Build a MAV bridge for Yuneec Typhoon H Plus or H520

1. Abstract

Without camera H Plus or H520 will not provide telemetry to ST16S thus and no automated flight modes are possible. If one needs another payload for those drones it needs another device for communication with ST16S. A MAVbridge on base of ESP8266 like this here is the solution. Lets bulid one.

Learn more about ESP8266 modules:
https://www.esp8266.com/wiki/doku.php?id=esp8266-module-family
http://stefanfrings.de/esp8266

2. Preparations

2.1 Download tools

Download following tools depending on your OS:

Espressif esptool: https://github.com/espressif/esptool/releases
ESPflasher: https://github.com/h-elsner/ESPflasher

ESPflasher is a simple GUI for the Espressif esptool offering only some important functions of the esptool but all we need here.

Note:

Supported OS	esptool	ESPflasher
Raspberry Pi (32bit)	esptool-v4.6.2-arm.zip	ESPflasher_arm.zip
Ubuntu LINUX	esptool-v4.6.2-linux-amd64.zip	ESPflasher_LINUX.zip
Windows 10	esptool-v4.6.2-win64.zip	ESPflasher_win.zip

2.2 Download firmware

MAV bridge ESP6266 firmware: http://www.grubba.com/mavesp8266/firmware-1.2.2.bin

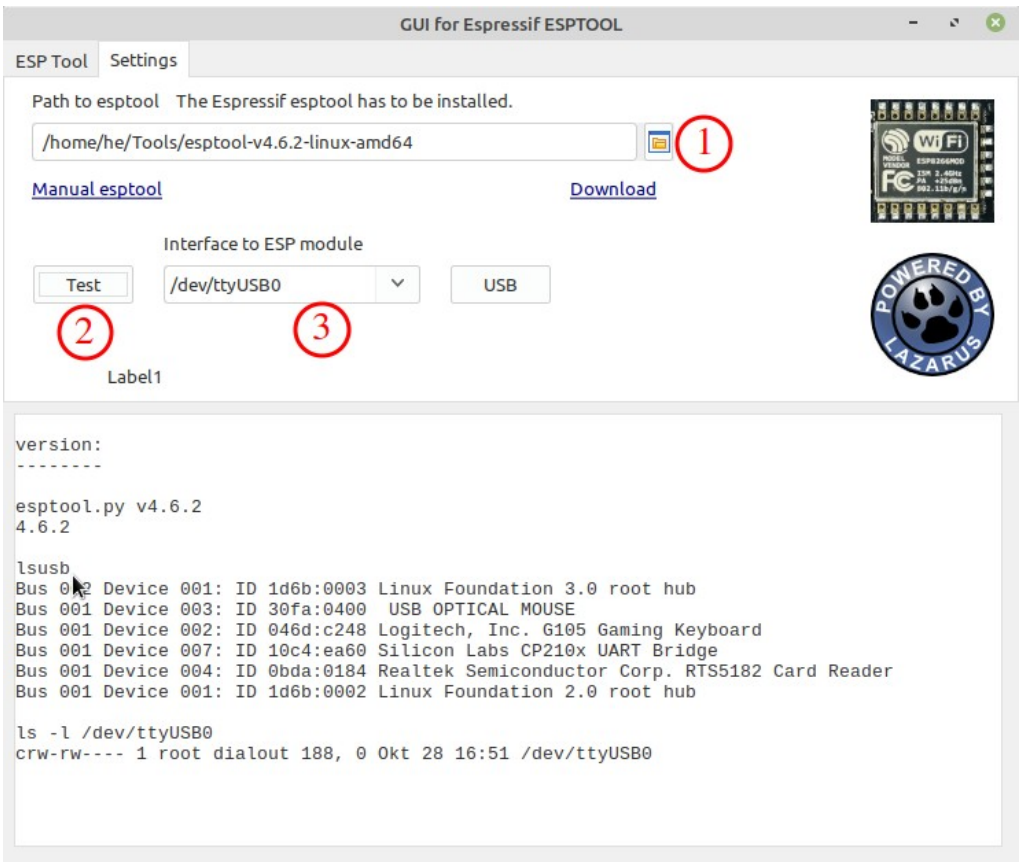
2.3 Installation

Create a folder where you want to have the flash tools. Create a subfolder /bin there. Move downloaded zip files to this folder and unzip all, move firmware file to a subfolder /bin. This subfolder can be used for all binary firmware files that you want to flash on an ESP8266, no matter for what pupose.

Note:

- For Windows you may have to install HW dependent drivers for the device you connect to USB. At ESP-01 this is a USB-Serial converter, at NodeMCU boards the onboard USB-Serial chip.
- For UNIX-like OS you may have to make esptool executable: `chmod +x esptool`

Open ESPflasher > go to Settings page > enter the path to esptool **(1)**.



Connect your device you want to flash to USB port (no need to bring it into flash mode at this point). Click on Test in ESPflasher **(2)**.

