**Raspberry pi communication using MAVLink**

This guide is used to explain the procedure of connecting and configuring a raspberry pi so that it can communicate with a flight controller (Pixhawk for example) using the MAVLink protocol using serial connection.

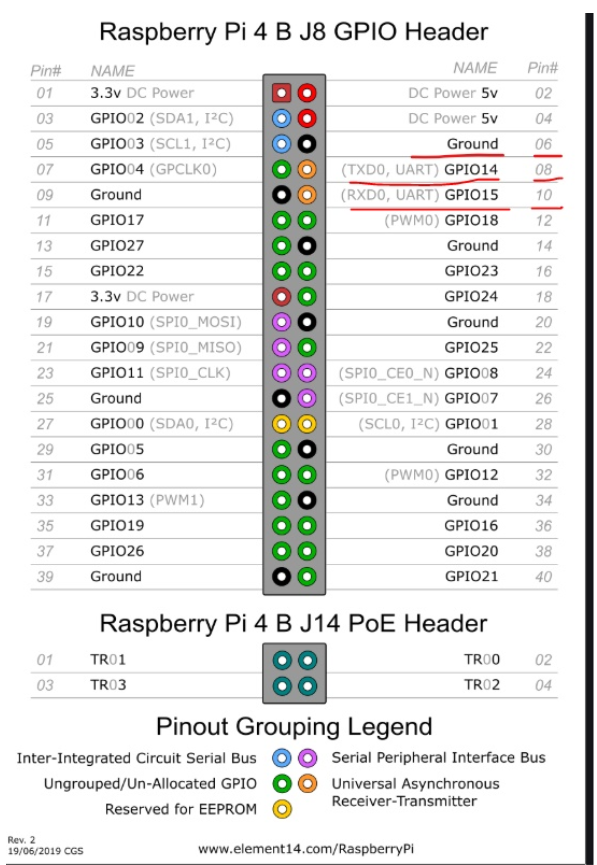
**Step 1:**

Download mission planner on your pc/laptop following this URL: https://ardupilot.org/planner/docs/mission-planner-installation.html

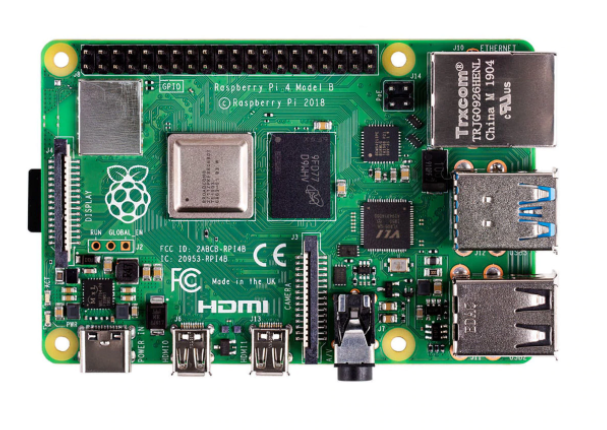
**Step 2:**

Locate the RPi’s and Pixhawk’s Ground, TX and RX pins.

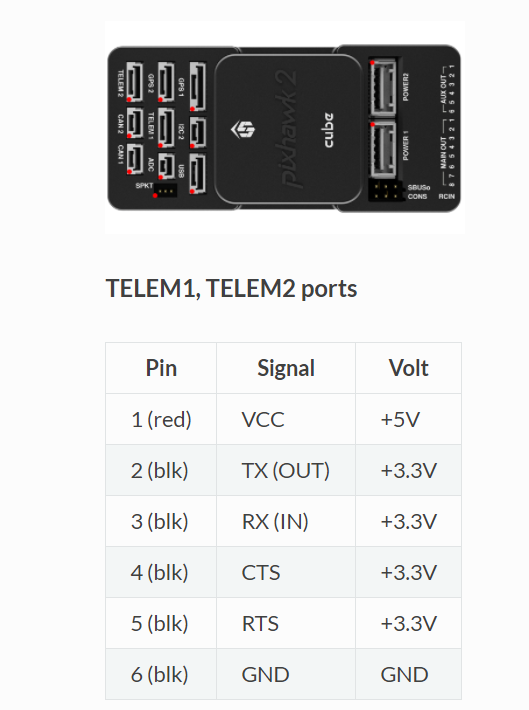
**Raspberry Pi 4 pinout diagram:**



GND TX RX



**Pixhawk pinout diagram:**



Hardware connection between the RPi and Pixhawk is shown below:



Wire from the Pixhawk: BLACK – GND; BLUE – RX; WHITE – TX   
When connecting the TX and RX from the Pixhawk, remember it goes into the opposite ports of the RPi. For example:

**RPi TX – Pixhawk RX  
RPi RX – Pixhawk TX**

Connect the USB cable from the Pixhawk to your laptop running mission planner.

