# Integration of Smart Radio with QGroundControl and the Pixhawk 4 Flight Controller

## Introduction

This guide expands on our previous application notes and guides for video streaming and wireless connectivity for Unmanned Aerial Systems [1, 2]. It demonstrates how a Doodle Labs Smart Radio can be used to send MAVlink telemetry data between QGroundControl (QGC) [3] and the Pixhawk 4 Flight Controller (FC) [4] running px4 [5], while integrating a companion computer for video streaming and robotics capabilities.

An overview of the system setup is shown in Figure 1. We assume QGroundControl (QGC) is used as the GCS, and a Pixhawk FC is used on the UAV. There are various ways in which the Video Streamer could be implemented, and in this guide, we will assume a SBC such as the Nvidia Jetson Nano running GStreamer. We will also include the commands required to stream video directly to QGC.

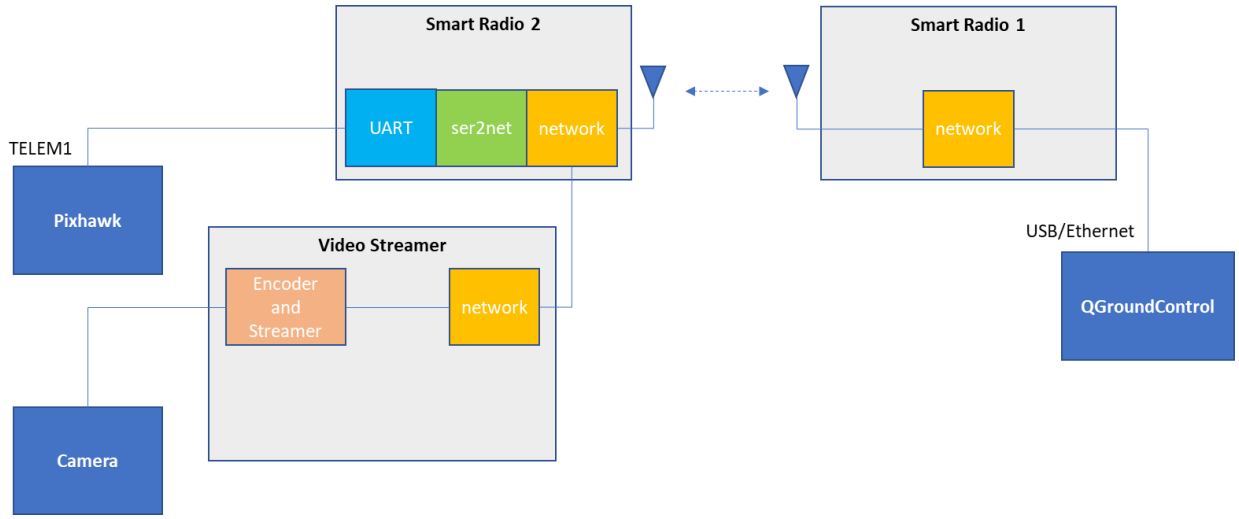



**Figure 1 – System Setup**

© 2020 Doodle Labs. All rights reserved.

# System Setup

## Hardware setup with Pixhawk

By default, the Pixhawk 4 supports telemetry on serial ports TELEM1 and TELEM2. The Smart Radio UART pinout can be found in the Smart Radio Integration Guide [6]. For details on how to setup a serial to serial bridge, please consult the Smart Radio Configuration Guide [6].

Figures 2a and 2b show the TELEM1 pixhawk port to UART Smart Radio port wiring. UART_RX from the Smart Radio should be connected to MCU_TX on the Pixhawk 4, and UART_TX from the Smart Radio should be connected to MCU_RX on the Pixhawk 4. A ground connection is required to complete the loop. VCC_5V, MCU_RTS, and MCU_CTS can be left open.

