QGC)** and as part of the compilation process. Both are explained below

3.1.1 Flashing Firmware Via QGroundControl

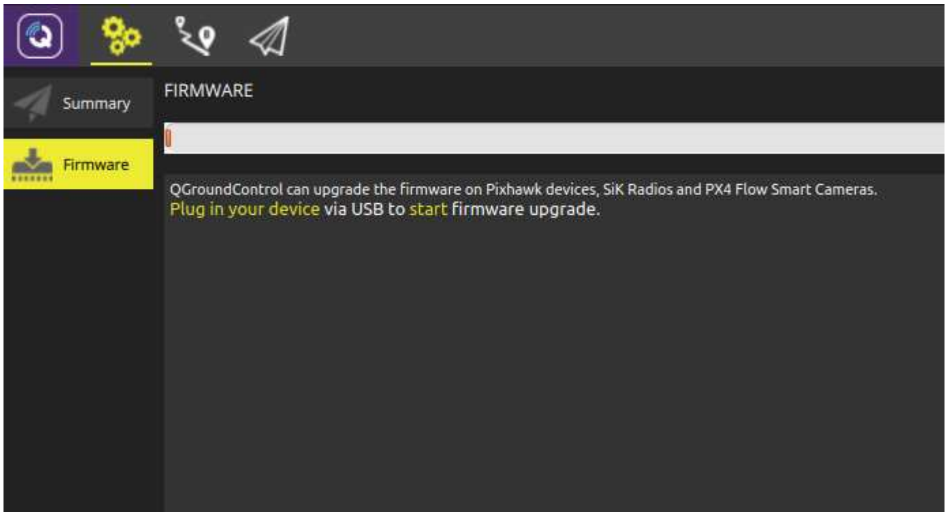

Figure 2: QGroundControl Firmware Flasher Screen

Built in to QGC is a firmware flashing utility that will download and install the software specific to your device. Several models of control systems are supported directly and the ability to select a local file is also available. This is the most straightforward method of updating the firmware on your AeroCore 2 and is recommended for most users. The following is a step-by-step guide.

1. Download and install QGC from the website. It is available for most platforms.

- The binaries are available from:

  http://qgroundcontrol.com/downloads/

- the software can also be built from source:

  https://github.com/mavlink/qgroundcontrol.git

2. Start the application and click on the settings button on the toolbar
3. In the left-hand column, click the firmware button
4. The main window instructs you to **'Plug in your device via Universal Serial Bus (USB) to start firmware upgrade.'** Plug a USB type-A to micro-B cable into the port labeled **'aerocore'** or **'stm console'** on the AeroCore 2 board and an available USB port on the host PC. (See Fig. 2)
5. If the software detects the board correctly, the main window will be greyed and a column will appear on the right-hand side. ensure the **'PX4 Flight Stack'** radio is selected and choose from the options below:
   - To download and install the default firmware image from the internet, click **'Ok'**
   - If a custom image is to be used, check the **'Advanced settings'** box, select **'Custom firmware file...'** from the drop-down menu, click **'Ok'** and select the custom **.px4** file in the pop-up dialog
6. QGC will then erase the existing firmware, if present, and upload the new image on to the AeroCore 2.

3.1.2 Compile-Time Flashing

More advanced users who wish to compile the firmware from source can flash their boards using scripts built in to the code repository. Section 3.2 has detailed instructions on how to set up your development environment and compile the code. Once the firmware has been generated, uploading it to the AeroCore 2 is a one-step process.

The Makefile for the PX4 firmware includes a recipe for uploading the newly built software directly onto the MAV controller. Simply add `upload` to the make command and the script will attempt to upload to a USB-connected board.
Eg:

```
$  make aerocore2_default upload
```

This may involve removing and re-inserting the USB cable.

3.2 Building Firmware

Gumstix® has customized the *MAVLink* PX4 software suite to the MAV's standard features and the source code is available online from GitHub at https://github.com/aerocore/Firmware. Compiling the firmware from source is not required as it is provided by the QGC flashing utility.