For **UNIX-like OS** you should get an output like that:

```
version:
--------
esptool.py v4.6.2
4.6.2

lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 30fa:0400  USB OPTICAL MOUSE
Bus 001 Device 002: ID 046d:c248 Logitech, Inc. G105 Gaming Keyboard
Bus 001 Device 005: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
Bus 001 Device 004: ID 0bda:0184 Realtek Semiconductor Corp. RTS5182 Card Reader
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

ls -l /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 0 Okt 28 08:24 /dev/ttyUSB0
```

If installation of esptool is OK and path to esptool proper set then you will get the **version number** of the esptool.
In the second section you will see the **devices** that are connected to USB. Check if you device is in the list.
In the third section you will get the **port the device is assigned to**. Usually it is **/dev/ttyUSB**. If your USB device is not there, unplug and plug it again USB connection. Often this helps.

Select the listed port as interface in ESPflasher **(3)**.

For **Windows** you should get an output like that:

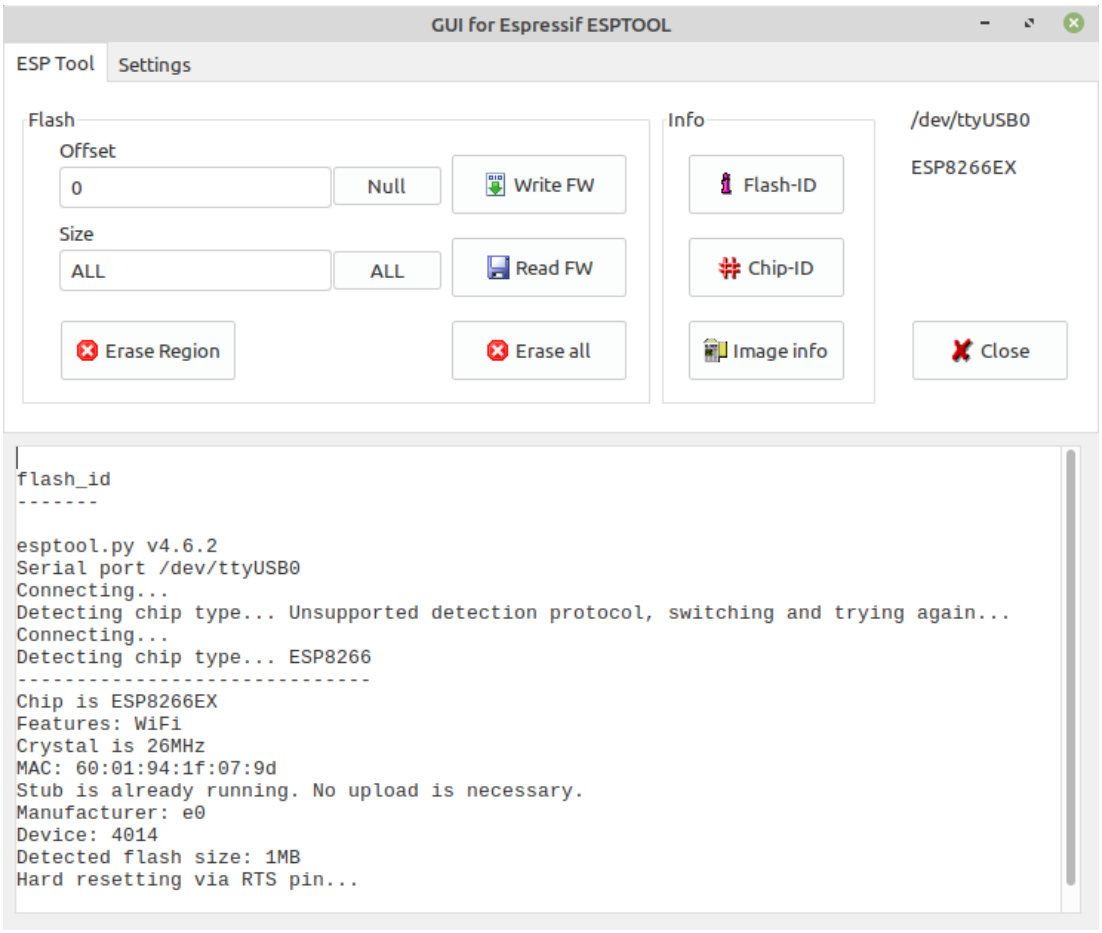
```
version:
--------
esptool.py v4.6.2
4.6.2

Ports:
------
COM1
COM4
COM6
```

If installation of esptool is OK and path to esptool proper set then you will get the **version number** of the esptool.
In the second section you will see the **available COM ports**. If no new COM port appeared, unplug and plug again USB connection. Often this helps and it helps to identify the correct COM port, usually the one with the highest number.

Select the used COM port from the list as interface in ESPflasher **(3)**.

Now we have a complete environment to flash ESP8266 modules with any binary firmware files.