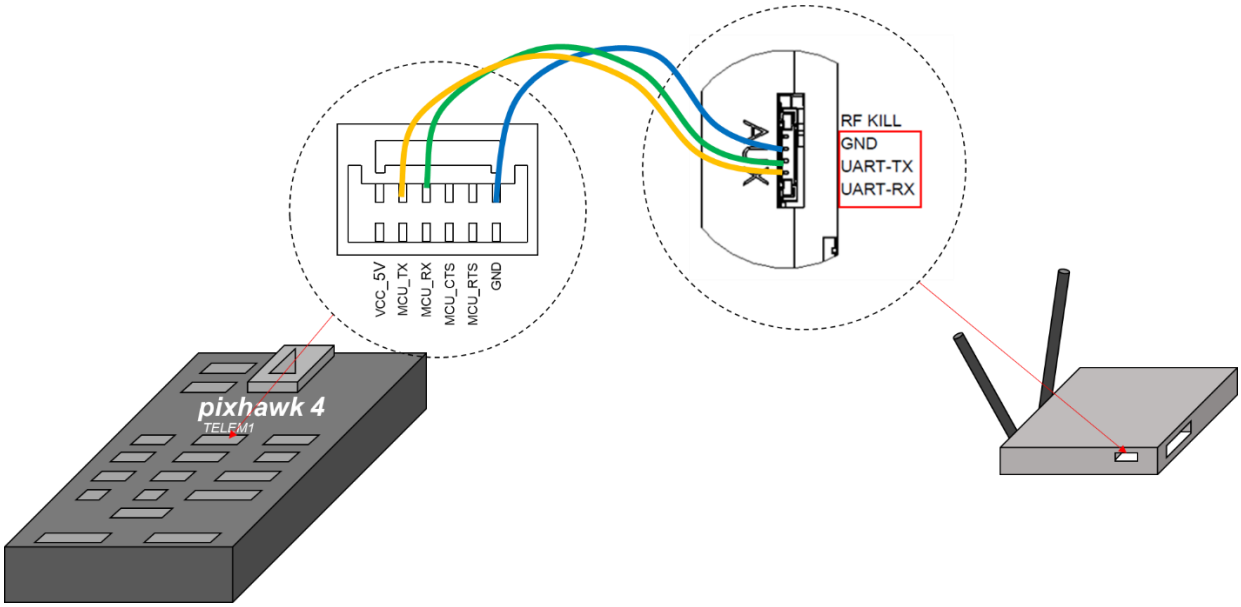



**Figure 2a – Smart Radio -H model UART to Pixhawk 4 TELEM1 wiring**

© 2020 Doodle Labs. All rights reserved.

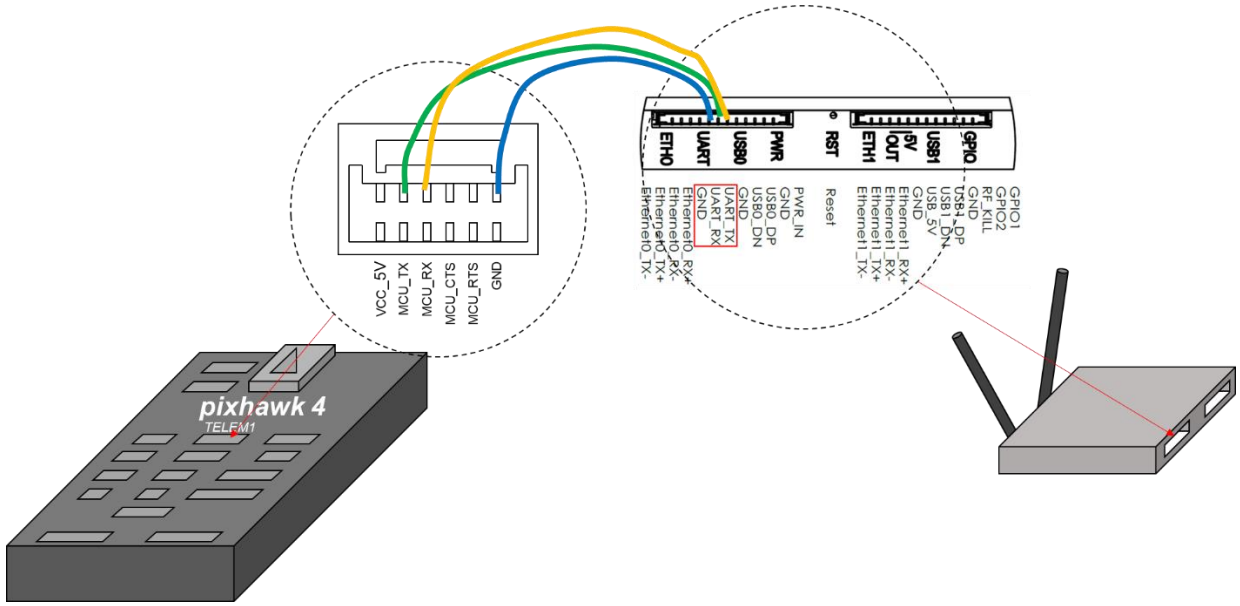



**Figure 2b – Smart Radio -J model UART to Pixhawk 4 TELEM1 wiring**

## Smart Radio Serial Setup

In Figure 1, the serial port on Smart Radio 2 needs to be configured. We suggest you read our Serial Interface Guide [6] for more details on how to use the serial interface. The Smart Radio uses a USB to serial converter IC to provide an external serial port. In -H Smart Radio variants, the serial device is accessible from `/dev/ttyUSB0`, while for -J Smart Radio variants, it is `/dev/ttyACM0`. To check which you have, first login to the radio over SSH. Below is the output for a -H variant.

```
$ ssh root@<IP ADDRESS>

$ ls /dev/tty*

/dev/tty      /dev/ttyS0    /dev/ttyUSB0
```

