Interfacing Raspberry-pi with APM flight controller

Akshit Gandhi

June 17, 2016

Contents

1 Objective

Topic: Interfacing Raspberry-pi with APM flight controller In this tutorial we will learn to how to setup a communication between the r-pi and APM flight controller, so that we can run/write our python scripts on r-pi and execute them on our drone. We will be going into details of each step.

2 Prerequisites

It's better if you first go through the official documentation on ardupilot's website on connecting r-pi with APM. The link is: `http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html`, it will be beneficial also for this tutorial and also for later tutorials.

3 Hardware Requirement

Drone or Quadcopter, Raspberry pi, telemetry port connector for connecting APM to r-pi and few jumpers(M-F).

4 Procedure

This tutorial explains how to connect and configure a Raspberry Pi (RPi) so that it is able to communicate with a APM flight controller using the MAVLink protocol over a serial connection

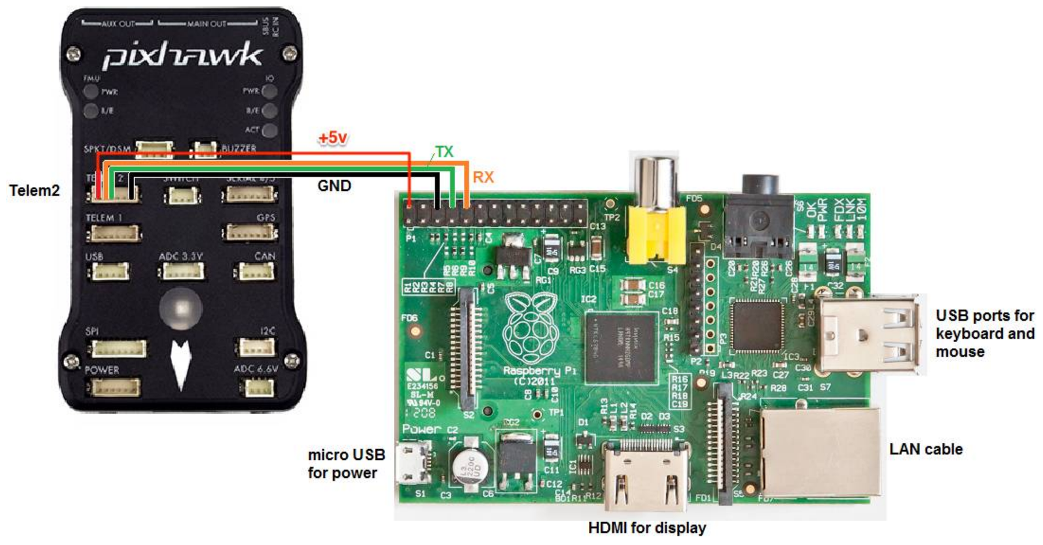


Figure 1: Connection diagram.

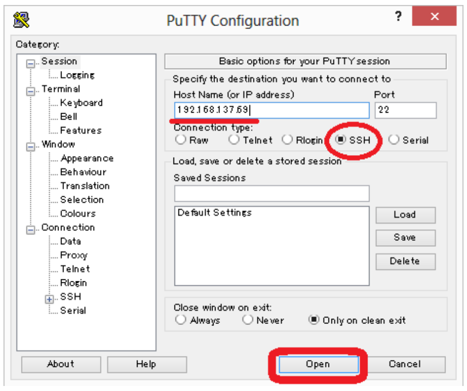
The picture shown above is for pixhawk controller but the same connection works for APM also.

4.1 Connecting to RPi with an SSH/Telnet client

NOTE: These steps assume that you have already set-up your RPi so that it is running Raspbian.

To avoid the requirement to plug a keyboard, mouse and HDMI screen into your RPi it is convenient to be able to connect from your Desktop/Laptop computer to your RPi using an SSH/Telnet client such as PuTTY.