Instructions for building the AeroCore 2 PX4 firmware are provided below:

1. Add dialout permissions to your user

```
$ sudo usermod -a -G dialout $USER
```

2. Remove *Modem Manager*

```
$ sudo apt-get remove modemmanager
```

3. Install dependencies

```
$ sudo apt-get install python3-serial openocd \
  flex bison libncurses5-dev autoconf texinfo build-essential \
  libftdi-dev libtool zlib1g-dev \
  python-empy -y
```

4. Clone the repository and select a branch: master, stable, or beta

```
$ git clone https://github.com/aerocore/Firmware.git aerocore2_firmware

$ cd aerocore2_firmware && git checkout aerocore2-⟨master OR stable OR beta⟩
```

5. Ensure all submodule repositories are up-to-date

```
$ git submodule update --init --recursive
```

6. Compile

```
$ make aerocore2_default
```

3.2.1 Flashing the Bootloader

It is rarely necessary to flash the AeroCore 2 bootloader as it is shipped pre-flashed. However, if you are having difficulty flashing the firmware to your board, re-flashing the bootloader may resolve the underlying issue.

To install the bootloader you will need dfu-util and the boot-loader binaries from Gumstix®.

- You can download the bootloader at:

  http://gumstix-misc.s3.amazonaws.com/aerocore2/px4aerocore_bl.bin

- Optionally, you can download and compile it from:

  https://github.com/aerocore/Bootloader/tree/aerocore2

Enter these commands to build the bootloader from source:

```
$ git clone https://github.com/libopencm3/libopencm3.git

$ cd libopencm3 && make

$ cd ..  && git clone https://github.com/aerocore/Bootloader.git

$ cd Bootloader && make
```

On a Ubuntu development machine (Windows and Mac OS users check dfu-util documentation):

1. Install `dfu-util`

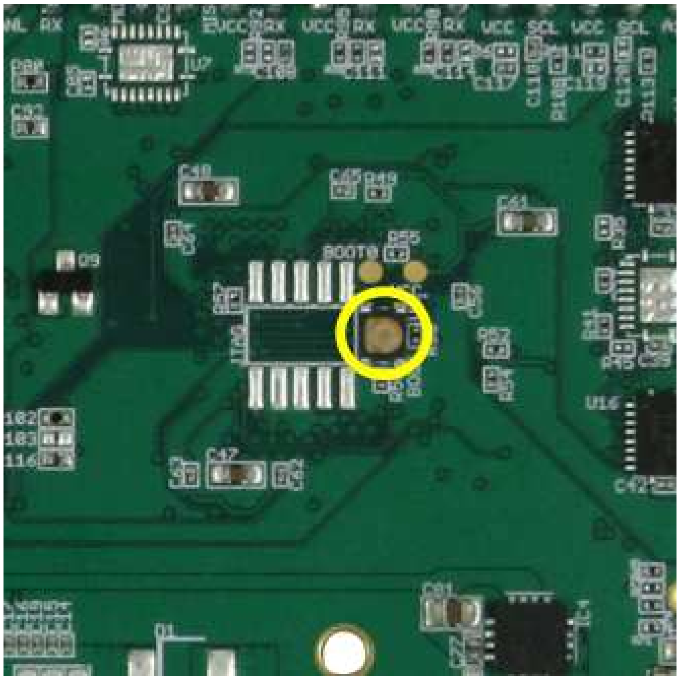

Figure 3: Location of **BOOT0**

```
$  sudo apt-get install dfu-util
```

2. Download the latest bootloader binary

```
$  wget http://gumstix-aerocore.s3.amazonaws.com/Bootloader/px4aerocore_bl.bin
```

3. locate the **BOOT0** button on the back of the AeroCore 2, circled in yellow in Fig 3
4. While holding down **BOOT0**, plug the microUSB connector into the USB port labled **stm_console** or **aerocore**. Then release **BOOT0**.
5. Flash the downloaded bootloader binary to the AeroCore 2

```
$  sudo dfu-util -a 0 -D px4aerocore_bl.bin -s 0x08000000
```

6. Press the **AeroCore 2 reset** button. There should be a solid blue Light Emitting Diode **(LED)** and flashing yellow LED

3.3 Hardware Configuration: Quadcopter Example

Methods for connecting and calibrating the AeroCore 2 will vary depending on the MAV to which it is connected. A popular example is the X-frame quadrotor. For this example, the **DJI Flamewheel F450** was used. Figure 4 shows the constructed drone using the AeroCore 2 for DragonBoard 410C.