### Setup using the GUI

The latest release of the Mesh Rider Operating System includes a GUI for serial port configuration. Open up a web browser and navigate to the IP address of the Smart Radio. The Serial Configuration GUI is found in the `Services → Socat Configuration` page. Figure 3 shows a screenshot of the page. The px4 flight controller expects a baud rate of 57600, and you can use device `/dev/uart0` as long as you don't have your own additional serial converter

© 2020 Doodle Labs. All rights reserved.

connected to the radio. You can either run `socat` as a client or a server. In client mode, you should use the IP address of the host running QGC, with UDP transport and port 14550. In server mode, `socat` will listen to an incoming connection on a port or your choice and you may use either TCP or UDP. After making your changes, click `Save & Apply`. Proceed to firewall configuration below.

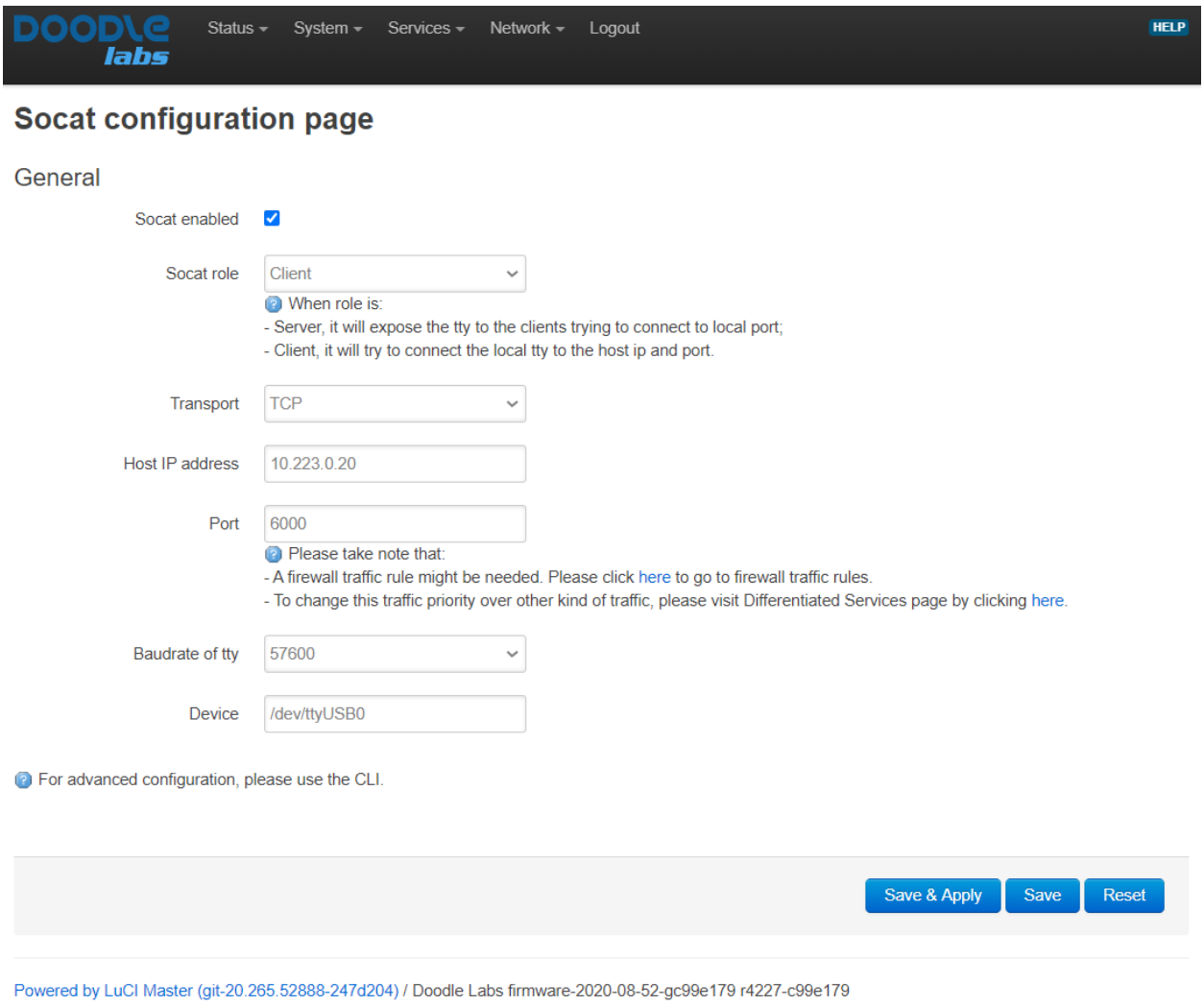



**Figure 3 – Serial Configuration GUI**

## Manual Setup using Ser2net

In general, we recommend updating to the latest software release, and using the GUI for configuration. `ser2net` is installed in the Smart Radio, however the default baud rate is 115200 whereas px4 uses 57600 baud by default. Use `vi` to modify `/etc/ser2net.conf` as necessary,

© 2020 Doodle Labs. All rights reserved.

and restart the service. The commands below show the required `ser2net.conf` configuration followed by a ser2net restart.

```
$ cat /etc/ser2net.conf
3000:raw:0:/dev/ttyUSB0:57600 NONE 1STOPBIT 8DATABITS max-connections=3
$ /etc/init.d/ser2net enable
