Pixhawk Serial Communication

Pixhawk uses serial communication to transmit data with any types of computer/microprocessor. The communication protocol is MAVLink (Micro Air Vehicle Link). As long as we want to get/send data with Pixhawk, there must be some libraries/source code involving MAVLink.

Pixhawk has two serial ports, microUSB and Telem2 (Figure 1). Generally speaking, microUSB is used to connect PC, and Telem2 connects to microprocessors.

Figure 1

PC Connection

In order to connect Pixhawk to PC, a Ground Control Station (GCS, a type of software), which has MAVLink decoder inside, needs to be installed (because we need to go through MAVLink). Some Ground Control Station include:

- Mission Planner
- APM Planner
- QGroundControl

They are compatible with Windows and Mac. These softwares offer visual representation of the vehicle's status (Figure 2), but their functions are limited because there is not a full-control terminal where we can give commands. It auto generates code based on the pre-set vehicle configurations. This is important because if we have a custom configuration, we want to write scripts ourselves to get the sensor reading and control the vehicle.

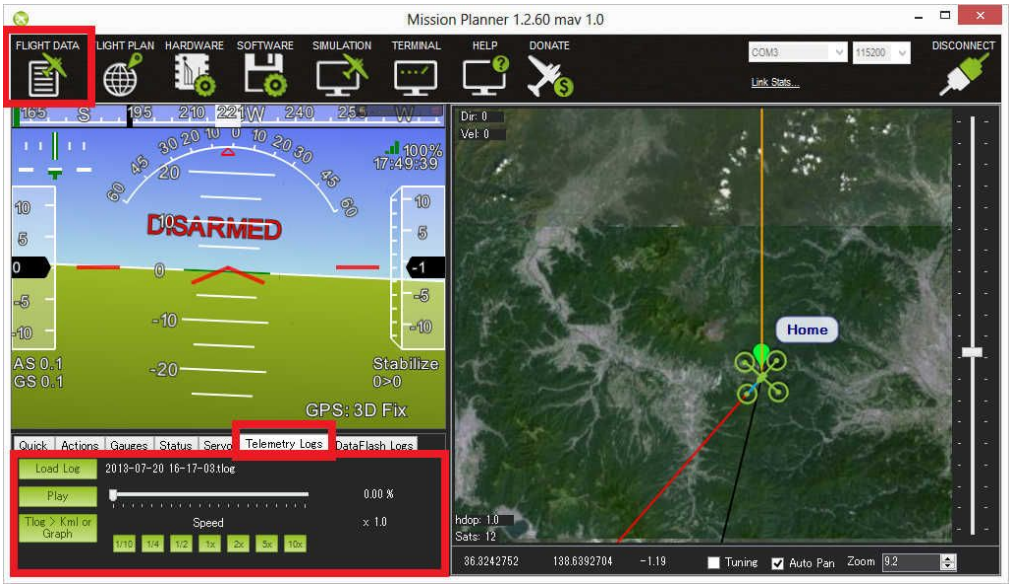


Figure 2 - Mission Planner

There are three other software, however, that gives full-control terminal.

1. First one is MATLAB, which Madhavan has established some code on.
2. Another one is MAVProxy, which is compatible with Windows and Linux.
3. Lastly, DroneKit module for Python 2.7. With this module, we can also write Python scripts that interface with Pixhawk.

Microprocessor Connection

Since MAVProxy and DroneKit are both available for Linux, these two methods also help connection with Raspberry Pi and other Linux-based microprocessors.

Another method is Arduino. There is MAVLink library available to decode the serial data (example). One drawback of this is, unlike MAVProxy or DroneKit that parse the data for us, we have to parse the data yourself in the source code. If the data is parsed in a wrong way, we are not able to decode the data at all.

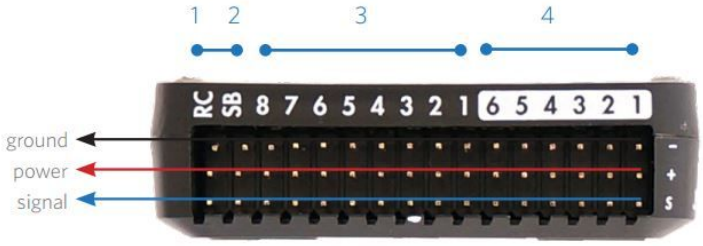
Motor Control

Pixhawk has 8 terminals that sends PWM signal to ESC in order to control the motors (Main Output in Figure 3). There are mainly two modes of control, remote control (RC) and off-board control. Remote control is essential manual control with RC transmitter (Figure 4); off-board control is user giving command from a computer/microprocessor.

An RC transmitter usually gives more than one channel of PWM signal, in order to input into Pixhawk, a PPM encoder (Figure 5) is used. PPM is short for pulse position modulation. The PPM encoder connects to RC terminal in Figure 3, and essentially integrates multiple channels of PWM into one. Pixhawk is able to understand this PPM signal, so it translates back to multiple channels of PWM and output to ESC.

Since PPM comes pretty useful in transferring signals, off-board control also uses it (though PWM is also doable). There are two methods of sending PPM in off-board control.

- Via MAVLink, send PPM-related message to Pixhawk
- Output PWM signal to PPM encoder, and connect PPM encoder to RC terminal, similar to remote control.