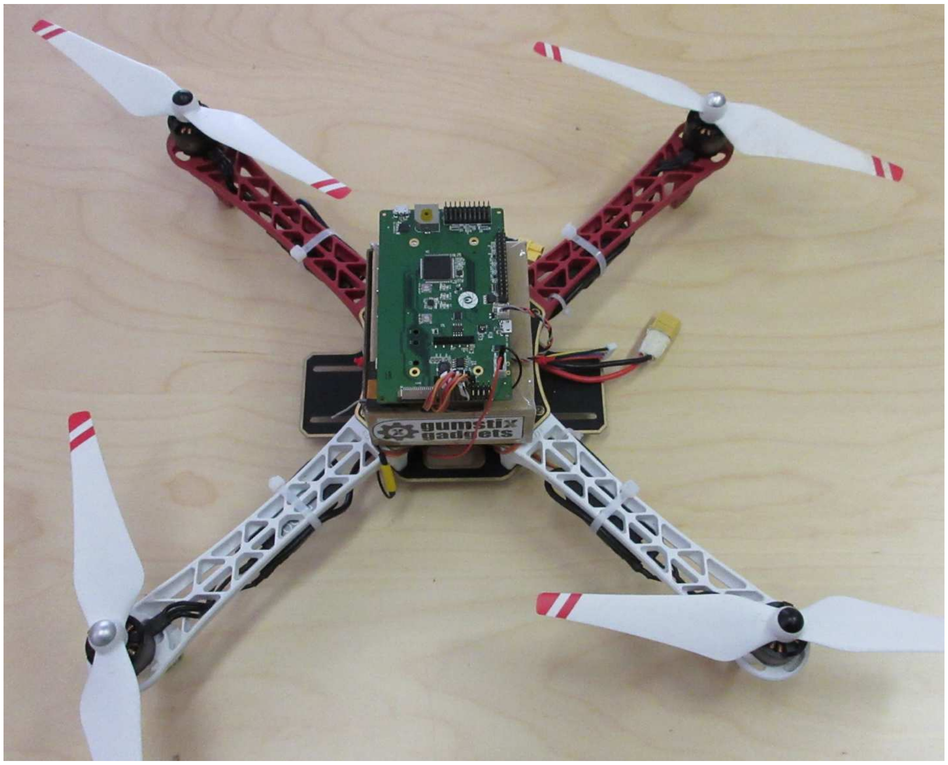

Figure 4: AeroCore 2-based Quad-rotor MAV

Prior to connecting the MAV's electronics to the AeroCore 2, secure it to its compute device and to the drone's chassis. Dampening vibration from the MAV chassis is recommended to maximize sensor data accuracy.

First, power must be provided from an on-board battery to the AeroCore 2's battery connector. This can be done by soldering jumper wires to V_{CC} $(+)$ and GND $(-)$ pads on the undercarriage of the frame and connected to the 3-pin battery connector. Alternately, An adapter for the battery's balance connector could be made to attach battery to board.

Next, the input wires for the Electronic Speed Controllers **(ESCs)** must be connected to the AeroCore 2's PWM header. The PX4 firmware requires that the PWMs be connected in a specific order. Figure 5 below illustrates that order.

The DSM radio receiver must be synced with a transmitter/controller. The steps involved in this process will differ dependent on manufacturer and model of both the receiver and controller. Please consult the user manuals for your hardware to complete this step. Once they are synced the receiver antenna can be connected to the AeroCore 2's 4-pin DSM connector.