$ /etc/init.d/ser2net restart
```

You will note that `stty` still shows the wrong configuration for `/dev/ttyUSB0`. This is because `ser2net` will not modify the settings on the serial port until a connection has been made. At this point, `ser2net` is listening on port `3000` for an incoming connection. `ser2net` cannot act as a client and connect directly to QGC's listening port. For that, you may choose to use `socat`, however we will not go over the details here and recommend that you read our Serial Interface Guide [6] instead.

### Firewall setup

You will need to open up the firewall at for your desired network port. Details can be found in the Configuration Guide. Navigate to `network→firewall` in the web GUI, and click on the `Traffic Rules` tab. Under the `Open ports on router` section, add a new rule for the port you want to open (this will be port 14550 or 3000 in the examples above) and click `Save & Apply`.

## QGroundControl Setup

Detailed guides for setting up QGroundControl and Pixhawk are available online [3][4], and we won't detail the steps here. The basic steps are to install QGC on your GCS, hook it directly up to the flight controller over USB, and then program the flight control software. After the flight control software is programmed, you can setup the hardware with the Smart Radio in the link as in figures 4 and 5.

© 2020 Doodle Labs. All rights reserved.

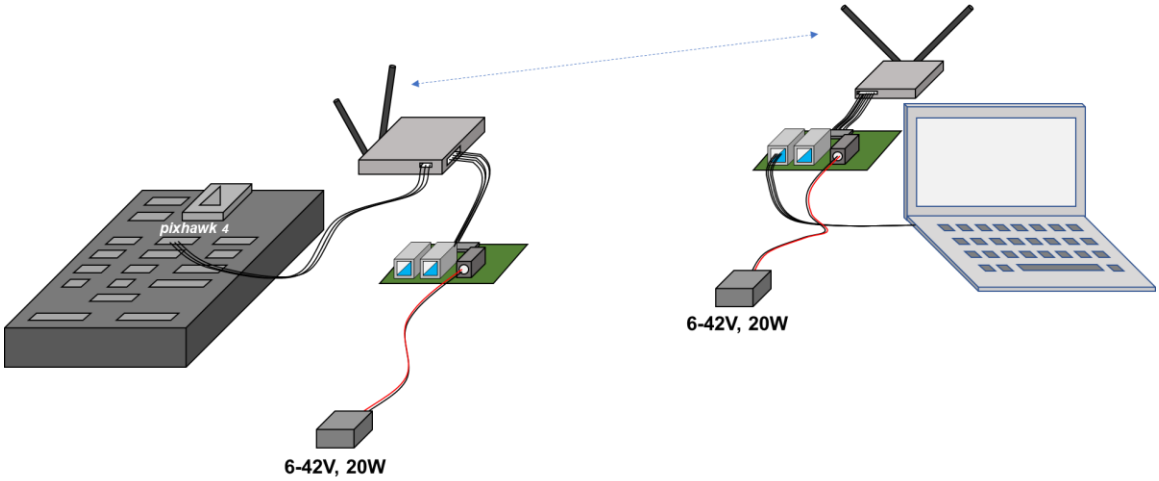



**Figure 4 – Laptop to Drone Telemetry Setup**

## Smart Radio Optimization

The following optimizations apply to both Smart Radio 1 and Smart Radio 2 in Figure 1.

