



Talking to a PX4 FMU with a RPi via Serial. #noUSB #ArduNope

■ MAVSDK



physicsman

3 Dec '19

Dec 2019

1 / 10

Dec 2019

A guide by a *n00b* for a *n00b*. Henceforth: RPi = Raspberry Pi

- ▶ Click here.
- ▶ ..and here.

1. Configuring the serial on the RPi

- Install Raspbian via [NOOBS](#)
- [Enable SSH](#)
- Most RPi's have two UARTs (serial) which it uses and if you tell it not to it gives you the shitty one of the two. So. [Go here](#) 2, ctrl-F " **Disabling Linux's use of console UART**", and disable Linux's use of console UART. If your RPi has bluetooth, murder it by editing config.txt with `$ sudo nano /boot/config.txt` and add the line: `dtoverlay=disable-bt` to the end of the file, then *double murder* it with `$ sudo systemctl disable hciuart`
- reboot
- The serial port we are going to use is now: `/dev/serial0` all other ttys are just imitating.

2. Make a connection.

- Acquire solder skillz
- Acquire wires and connectors.
- Make thing that connects the RPi UART to TELEM 2 on your flight controller.
 - GND (Pin 6) to GND
 - TXD (Pin 8) to RXD
 - RXD (Pin 10) to TXD

3. Install/build MAVSDK (This gonna suck)

- We're going to build this on the RPi if you know how to cross compile and have it work then wtf are you doing here?
- Get the RPi ready for the pain:
 - `$ sudo apt update`
 - `$ sudo apt upgrade <-- Upgrade all the things`
 - `$ sudo apt install python3-pip`
- [Go here](#) 2 and Follow the '**Building SDK from Source**' instructions for '**Linux**'. Step 5 will take a while so get a drink or four then finish with `cmake --build build/default` then drink some more.
- Install system-wide [as described](#) 4.
- Now we're going to [build the "backend"](#) 1 that will be needed by [MAVSDK-Python](#) 1 because you're a n00b and as far as we're concerned c++ is c+=1. If you follow the instructions the configuration(?) will fail and you'll cry so do this instead:
 - Go to the MAVSDK directory.
 - `$ sudo cmake -DBUILD_BACKEND=ON --symlink-install --cmake-args "-DCMAKE_SHARED_LINKER_FLAGS='-latomic'" "-DCMAKE_EXE_LINKER_FLAGS='-latomic'" -Bbuild/default -H.`
 - `$ sudo cmake --build build/default`
 - Take note, the file `mavsd_server` that you will need for MAVSDK-Python is in the totally obvious directory: `MAVSDK/build/default/src/backend/src/`

4. Install MAVSDK-Python

- Grab a beer or three and [go here](#) 1. Scroll down to "**Build and run from sources**" and follow along from there. You might have an issue when you try `$ which protoc-gen-dcsdk` and nothing shows up. In that case you'll have to add `/usr/local/bin/` to your PATH. I'd explain but I forgot how to do that.
- You tried `$ pip3 install mavsdk`, didn't you?
- Copy that `mavsd_server` file into `MAVSDK-Python/mavsd/bin/`

5. Give it a whirl

- Use that fancy connector to connect the RPi UART to TELEM 2 on your flight controller.
- In QGroundControl set `MAV_2_CONFIG` to `TELEM 2` and reboot the flight controller. (The baud rate is probably going to default to 921600 which you can check by looking at `SER_TEL2_BAUD` in QGroundControl)
- I assume you can SSH into your RPi at this point over WiFi and probably already have. If not, you should do that now. Then go to the MAVSDK-Python directory and `cd examples/`. From there we are going to edit an example with `$ sudo nano firmware_version.py` and replace the `await drone.connect...` line with:

Feb 19

```
await drone.connect(system_address="serial:///dev/serial0:921600")
```

- Now cross your fingers and type (while still in the examples directory) `$ python3 firmware_version.py`. If all went well you should see something like:

```
Waiting for mavsdk_server to be ready...
Connected to mavsdk_server!
Waiting for drone to connect...
Drone discovered with UUID: 8670068434521822000
Version: [flight_sw_major: 1, flight_sw_minor: 10, flight_sw_patch: 0,
flight_sw_vendor_major: 0, flight_sw_vendor_minor: 0, flight_sw_vendor_patch: 0,
os_sw_major: 7, os_sw_minor: 29, os_sw_patch: 0]
```

If you just see it hang at `Waiting for drone to connect...` then a bad thing happed somewhere. Sorry bro.