At this point, the Quadcopter is flyable with a DSM radio controller. In order to automate flight plans, GPS data is required. All AeroCore 2 boards include a 5-pin header for connecting Gumstix Pre-GO GPS modules for this purpose. Connecting the module to the header and mounting it to the frame with the antenna exposed and pointed upward provides real-time GPS data to the *MAVLink* software. When the module has obtained a GPS lock, a LED will blink on the Pre-GO, and the AeroCore 2's buzzer will go silent.

Additional devices and sensors can be connected to the board via USB, GPIO, I^2C, SPI, UART and CAN connections. Refer to Figure 1 for a callout of all board connections.

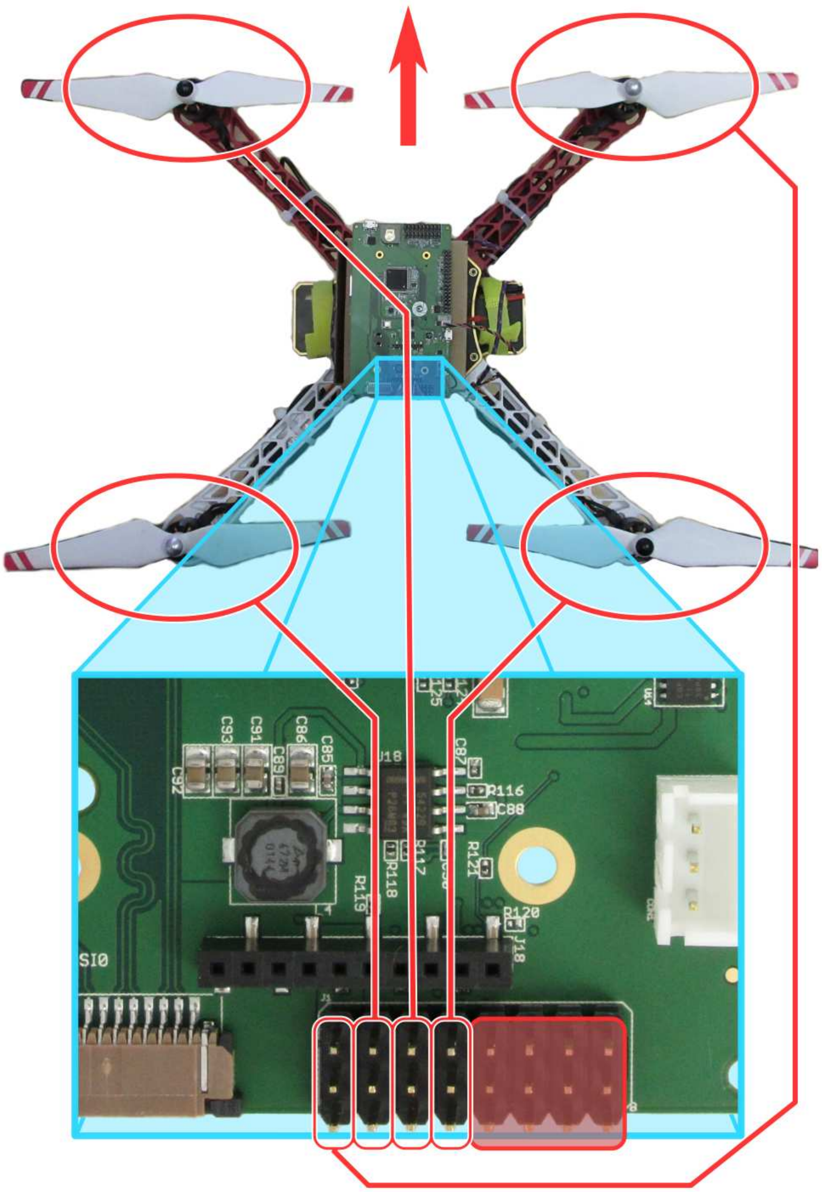

Figure 5: AeroCore 2 PWM Configuration

3.3.1 Calibration

The simplest method of hardware calibration is to use the setup tools provided in QGC. It uses a familar "Green Means Go" model by which users manipulate the drone and its controls as instructed. With the MAV connected as in the firmware upload procedure and the QGC application running, the summary screen on the **"Setup"** tab displays a series of boxes