The Smart Radio Configuration Guide includes a section on Common Network Settings which we suggest you review for details on how to configure the Smart Radio. We recommend

- For the RM-2450, 2.4-GHz ISM band radio, do not use 20-MHz bandwidth, use 5/10/15 MHz bandwidths instead to avoid interference from WiFi devices.
- Use the C&C and Voice queue for telemetry (MAVLink) data. In the example above, you would map port 3000 (or 14550) to the CS6 queue. Enable “Optimize Command & Control for Voice and URLLC”.
- Use the Video queue for Video. Enable “Optimize Video Streaming”.

## Network Setup

In Figure 1, the system running QGC, Smart Radio 2, and the Video Streamer all need to be on the same IP subnet. One easy way to do that is to run a DHCP server on one of the systems, and run DHCP clients on the other two systems. However, take note that most Android phones and tablets run a DHCP server by default at the address 192.168.42.129/24. Therefore, for Android systems, no additional setup is necessary for the Smart Radio. For Windows/Linux based

© 2020 Doodle Labs. All rights reserved.

systems, you may run the DHCP server on the Smart Radio. Please consult the Configuration Guide for details on how to setup a DHCP server.

## Video Streamer Setup

A separate application note, “Smart Radio Video Streaming with the NVidia Jetson Nano” [6] details how you can go about setting up a video streamer. , A tip that in our tests, QGC ran faster with  RTP stream using the GStreamer command,

```
gst-launch-1.0 nvarguscamerasrc ! 'video/x-raw(memory:NVMM),width=1920,
height=1080, framerate=30/1, format=NV12' ! nvvidconv ! omxh264enc
iframeinterval=15 control-rate=constant profile=baseline ! video/x-h264,
stream-format=byte-stream ! rtph264pay ! udpsink host=10.223.0.80 port=5000
```

Change the host to the IP address of the GCS, and choose a network port. Note that Video Streaming is optional.

## Android Setup

An Android tablet or phone setup requires root access. Without root access, it may be difficult to get basic information out of the device. A network utilities app may be useful in this case [7].

The recommended setup is shown in Figure 5. Note that Smart Radio is a USB Host/Master. Hence devices with USB slave or OTG capabilities are required. 5V power supply is required to indicate to the USB OTG port that the connected device is a USB Host. On models with -2J suffix, you may use the 5-V output pin on the radio.

© 2020 Doodle Labs. All rights reserved.