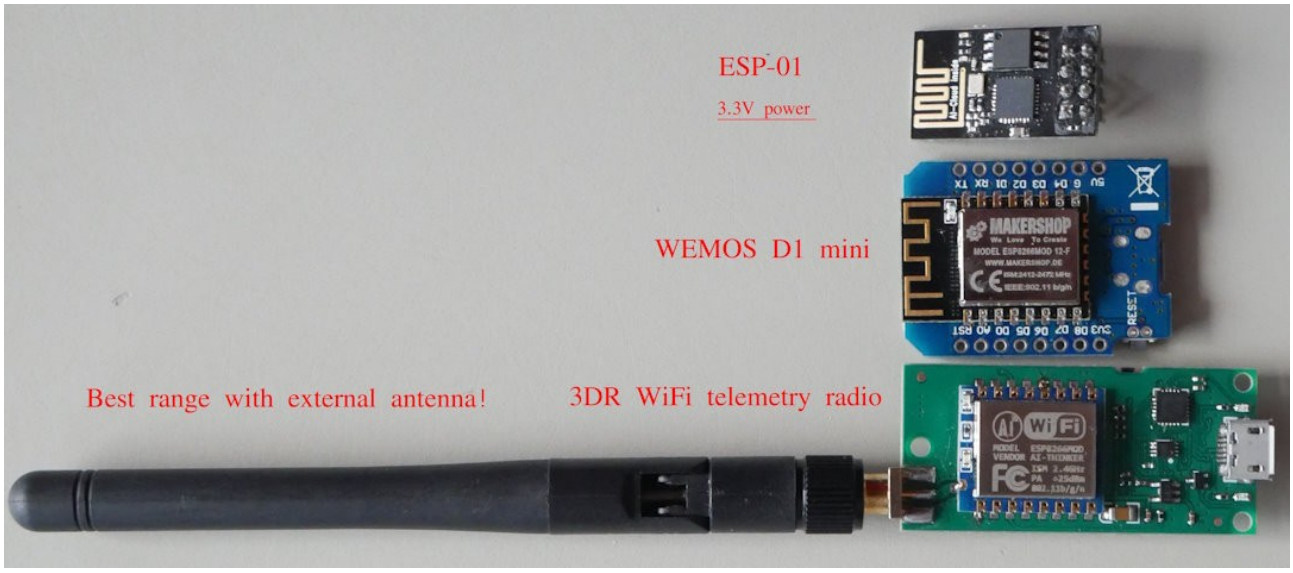


3. Flash ESP8266 module

Check **documentation of the ESP8266 modul** you want to use. Important items are

- pin layout
- power supply (3.3V or 5V or whatever you module can handle)
- signal voltage of the Tx and Rx pins (usually 3.3V)
- how to bring it in flash mode (GPIO0 to ground, automatically or by button).

I have tested this procedure with following modules:



Bring ESP module into flash mode. In ESPflasher, page ESP tool click on Flash-ID. The ESP8266 should respond with some data about it. If so, we can go ahead flashing the firmware file.

Example for ESP-01 module:

```
flash_id
-------


esptool.py v4.6.2
Serial port /dev/ttyUSB0
Connecting....
Detecting chip type... Unsupported detection protocol, switching and trying again...
Connecting...
Detecting chip type... ESP8266
------------------------------
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 60:01:94:1f:07:9d
Uploading stub...
Running stub...
Stub running...
Manufacturer: e0
Device: 4014
Detected flash size: 1MB
Hard resetting via RTS pin...
```