- Connect the RPi to your local network by one of the following methods:
  - Connecting an Ethernet LAN cable from the RPi board to your Ethernet router, or
  - Use a USB dongle to connect your RPi to the local wireless network, or
  - Connect the Ethernet LAN cable between the RPi and your desktop/laptop computer. Open the control panels Network and Sharing Center, click on the network connection through which your desktop/laptop is connected to the internet, select properties and then in the sharing tab, select Allow other networks to connect through this computers Internet connection
- Determine the RPis IP address:
  - If you have access to the RPis terminal screen (i.e.you have a screen, keyboard, mouse connected) you can use the /sbin/ifconfig command.
  - If your Ethernet router has a web interface it may show you the IP address of all connected devices
- Connect with Putty:



Specify the destination you want to connect to
Host Name (or IP address): 192.168.137.69
Port: 22
Connection type: Raw Telnet Rlogin [✓ SSH] Serial
Close window on exit: Always Never [✓ Only on clean exit]

Figure 2: Connect screen with Putty.

If all goes well you should be presented with the regular login prompt to which you can enter the username/password which defaults to pi/raspberry

4.2 Install the required packages on the Raspberry Pi

Log into the RPi board (default username password is pi/raspberry) and check that its connection to the internet is working. If there's some problem then troubleshooting guide can be found on the documentation site `http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html`

After the internet connection is confirmed to be working install these packages:

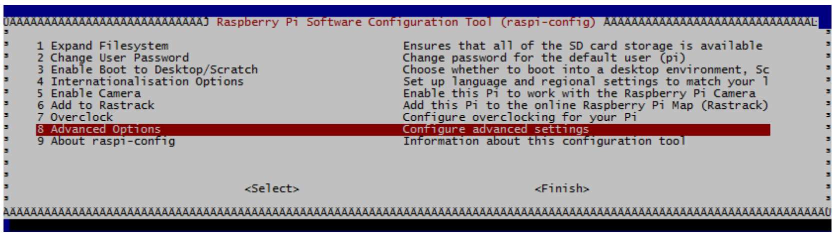
- sudo apt-get update
- sudo apt-get install screen python-wxgtk2.8 python-matplotlib python-opencv python-pip python-numpy python-dev libxml2-dev libxslt-dev
- sudo pip install pymavlink
- sudo pip install mavproxy

4.3 Disable the OS control of the serial port

Use the Raspberry Pi configuration utility for this.

Type:

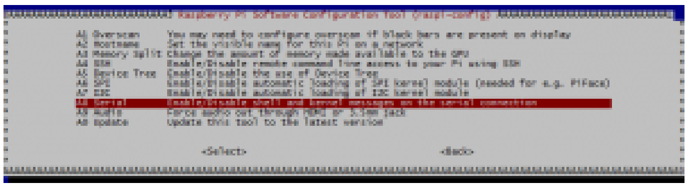
- sudo raspi-config
- And in the utility, select Advanced Options:



```
Raspberry Pi Software Configuration Tool (raspi-config)

  1 Expand Filesystem                  Ensures that all of the SD card storage is available
  2 Change User Password               Change password for the default user (pi)
  3 Enable Boot to Desktop/Scratch     Choose whether to boot into a desktop environment, Sc
  4 Internationalisation Options       Set up language and regional settings to match your l
  5 Enable Camera                      Enable this Pi to work with the Raspberry Pi Camera
  6 Add to Rastrack                    Add this Pi to the online Raspberry Pi Map (Rastrack)
  7 Overclock                          Configure overclocking for your Pi
  8 Advanced Options                   Configure advanced settings
  9 About raspi-config                 Information about this configuration tool

                  <Select>                         <Finish>
```

Figure 3: RasPiConfiguration Utility: Serial Settings: Advanced Options.