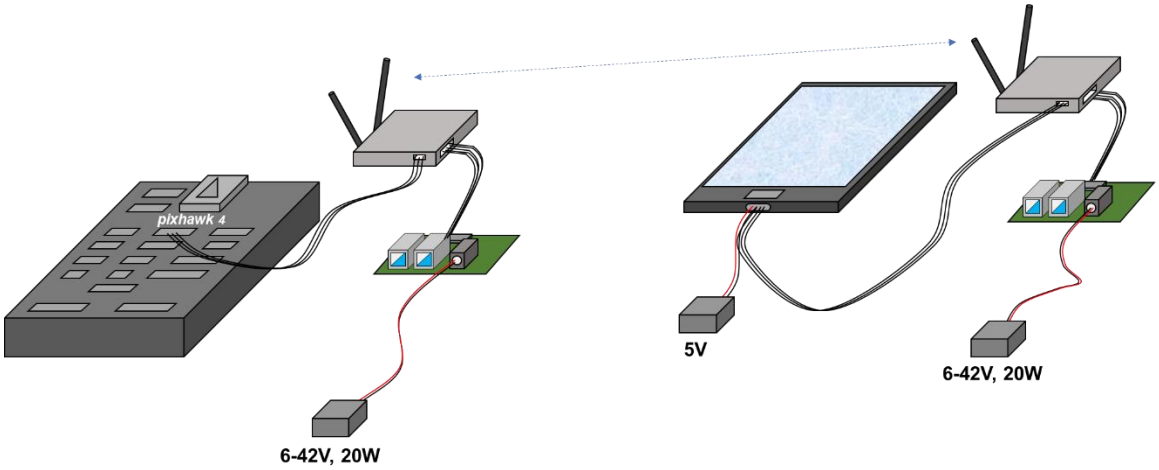



**Figure 5 – QGroundControl MAVLink Verification**

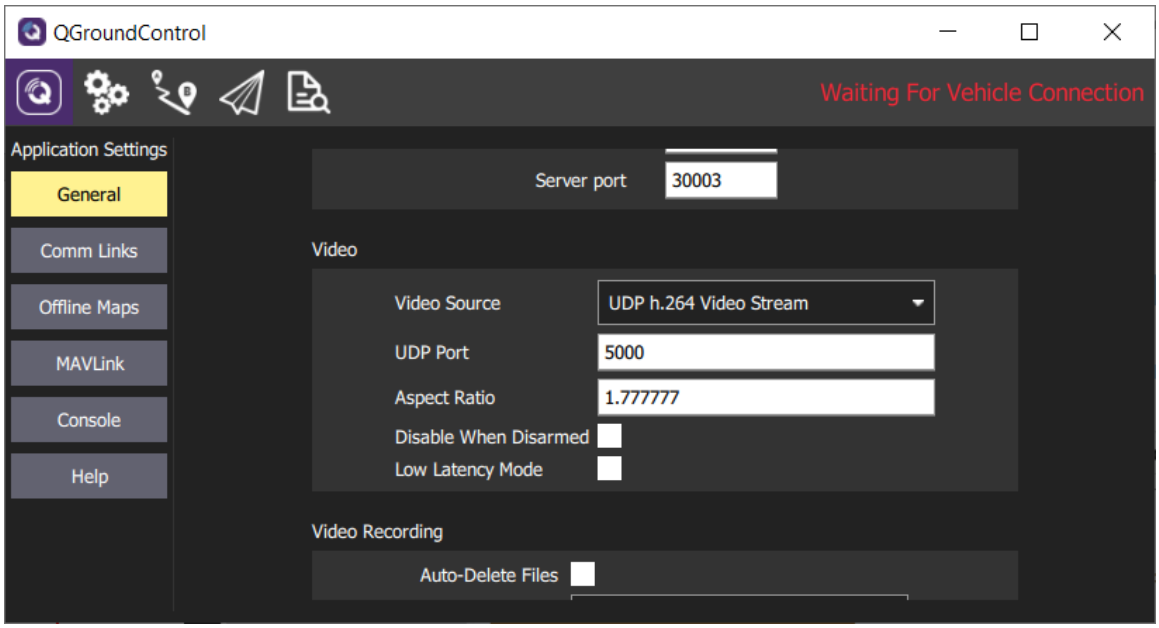
# Testing

We tested the two setups described above with a PC running QGC in Linux, and a Tablet running QGC in Android.

## Video Streaming

You can view your video stream directly in QGC by navigating to the Application Settings menu and scrolling down to the video section. Following our GStreamer Example, you would tell QGC to listen at port 5000 as shown below. You should then be able to see the video stream directly in the main window. For more details, refer to the detailed guide ….

© 2020 Doodle Labs. All rights reserved.




## Mavlink Test

In QGroundControl, add a new Comm Link as shown in Figure 6. Change the host address to the IP address of the Smart Radio connected to the drone, and the network port to the listening port used by `ser2net`. Click *OK*, and then *Connect*.