To flash MAVbridge firmware click on Write FW. You will be asked for a firmware file. Select MAVlink_firmware-1.2.2.bin from the bin folder. Once selected the flash procedure starts. Wait until flashing is comleted.
Output shall look like this

```
write_flash: MAVlink_firmware-1.2.2.bin
-------------------------------------


esptool.py v4.6.2
Serial port /dev/ttyUSB0
Connecting...
Detecting chip type... Unsupported detection protocol, switching and trying again...
Connecting...
Detecting chip type... ESP8266
------------------------------
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 60:01:94:1f:07:9d
Stub is already running. No upload is necessary.
Configuring flash size...
Flash will be erased from 0x00000000 to 0x00051fff...
Compressed 332768 bytes to 232984...
Wrote 332768 bytes (232984 compressed) at 0x00000000 in 20.7 seconds (effective 128.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

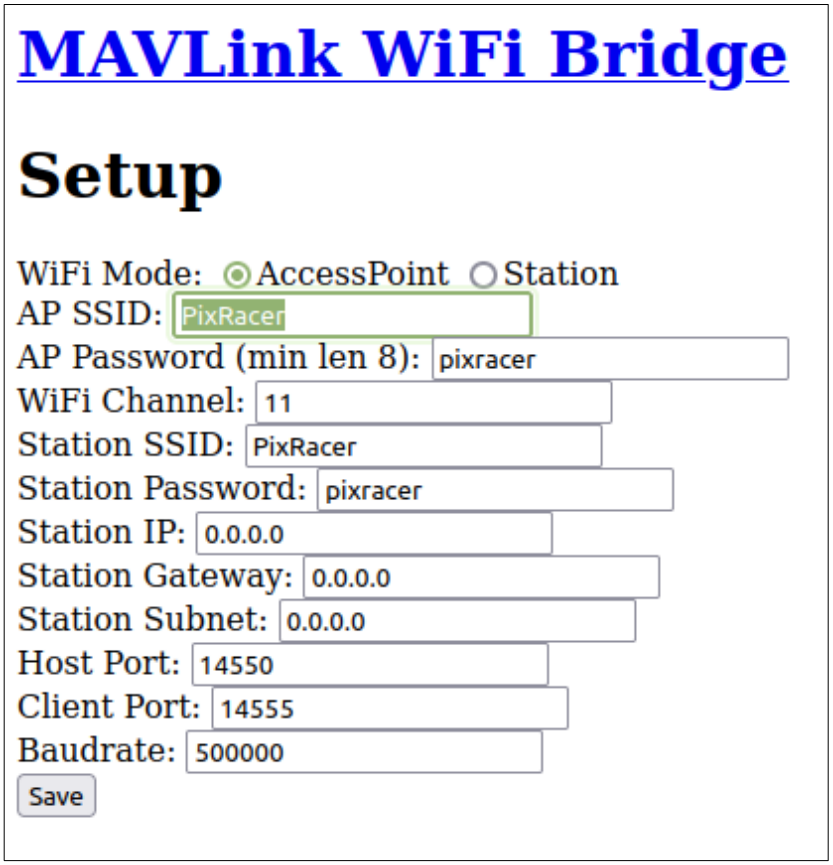
Flashing done. Remove ESP8266 module from USB port. Now this module is a MAVbridge. That means it will transfer MAV link messages between ground station and drone over a 2.4GHz WiFi network. The MAV bridge will open a WiFi access point (WiFi hotspot).

4. Test and setup the MAVbridge

Power on the MAVbridge. Use a phone, tablet or PC to connect to the WiFi access point of the MAVbridge. If the SSID oixracer appears in your WiFi networks list then the flashing was successful. Connect to this network. Defaults settings:
SSID: pixracer
Password: pixracer

Open an internet browser and type 192.168.4.1 in addess bar. You should now get the home page of the build-in webserver of the MAVbridge.

Click to Setup.



Change the AP SSID to a unique name that you will use for further login. Change AP password to a unique password of your own. Change Baudrate to **500000** (the rate the CGO cameras using). Save