  - And then Serial to disable OS use of the serial connection:



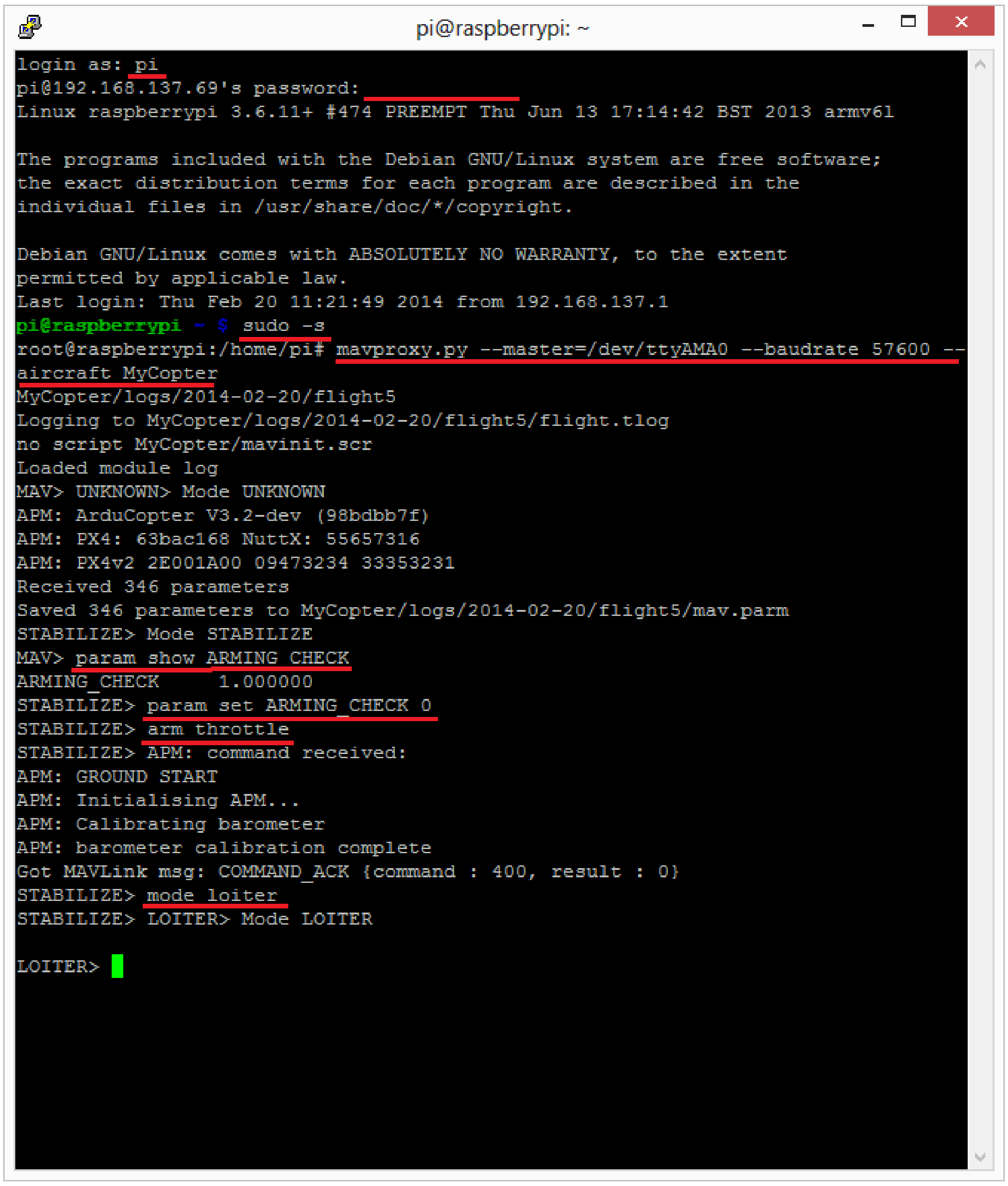
```
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan       You may need to configure overscan if black bars are present on display
A2 Hostname       Set the visible name for this Pi on a network
A3 Memory Split   Change the amount of memory made available to the GPU
A4 SSH            Enable/Disable remote command line access to your Pi using SSH
A5 Device Tree    Enable/Disable the use of Device Tree
A6 SPI            Enable/Disable automatic loading of SPI kernel module (needed for e.g. PiFace)
A7 I2C            Enable/Disable automatic loading of I2C kernel module
A8 Serial         Enable/Disable shell and kernel messages on the serial connection
A9 Audio          Force audio out through HDMI or 3.5mm jack
A0 Update         Update this tool to the latest version

          <Select>                    <Back>
```

Reboot the Raspberry Pi when you are done.