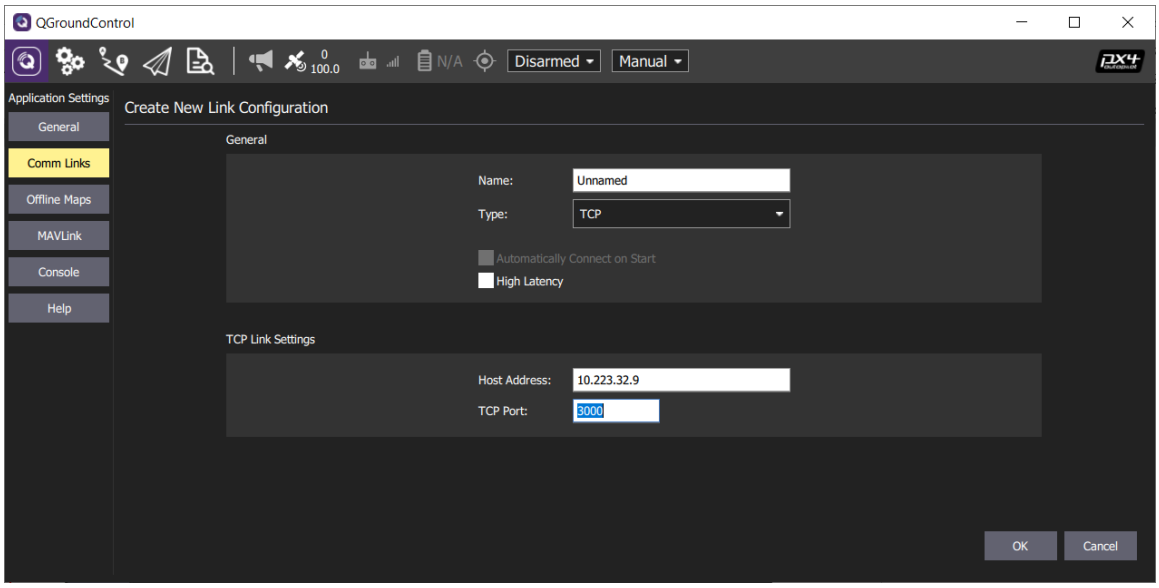



**Figure 6 – QGroundControl Setup**

© 2020 Doodle Labs. All rights reserved.

## Linux/Windows

You can verify the MAVLink connection is good by going to the tab shown in Figure 7, and inspecting the MAVLink heartbeat.

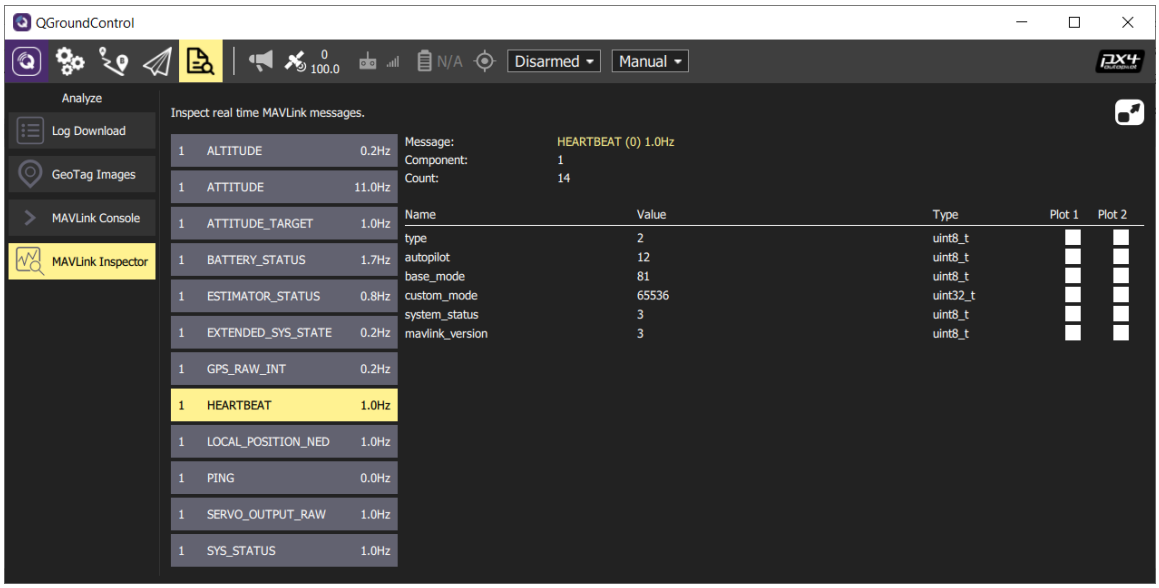



**Figure 7 – QGroundControl MAVLink Verification**

## Android

As far as QGC is concerned, the setup for Android is the same as for Linux. The pictures below show results after a successful connection.

© 2020 Doodle Labs. All rights reserved.