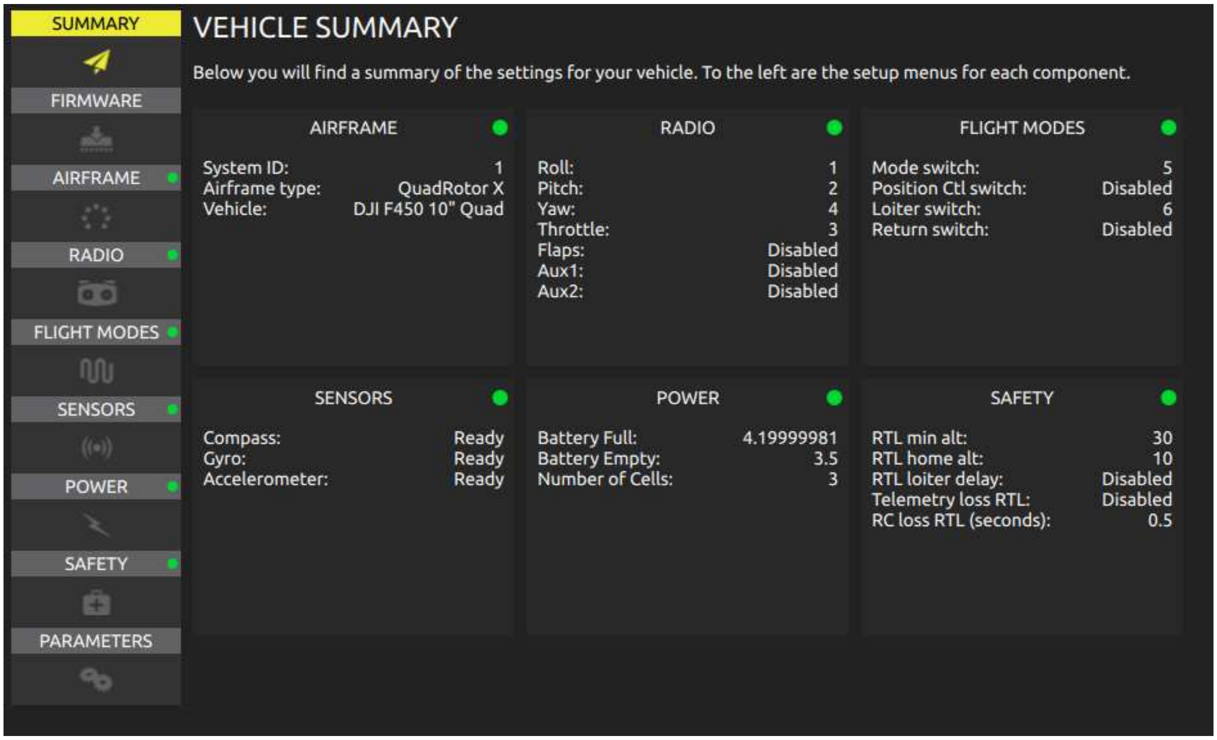

Figure 6: The QGroundControl Vehicle Summary Screen

with coloured indicators in the top right corners. This screen is shown in Figure 6. When all of these indicators are green, then the MAV, its radio controller, and its sensors are all set up correctly.

Either by clicking on a box's title bar or on the associated button in the UI's left-hand margin, users will find clear instructions on configuring and calibrating the various attributes specific to the MAV platform, including airframe configuration, joystick range of motion, gyro and accelerometer attributes, and battery power characteristics. The drone's safety and "return home" procedures – delay, transit altitude, speed and rate of descent – can also be defined within this interface.