In our example we use the SSID **MAVbridge** and the world famous password **1234567890** (not recommended).

5. Hardware setup

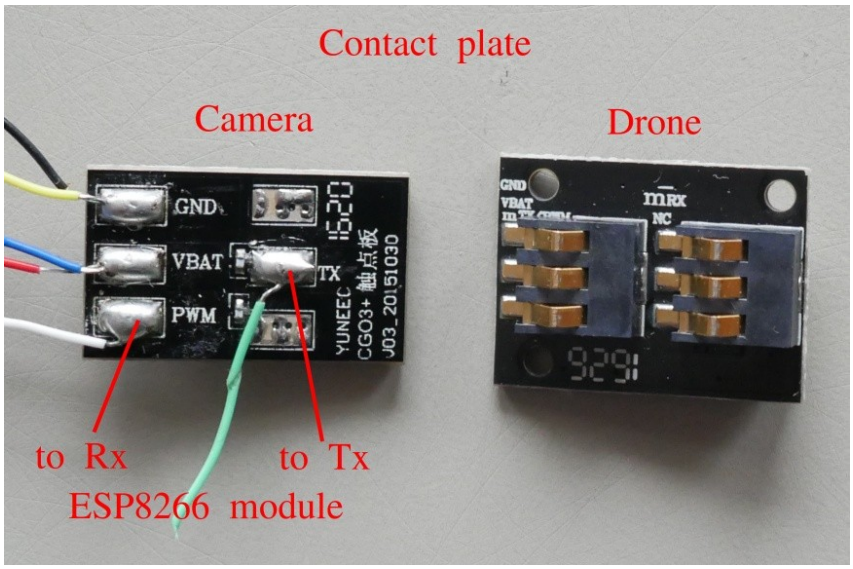
How you build the hardware depends on the material and the use case you have. The examples described here are only to illustrate proposals. You can create your own module, internal or external. In most cases the MAVbridge will be mounted at the camera mounting plate using the contacts there.
You need a step-down voltage converter to 5V or 3.3V depending on the ESP module you use. ESP-01 needs 3.3V, NodeMCU boards usually need 5V.

Note: **Never connect ESP8266 modules to Ubatt!**

3D print:

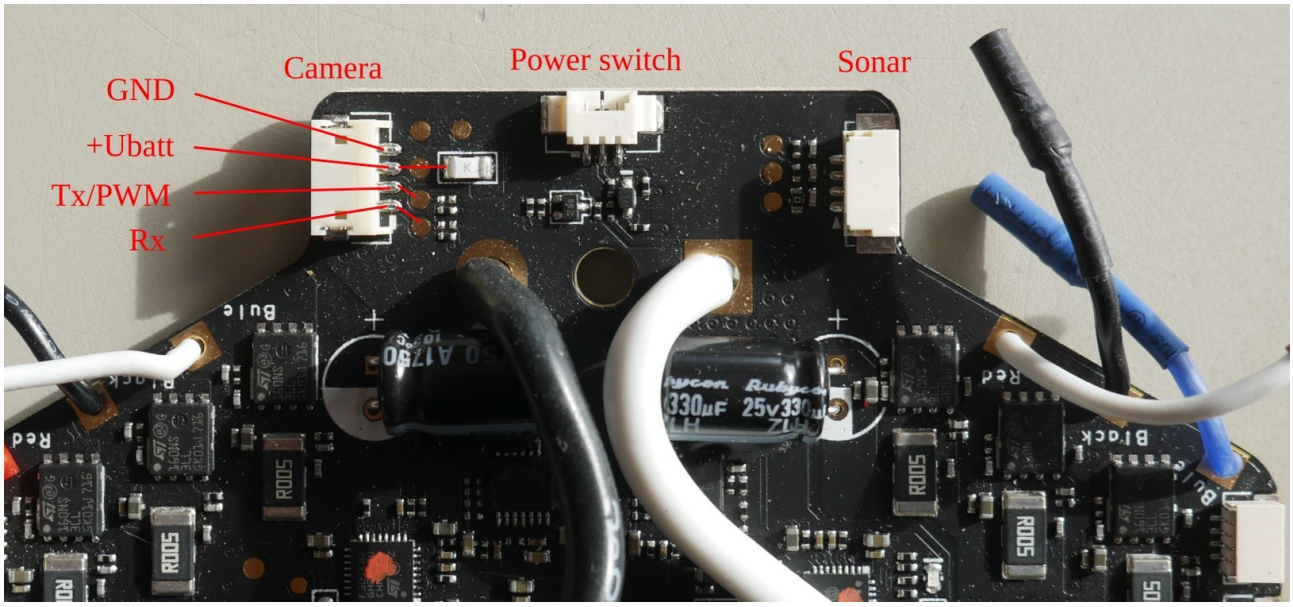
- Camera mounting plate: https://www.thingiverse.com/thing:1663476
- GoPro-style payload holder: https://www.thingiverse.com/thing:4737276