4.4 Testing the connection

To test the RPi and APM are able to communicate with each other first ensure the RPi and APM are powered, then in a console on the RPi type:

- sudo -s
- mavproxy.py –master=/dev/ttyAMA0 –baudrate 57600 –aircraft MyCopter Note: In the above command it's MyCopter and My-Copter
- Once MAVProxy has started you should be able to type in the following command to display the `ARMING_CHECK` parameters value
  - param show `ARMING_CHECK`
  - param set `ARMING_CHECK` 0
  - arm throttle
    - And then Serial to disable OS use of the serial connection:



```
login as: pi
pi@192.168.137.69's password:
Linux raspberrypi 3.6.11+ #474 PREEMPT Thu Jun 13 17:14:42 BST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb 20 11:21:49 2014 from 192.168.137.1
pi@raspberrypi ~ $ sudo -s
root@raspberrypi:/home/pi# mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --
aircraft MyCopter
MyCopter/logs/2014-02-20/flight5
Logging to MyCopter/logs/2014-02-20/flight5/flight.tlog
no script MyCopter/mavinit.scr
Loaded module log
MAV> UNKNOWN> Mode UNKNOWN
APM: ArduCopter V3.2-dev (98bdbb7f)
APM: PX4: 63bac168 NuttX: 55657316
APM: PX4v2 2E001A00 09473234 33353231
Received 346 parameters
Saved 346 parameters to MyCopter/logs/2014-02-20/flight5/mav.parm
STABILIZE> Mode STABILIZE
MAV> param show ARMING_CHECK
ARMING_CHECK     1.000000
STABILIZE> param set ARMING_CHECK 0
STABILIZE> arm throttle
STABILIZE> APM: command received:
APM: GROUND START
APM: Initialising APM...
APM: Calibrating barometer
APM: barometer calibration complete
Got MAVLink msg: COMMAND_ACK {command : 400, result : 0}
STABILIZE> mode loiter
STABILIZE> LOITER> Mode LOITER

LOITER>
```

Figure 4: Terminal screen.

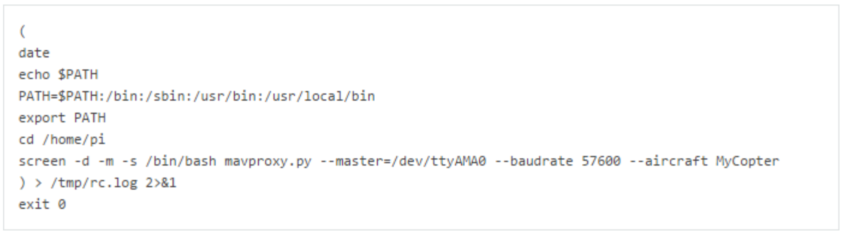
NOTE: If you get an error about not being able to find log files or if this example otherwise doesnt run properly, make sure that you havent accidentally assigned these files to another username, such as Root.

Entering the following at the Linux command line will ensure that all files belong to the standard Pi login account:

∗ sudo chown -R pi /home/pi

4.5 Configure MAVProxy to always run

To setup MAVProxy to start whenever the RPi is restarted open a terminal window and edit the /etc/rc.local file, adding the following lines just before the final exit 0 line:



```
(
date
echo $PATH
PATH=$PATH:/bin:/sbin:/usr/bin:/usr/local/bin
export PATH
cd /home/pi
screen -d -m -s /bin/bash mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --aircraft MyCopter
) > /tmp/rc.log 2>&1
exit 0
```

To open /etc/rc.local file type in sudo nano /etc/rc.local and type the code below and save it in the file

If you wish to connect to the MAVProxy application that has been automatically started you can log into the RPi and type: sudo screen -x

It is also worth noting that MAVProxy can do a lot more than just provide access to your APM. By writing python extension modules for MAVProxy you can add sophisticated autonomous behaviour to your vehicle. A MAVProxy module has access to all of the sensor information that your APM has, and can control all aspects of the flight.

5 References

`http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html`

Everything mentioned above bit by bit is from the official documentation.