6. Install MAVLINK Router (Optional)

- Quick and dirty because Im tired:

```
■ Clone this repo and $ cd mavlink-router/
■ $ git submodule update --init --recursive
■ $ sudo apt install python-future
■ $ sudo apt install python3-future
■ $ sudo apt install libtool
■ $ sudo apt install autoconf
■ $ sudo -s <-- root shell (yes rly)
■ # ./autogen.sh && ./configure CFLAGS='-g -O2' \ --sysconfdir=/etc --
    localstatedir=/var --libdir=/usr/lib \ --prefix=/usr
■ # make
■ # make install
```

- Example `main.conf` file:

```
[General]
#Mavlink-router serves on this TCP port
TcpServerPort=5790
ReportStats=false
MavlinkDialect=common
Log=/home/pi/logs/mavlink-router
DebugLogLevel=debug

[UartEndpoint alpha]
Device=/dev/serial0
Baud=921600,115200,57600
FlowControl=false

[UdpEndpoint bravo]
Mode=normal
Address=127.0.0.1
Port=14550
```

Goodnight you Princes of Maine, you Kings of New England!

6

created	last reply	9	394	4	8	9	
Dec '19	Feb 19	replies	views	users	likes	links	



JonasVautherin Dev Team

Dec '19

Nice report! This reminds me that we should really try to get a python wheel for RPi on PyPi, so that you could just `pip install mavsdk` there as well 🤖...

2



physicsman

Dec '19

A follow up to the issue with `$ which protoc-gen-dcsdk` not showing up in `$PATH`.

I used `$ sudo find / -name protoc-gen-dcsdk` to find the directory that contained `protoc-gen-dcsdk` (in my case it was: `/home/pi/.local/bin/`) I then added that directory to `$PATH` with the command `$ export PATH=$PATH:<path to protoc-gen-dcsdk>`. After that I was all set.



Quirks (Part 1?):


The RPi Zero W uses armv6 and has issues with `libatomic` which is why I added (read: looked up error on google and copied soln) the `-latomic` flags for the MAVSDK build. For other RPi with armv7 + this might not be needed.

The RPi 3 A+ does not have enough memory to build MAVSDK which I didn't realize until just now. If I have a free USB port on my RPi I usually put a low profile flash drive in and make it swap which solves memory issues and I suspect helps take load off the SD card. (<https://askubuntu.com/questions/173676/how-to-make-a-usb-stick-swap-disk>)

21 DAYS LATER

1 MONTH LATER

Hello! Looks like you're enjoying the discussion, but you haven't signed up for an account yet.

When you create an account, we remember exactly what you've read, so you always come right back where you left off. You also get notifications, here and via email, whenever someone replies to you. And you can like posts to share the love. 

Sign Up

Remind me tomorrow

[no thanks](#)

Suggested Topics

Topic

Replies

Views

Activity

Topic	Replies	Views	Activity
Get Gimbal Pitch and Yaw ■ MAVSDK	8	279	5d
[VTOL SITL] Issue with offboard_velocity example ■ MAVSDK	0	88	Sep '19
Parrot bebop2 px4 v1.9.1 and MAVSDK ■ MAVSDK	1	119	Jul '19
MavROS parallel DroneCodeSDK ■ MAVSDK	2	187	May '19
Mavsdk or Mavlink compatible ■ MAVSDK	1	134	Oct '19

Want to read more? Browse other topics in ■ MAVSDK or [view latest topics](#).