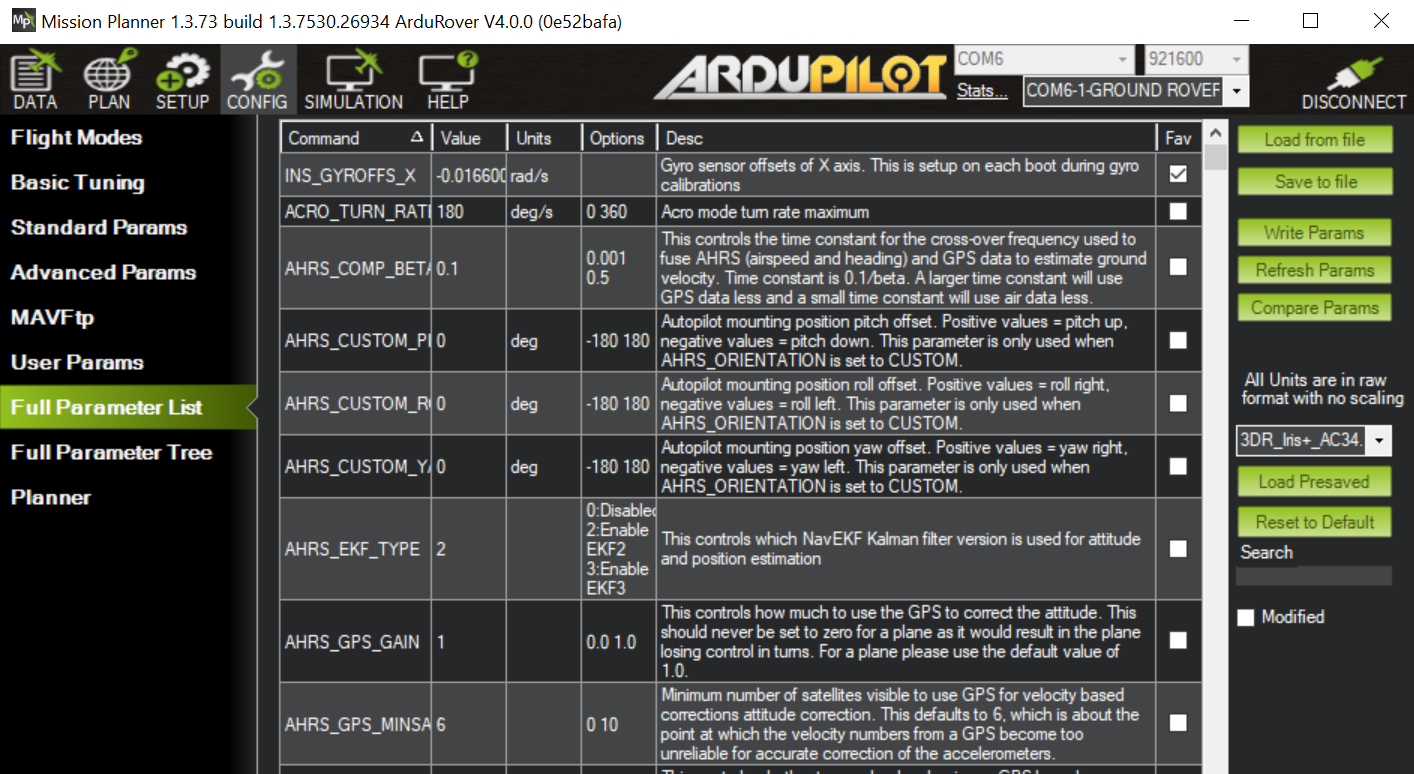


**Step 3:**

Connect the Pixhawk to Mission Planner by clicking on the “CONNECT” button located on the top right corner:



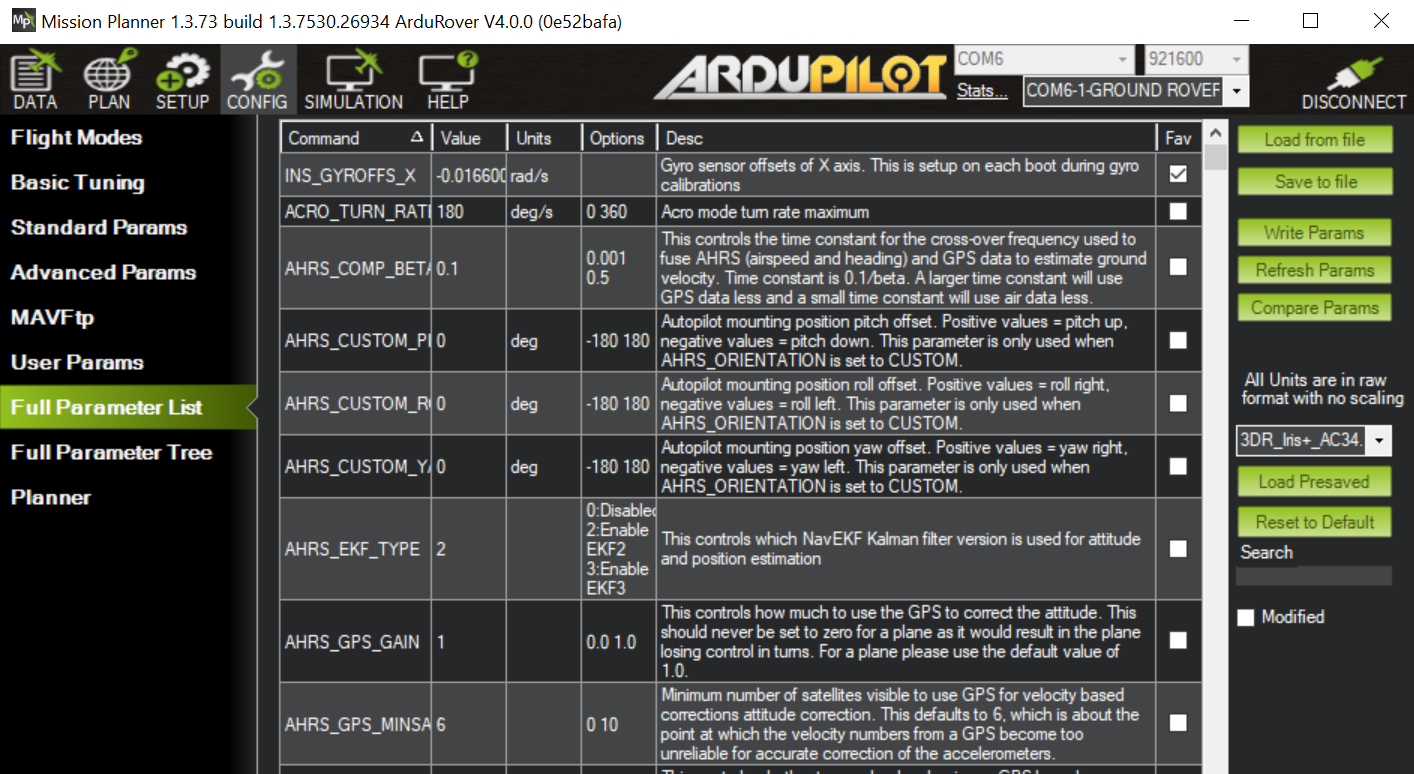
Click on “Config” and then “Full parameter list” to get the entire list of parameters:



Change the following parameters in mission planner software:

* [**SERIAL2\_PROTOCOL**](https://ardupilot.org/copter/docs/parameters.html#serial2-protocol)**= 2** (the default) to enable MAVLink 2 on the serial port.
* [**SERIAL2\_BAUD**](https://ardupilot.org/copter/docs/parameters.html#serial2-baud)**= 921** so the flight controller can communicate with the RPi at 921600 baud.
* [**LOG\_BACKEND\_TYPE**](https://ardupilot.org/copter/docs/parameters.html#log-backend-type)**= 3** if you are using APSync to stream the dataflash log files to the RPi

Once changed, click on “Write Params” and then on “Disconnect” and repower the Pixhawk by removing it from the power supply and plugging it back in:



After that, press “Connect” as shown above.

**Step 4:**

Setup the RPi software for communicating with the flight controller; it uses the MAVLink protocol for communication. We will use MAVProxy as the software to be installed on the RPi.

MAVProxy is used to send commands to the flight controller from the RPi to be able to control the Pixhawk remotely.

Parameters on the Pixhawk can be changed using MAVProxy and the Pixhawk can also be controlled using the same MAVProxy.