3.3.2 Note: Design Variations

Assembly and calibration will differ dependent on the airframe, R/C controller design, user preferences and project specifications. Take the time to verify that all settings are configured correctly and all physical connections are correct to your platform and secure before attempting flight.

Miscalibrated sensors or controls, miswiring or disconnection of ESCs, or inappropriate failsafe settings can cause damage and serious injury.

4 R/C Operation Example: DSM radio control with Live USB Webcam and Telemetry Feed

For this example, The AeroCore 2's connected compute device will be used as a pipeline for a H.264 video feed and a proxy for PX4 data transmission. These instructions assume the following:

- A constructed, flashed, and calibrated AeroCore 2 MAV, as described in the previous section
- A connected compute device running Yocto Linux 3.18+
- A wireless or cellular connection to the internet and to a host machine running QGC
- A connected Logitech® C920 USB webcam

4.1 Software Setup

To provide the channels through which the PX4 instructions and video data will be transmitted, the on-board compute device requires some readily available applications. To install them, first access the console of the attached compute device (I.E. Overo, DuoVero, etc.). This can be done over USB through the microB port labeled **'console'**.

Connect a USB cable to your PC and this console port, power up the AeroCore 2 and enter the following command on a Linux or MacOS machine:

```
$  sudo screen /dev/ttyUSB0 115200
```

You may also connect to the terminal over the network with Socket Secure Shell **(SSH)** with this command:

```
$  ssh username@xxx.xxx.xxx.xxx
```

where `xxx.xxx.xxx.xxx` is replaced with the IP address of the device.

There is more than one way to pipe the mavlink messages to and from the PX4. Two examples are listed here: `mavproxy.py` and `mavlink-socket`. Once the PX4 data is flowing, the video feed can be set up.

4.1.1 Mavproxy.py

Mavproxy.py is a python script from the ArduPilot development team. It is a command-line based ground control station that can also re-transmit commands to another station. This behavior can provide a T-junction wherein QGC can send commands over WiFi and the on-board compute device can also issue in-situ commands. To set up `mavproxy.py` enter the following commands:

```
$ sudo apt-get update

$ sudo apt-get install python-wxgtk3.0 python-pip python-numpy python-dev \
  libxml2-dev libxslt-dev gstreamer1.0-tools

$ pip install pymavlink

$ sudo pip install mavproxy
```

Once the above commands download and install the necessary software, the next step is to establish two data streams to the host machine. The first is the *MAVLink* connection. This is required in order to communicate with QGroundControl for transmitting location and telemetry, and, for pre-defined flight paths and other remote operation functions, receiving commands. *MAVProxy* allows you to pipe PX4 data to and from the on-board UART bridge and the host machine. This is done with a command simmilar to:

```
$  mavproxy.py --master=/dev/ttyMSM1 --baudrate 115200 \
  --out xxx.xxx.xxx.xxx:14550 --aircraft MyCopter --daemon &
```

Where `xxx.xxx.xxx.xxx` is the IP address of your PC. The "**--daemon**" and "**&**" arguments configure the proxy to be "headless" – bypassing the user interface – and force it to run in the background, releasing the command line.

4.1.2 Mavlink-Socket

A more permanent solution is to set up a Linux service that acts as a socket server. Such a service application is available from the Gumstix package repository for Overo and DuoVero. In order to install it on these platforms, `screen` or `ssh` into the COM's console and enter the following commands:

```
$ smart channel -add https://packages.gumstix.com/<release> -y

$ smart update

$ smart install mavlink-socket
```

Mavlink-socket has not been deployed for 3rd party devices, but the source code is available from `https://github.com/GumstixExperimental/mavlink-socket/`.

Once the package has been installed, modify `/etc/mavlink-socket.conf` to match your configuration.

- `TARGET_IP` should be set to your base station's IP address
- `MAVLINK_DEVICE` should be `/dev/tty01` for Overo and DuoVero COMs.