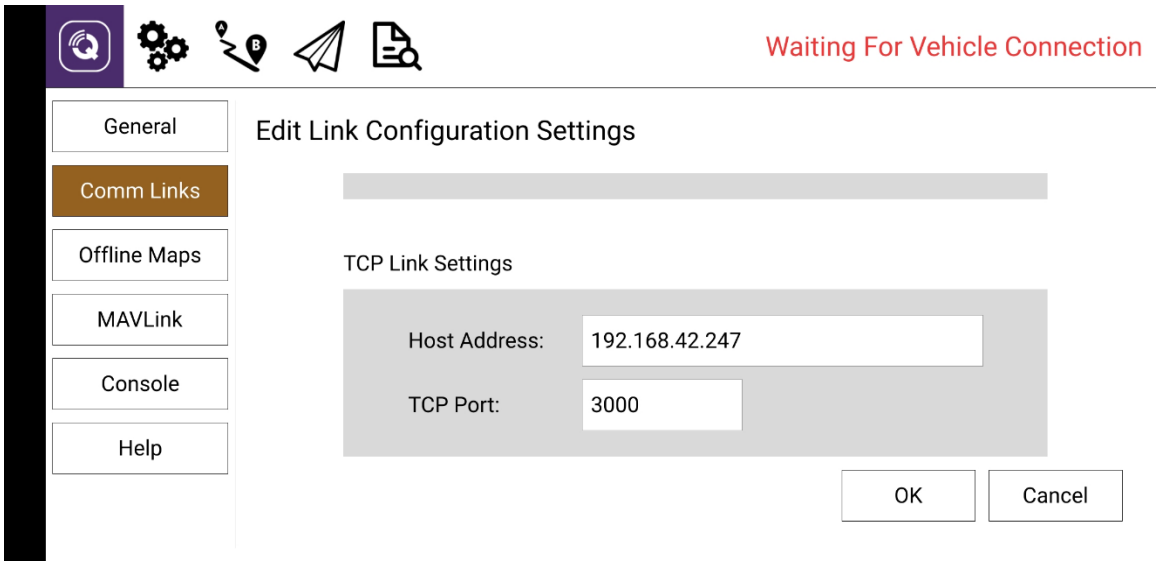



**Figure 9 – QGroundControl A-ndroid Setup**

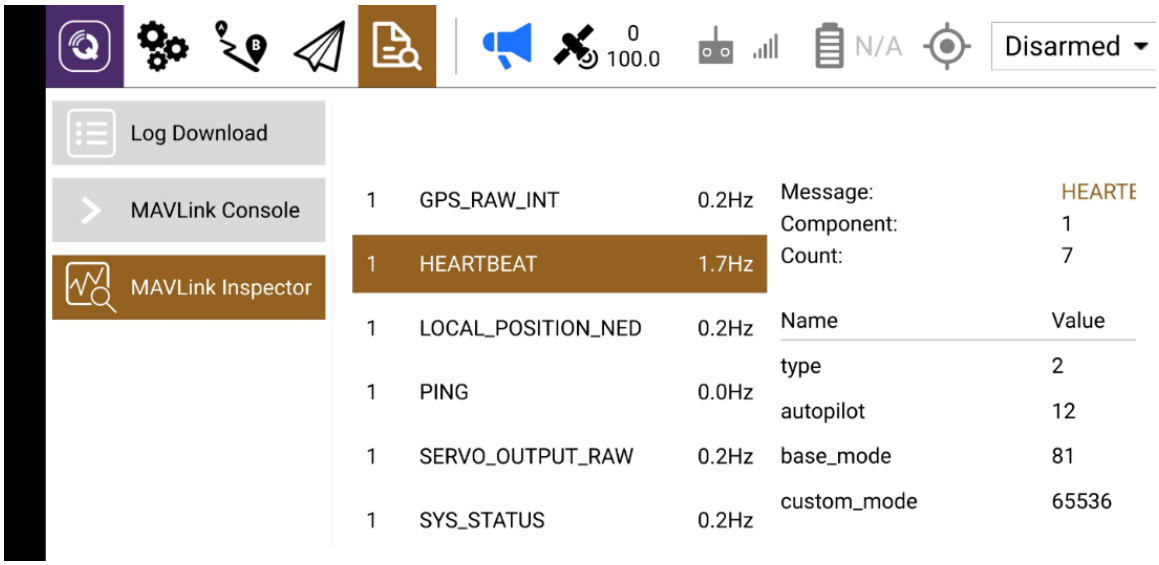



**Figure 10 – QGroundControl Android MAVLink Verification**

© 2020 Doodle Labs. All rights reserved.

# References


[1] Doodle Labs, “Unmanned Aerial Systems”, https://doodlelabs.com/solutions/unmanned-aerial-systems/, 9-7-2020

[2] Doodle Labs, “Smart Radio Video Streaming Tutorial”, https://doodlelabs.com/technologies/technical-library/, 9-7-2020

[3] qgroundcontrol, http://qgroundcontrol.com/, 21-6-2020

[4] Pixhawk 4, https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk4.html, 30-March-2020

px4 autopilot,

[5] https://px4.io/, 21-June-2020

[6] Doodle Labs Technical Library, https://doodlelabs.com/technologies/technical-library/, 5-10-2020

[7] Network Utilities, https://play.google.com/store/apps/details?id=com.myprog.netutils&hl=en_SG, 22-June-2020


© 2020 Doodle Labs. All rights reserved.