To install MAVProxy on the RPi, use the following code in the command terminal as shown below:

**For Python 3 on Debian based systems (including Ubuntu, WSL, Raspian):**

sudo apt**-**get install python3**-**dev python3**-**opencv python3**-**wxgtk4**.**0 python3**-**pip python3**-**matplotlib python3**-**lxml python3**-**pygame

pip3 install PyYAML mavproxy **--**user

echo "export PATH=$PATH:$HOME/.local/bin" **>>** **~/.**bashrc

**For Python 2 on Debian based systems (including Ubuntu, WSL, Raspian):**

sudo apt**-**get install python**-**dev python**-**opencv python**-**wxgtk4**.**0 python**-**pip python**-**matplotlib python**-**lxml python**-**pygame

pip install PyYAML mavproxy **--**user

echo "export PATH=$PATH:$HOME/.local/bin" **>>** **~/.**bashrc

Reboot the RPi after running the above installation.

Further installations:

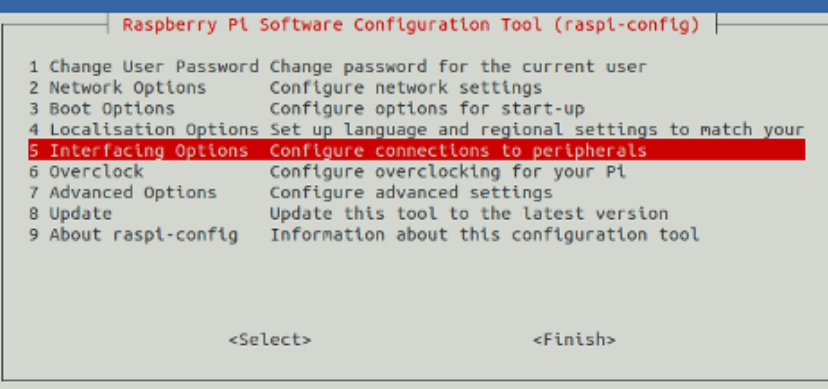
* sudo apt-get install screen python-wxgtk2.8 python-matplotlib python-opencv python-pip python-numpy python-dev libxml2-dev libxslt-dev
* sudo pip install pymavlink
* sudo pip install mavproxy
* sudo pip install future

**Step 5:**

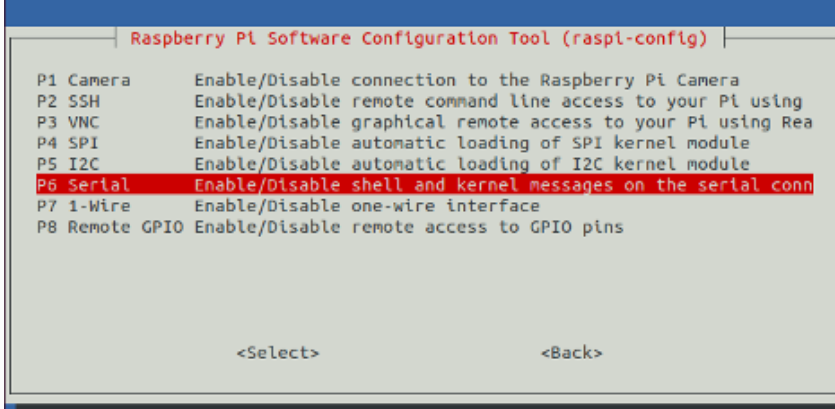
Configure the serial port (UART) using the following command:

**sudo raspi-config**

In the utility, select “Interfacing Options”:



And then “Serial”:



When prompted, select no to “Would you like a login shell to be accessible over serial?”.

When prompted, select yes to “Would you like the serial port hardware to be enabled?”.

Reboot the Raspberry Pi when you are done.

The Raspberry Pi’s serial port will now be usable on /dev/serial0.

**Step 6:**

To test the RPi and flight controller can communicate with each other first ensure the RPi and flight controller are powered, then in a console on the RPi type:

python3 mavproxy**.**py **--**master**=/**dev**/**serial0 **--**baudrate 921600 **--**aircraft MyCopter

The baudrate should match the one specified in mission planner. The port also should be the same and the connections should be correct in order to get a heartbeat.

Three files are created and stored:

1. mav.parm (contains all the parameters that have been set in mission planner)
2. flight.tlog (contains the data that have been updated while the Pixhawk was in flight)
3. flight.tlog.raw

**NOTE:**

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

Refer to the following URL for messages in MAVLink:

https://mavlink.io/en/messages/common.html