- Housing: to be made by yourself based on camera mounting plate.

Camera mounting contact plate:



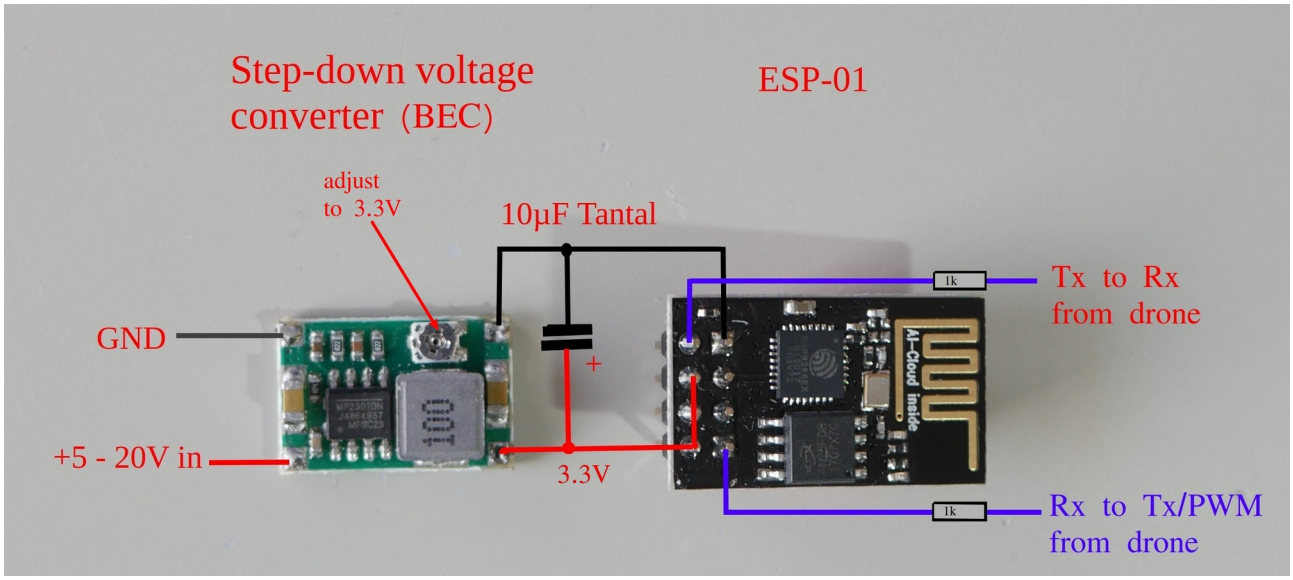
Note:
Droine side print indicate Rx/Tx from drone. The counter part print indicates Rx/Tx from camera/NodeMCU.

Internal camera connections (Example: Typhoon H Plus main board):



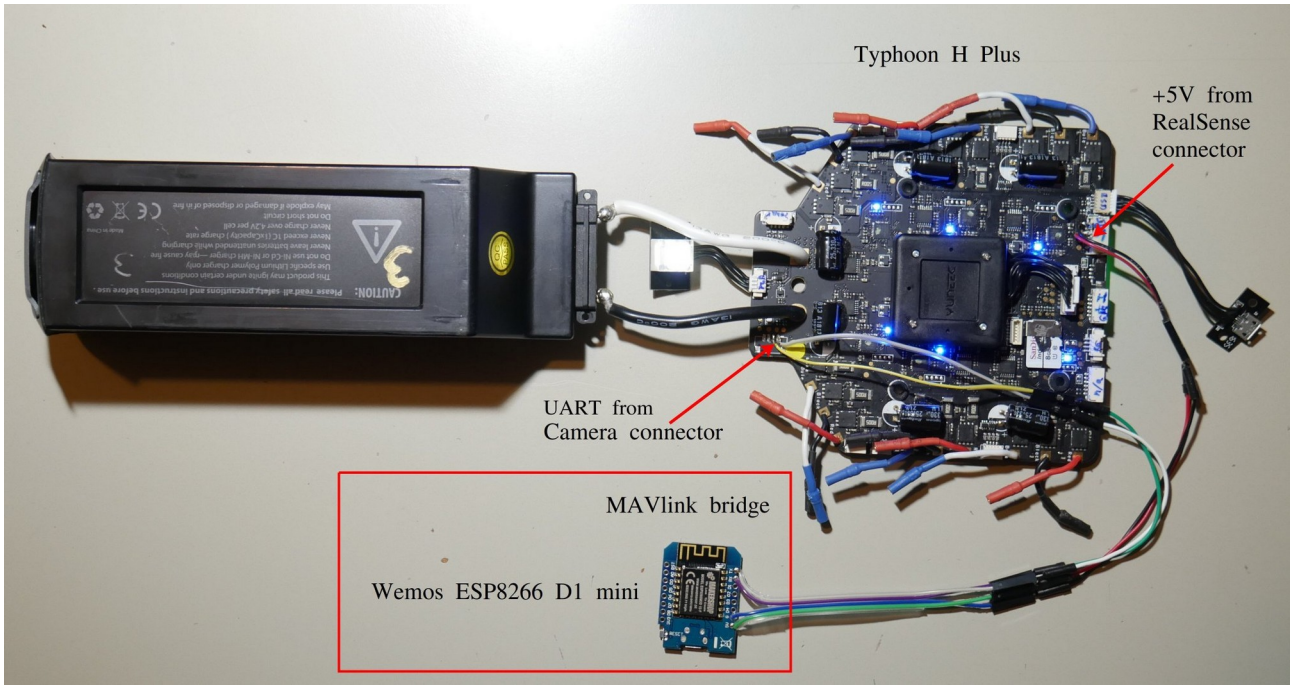
5.1 ESP-01

Vcc is 3.3V. GPIO0 to GND and power on brings it in flashing mode. For normal work GPIO pins remains empty.



5.2 D1 mini NodeMCU

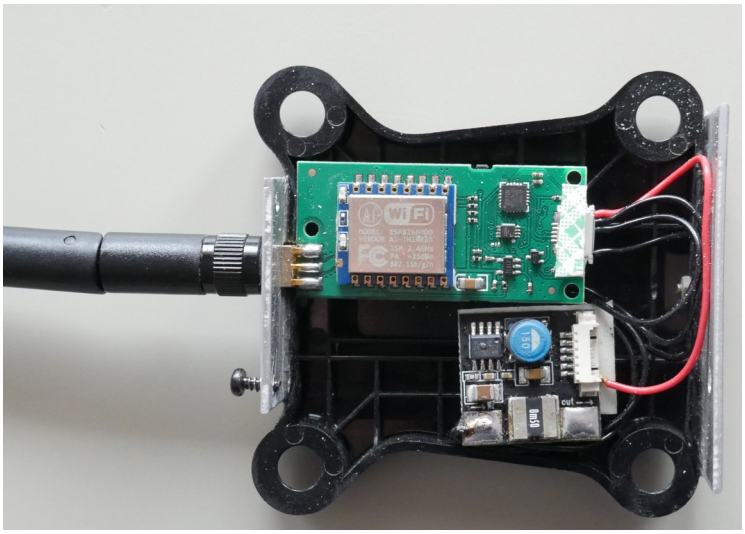
Vcc is 5V. For internal use the 5V can be taken from RealSense connector. Rx/Tx are available on camera connector.



5.3 3DR WiFi telemetry radio (with antenna)

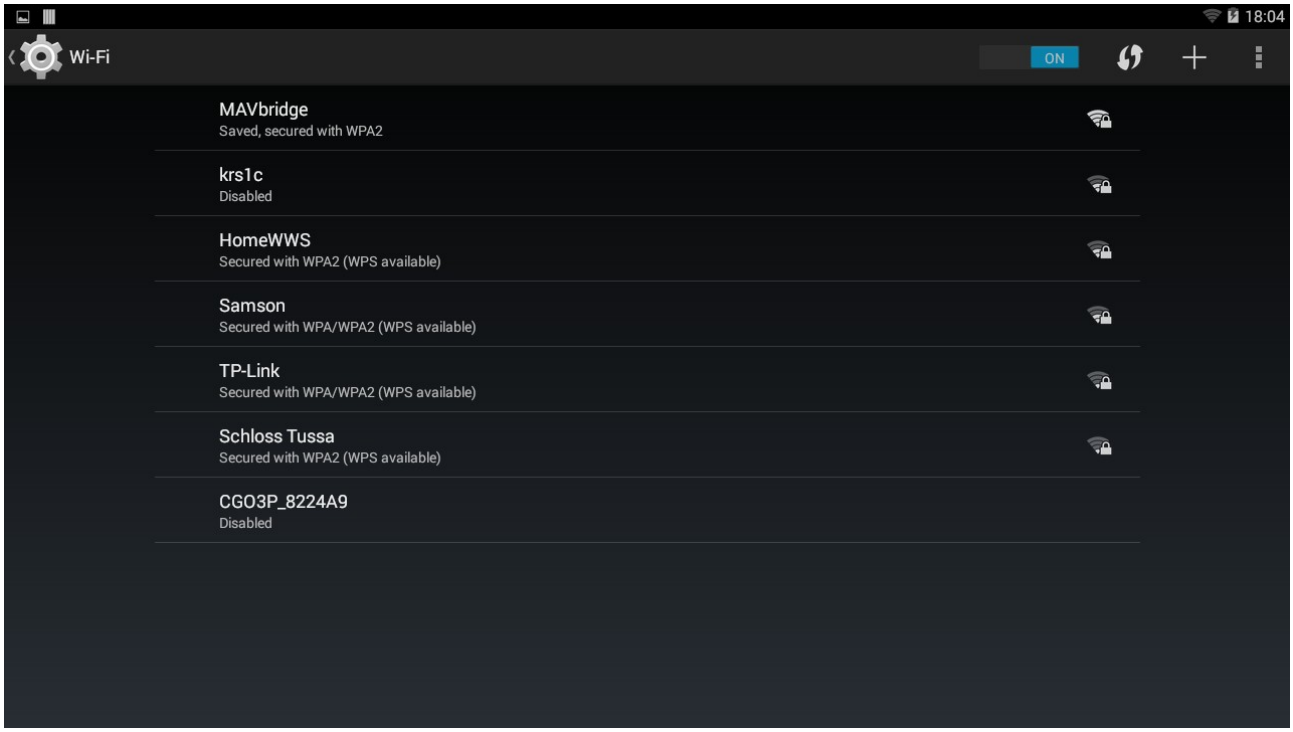
A BEC from model shop provides 5V for the WiFi module. The rod antenna has better range compared to the internal antenna. Of course there a better 2.4GHz WiFi antennas on the market.