  For Dragonboard 410C, use `/dev/ttyMSM1`

  for Intel® Joule™ module, use `/dev/tty**`
- `MAVLINK_BAUD` should be set to 115200

4.1.3 Video Feed

At this point, QGC will detect the UDP connection and display the MAV's summary page as it did during calibration. If you click on the flight tab, you will see the drone's telemetry data visualized, including gyroscope, compass, altitude and ground speed. It is possible to add more data points to the HUD by clicking on the settings cog.

You will notice that there is a grey box in the lower left-hand corner of the window. This is where the video feed will be monitored. To get the feed set up, another command must be run on the AeroCore 2's compute device. Over your telnet or SSH connection, enter the following *GStreamer* command, or one similar:

```
$ sudo gst-launch-1.0 -v -e v4l2src device=/dev/video1 !  \
  video/x-h264,width=1280,height=720, framerate=30/1 !  \
  h264parse !  rtph264pay pt=127 config-interval=4 !  \
  udpsink host=xxx.xxx.xxx.xxx port=5600 &
```

Again, `xxx.xxx.xxx.xxx` should be the IP address of your PC.

GStreamer command parameters are specific to the type and configuration of the camera being used and it may be necessary to add, remove, or alter entries to suit your particular device. Once *GStreamer* has started, the video feed from the MAV will automatically be displayed by QGC in the flight screen. It is now time to make sure it is ready to fly.

4.1.4 Note: Connection Considerations

Before moving to flight prep, it is important to observe that, if your MAV will be accessing QGC from an external internet connection during operation, ensure that you use your PC's public IP address. If you are behind a router or firewall, it may be necessary to set up port forwarding and/or whitelist parameters for proper communication. Both ports 14550 and 5600 will be needed by QGC in this example.

4.2 Flight

When it is time to deploy the MAV, ensure that it is configured and connected correctly to minimize the risk of drone failure, property damage, or personal injury.

Pre-Flight Checklist

- ESCs connected correctly and securely to PWMs (See Fig 5)
- Rotors firmly attached to frame and rotor blades to rotors
- Battery charged and strapped firmly to frame
- Compute device firmly connected to AeroCore 2
- AeroCore 2 secured to frame
- DSM radio satellite reciever connected and attached
- Secondary power connection from battery connected to AeroCore 2
- *(optional)* GPS module connected and attached with line-of-sight to the sky
- *(optional)* Camera secured to frame and connected to AeroCore 2's USB port

After the drone has been checked, connect the power supply and give the system time to start up. The ESCs will chirp for a few moments and, once the PX4 system as initiated, the buzzer will emit a "STARTUP OK" signal[1].

From your PC, SSH into the MAV's compute device and enter the *MAVProxy* and *GStreamer* commands from the previous section to connect to QGC. Next, activate your R/C controller and arm the drone. The controller's arming and disarming method may differ from one DSM controller to the next. Be sure to read the user manual carefully prior to operation.

Once the rotors spin up, which often takes a moment after arming, the drone is ready for liftoff. Figure 7 labels the axes for common 5+ channel DSM radio controllers, such as the **Spektrum™ DX5e**, providing both aeronautical and colloquial labels. The toggle switches are generally used to switch between flight modes, but for the example, the MAV will remain in "**manual**" mode.

Experiment with manipulating the controls. Increase the throttle to gain altitude. Pan left and right with the yaw axis. Use pitch and roll to move along a horizontal plane. When you are ready to land the drone, ensure that throttle decreases at a slow, steady rate to reduce the force of impact when it hits the ground. Once the drone is stationary on the ground, it is safe to disarm it using the method described in your radio controller's user guide.