The antenna connector on ESP8266 MCUmodule is the same as used for the ST16 antennas (and many others). Please keep in ind you cannot use the 5GHz mushroom antenna for the MCU board. It uses **2.4GHz** WiFi band. You need an antenna for 2.4GHz if you want to have a better one.

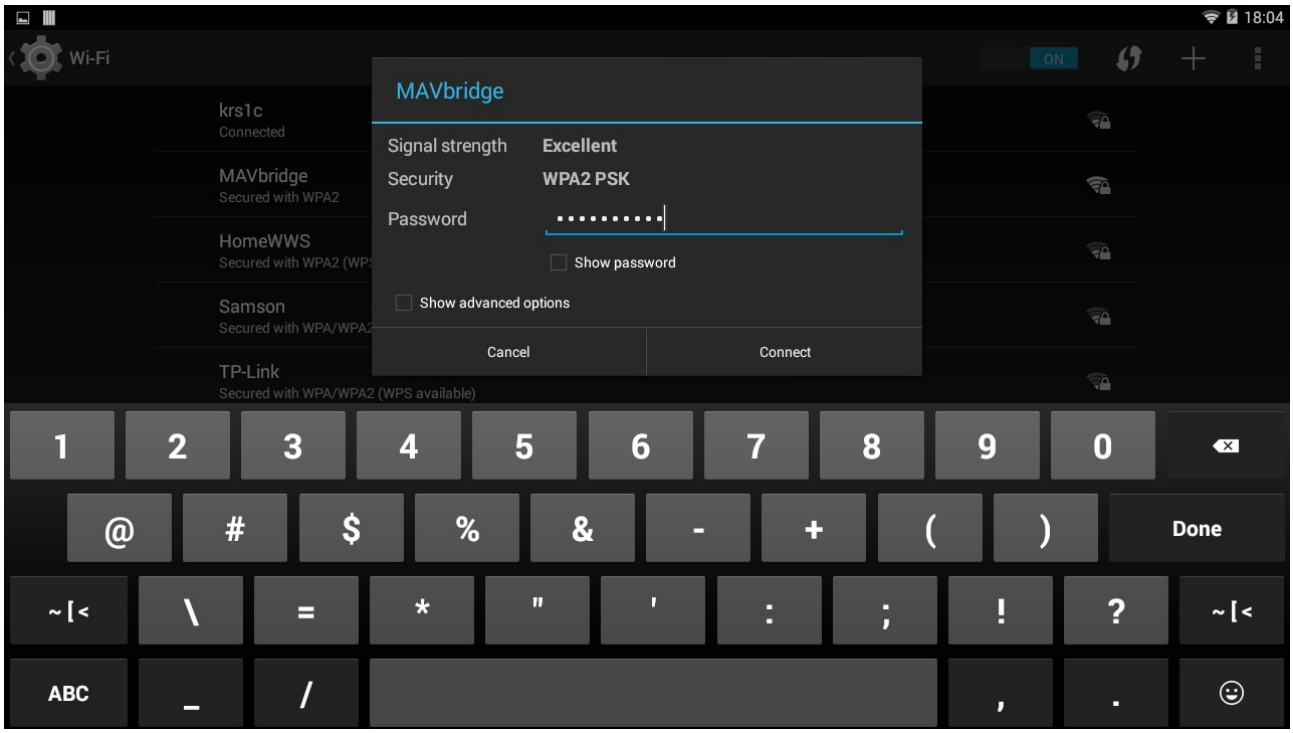


6. Connect drone with MAVbridge to ground statioon ST16

Power on ST16S, exit flight mode app and go to PAD mode > Settings > WiFi.

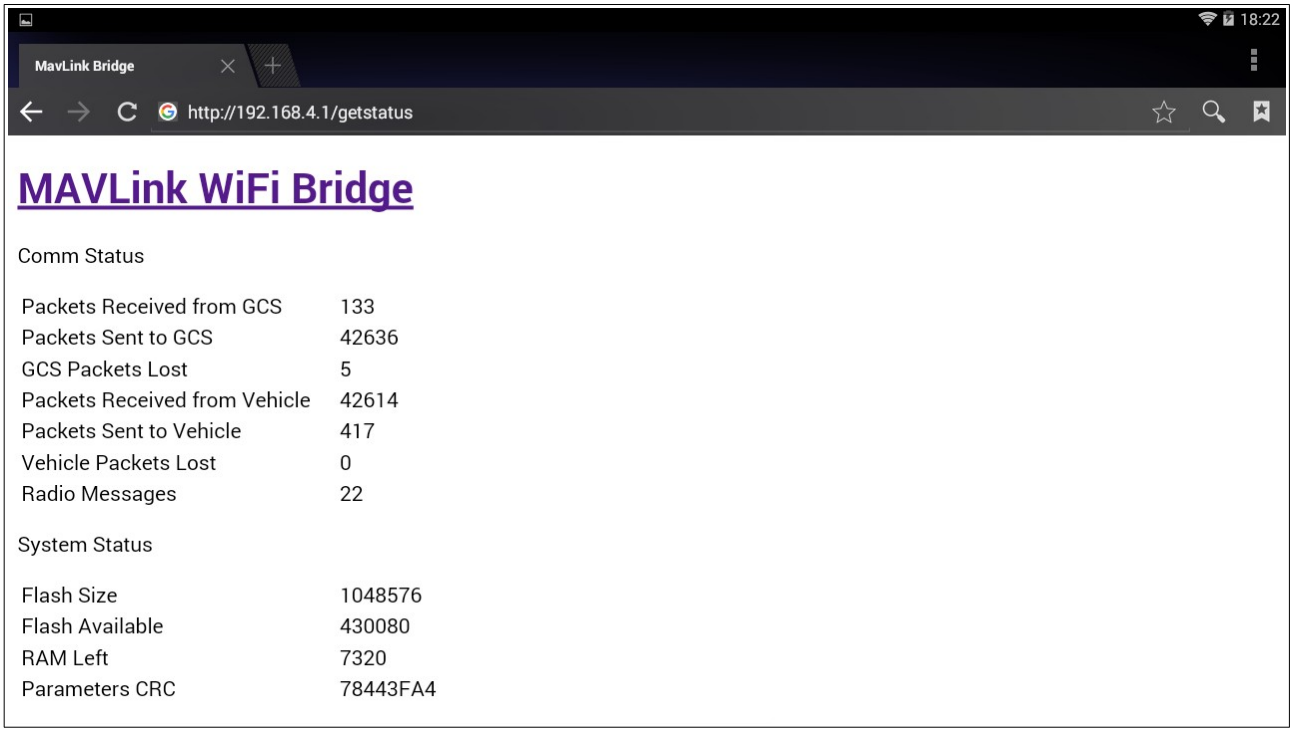


Select ***MAVbridge*** from network list (list of SSIDs).



Enter the above password. Click on Connect. Go back to flight mode screen. Now we will see telemetry data from the drone. Done!

Also on ST16, you can access to the build-in web server. have a look on the Status page. Here you see among other things statistics about data transfer between drone and ground station.



Have fun and fly safe.