[1] For audio recordings of PX4 buzzer signals go to:
http://www.ardupilot.org/copter/docs/common-sounds-pixhawkpx4.html

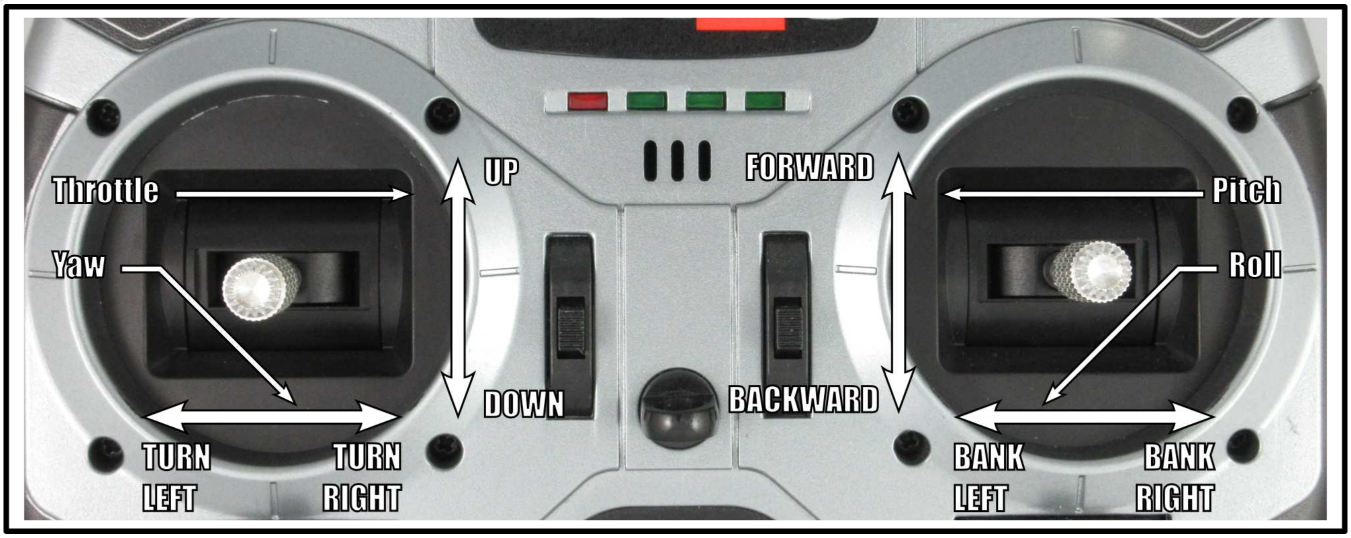

Figure 7: DSM Radio Controller Axes

Finally, shut down the R/C and disconnect the MAV's battery. When you return to QGC it will ask you to save your flight log. Give it a filename and click "**Save**." You will then be able to replay your flight within the QGroundControl interface.

5 Next Steps

In this guide, The fundamental configuration and operation of a MAV powered by an AeroCore 2 expansion board from Gumstix®. Following the above steps will have your drone airborne quickly. What remains is the high-level applications specific to your project. Leverage the computing power of our Gumstix® COMs, Intel® Joule™ module , or the DragonBoard 410C to automate your drone.

The AeroCore 2 expansion boards were designed by Gumstix® engineers in Geppetto™and can be cloned and modified to your project's specifications. for more details on Geppetto™and tutorial videos, go to https://www.gumstix.com/geppetto. Or go straight to https://geppetto.gumstix.com/#!/design/1389/ to begin modifying your own personal addition to the AeroCore 2 family.

List of Acronyms

CAN	Controller Aera Network
COM	Computer on Module
CSI	Camera Serial Interface
DSM	Digital Spectrum Modulation
ESC	Electronic Speed Controller
GPIO	General Purpose Input/Output
GPS	Global Positioning System
I^{2}C	Inter-Integrated Circuit
IMU	Internal Measurement Unit
LED	Light Emitting Diode
MAV	Micro Air Vehicle
MCU	MicroController Unit
MIPI	Mobile Industry Processor Interface
PWM	Pulse Width Modulation
QGC	QGroundControl
R/C	Radio Controlled (Remote Controlled)
RTOS	Real-Time Operating System
SBC	Single Board Computer
SPI	Serial Peripheral Interface
SSH	Socket Secure Shell
UART	Universal Asynchronous Reciever/Transmitter
USB	Universal Serial Bus

External Links