1 Radio control receiver input
2 S.Bus output
3 Main outputs
4 Auxiliary outputs

Figure 3 - Pixhawk Motor Control

Figure 4 - RC Transmitter

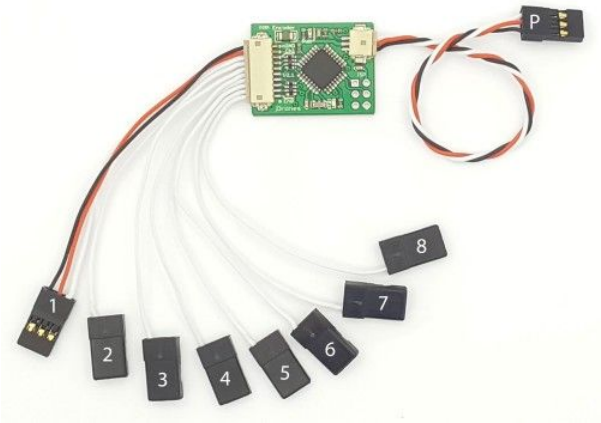


Figure 5 - PPM Encoder

Suggested Connection Network

Figure 6 shows my suggested connection network. Remote control mostly take place in controlling the vehicle, while the microprocessor gets sensor data and takes over the control when in danger (eg, avoid obstacle). The PC on the right have GCS does not have to give command, but offers visual of the status of the vehicle.

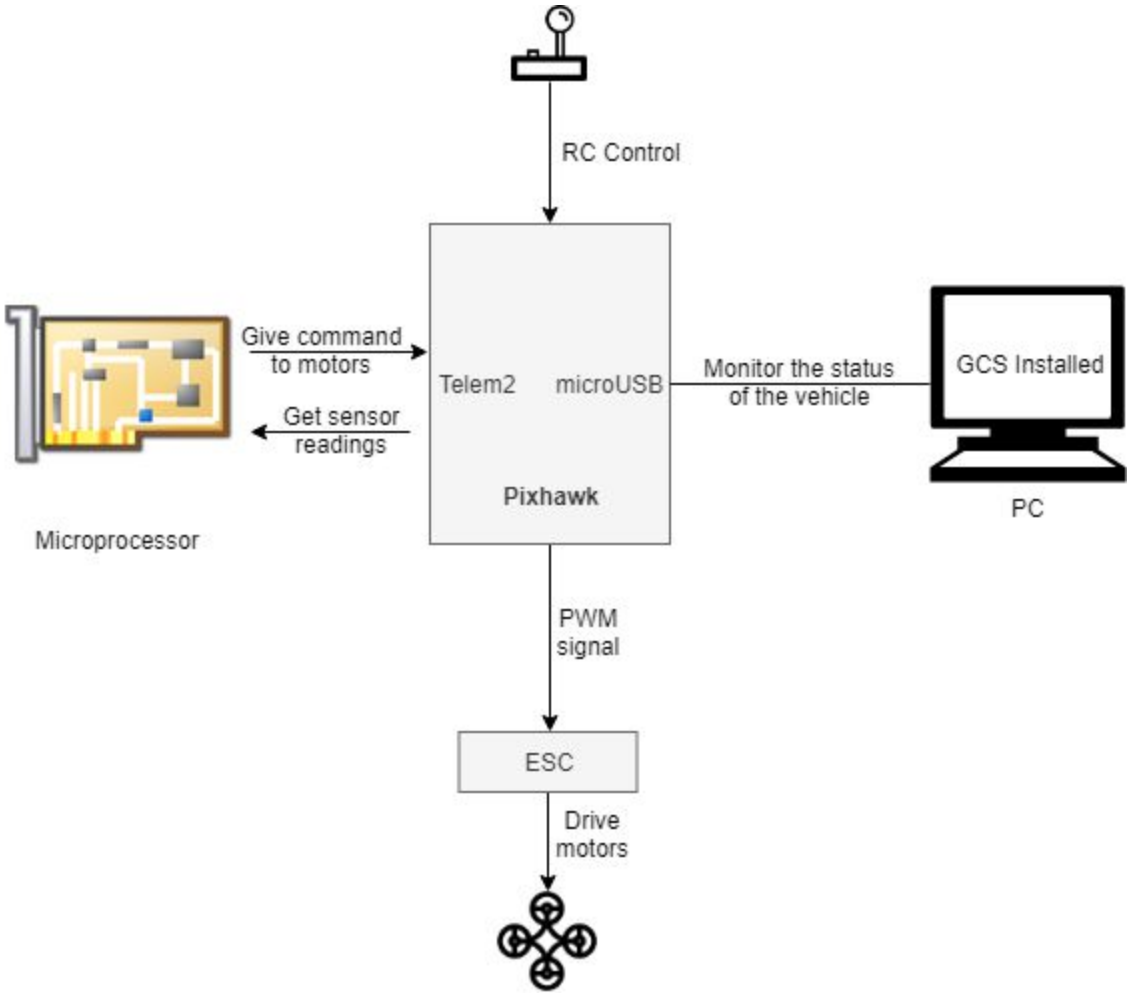


Figure 6 - Suggested Network

There is another connection network that is less complex (Figure 7), where the PC is responsible to get sensor readings, give command and meanwhile, give the user a visual demonstration of the status. The PC has the capability for these tasks if we use MATLAB or MAVProxy. However, since this PC is heavily used in these many tasks, the drawback of this connection is that the operation is very software-reliant.

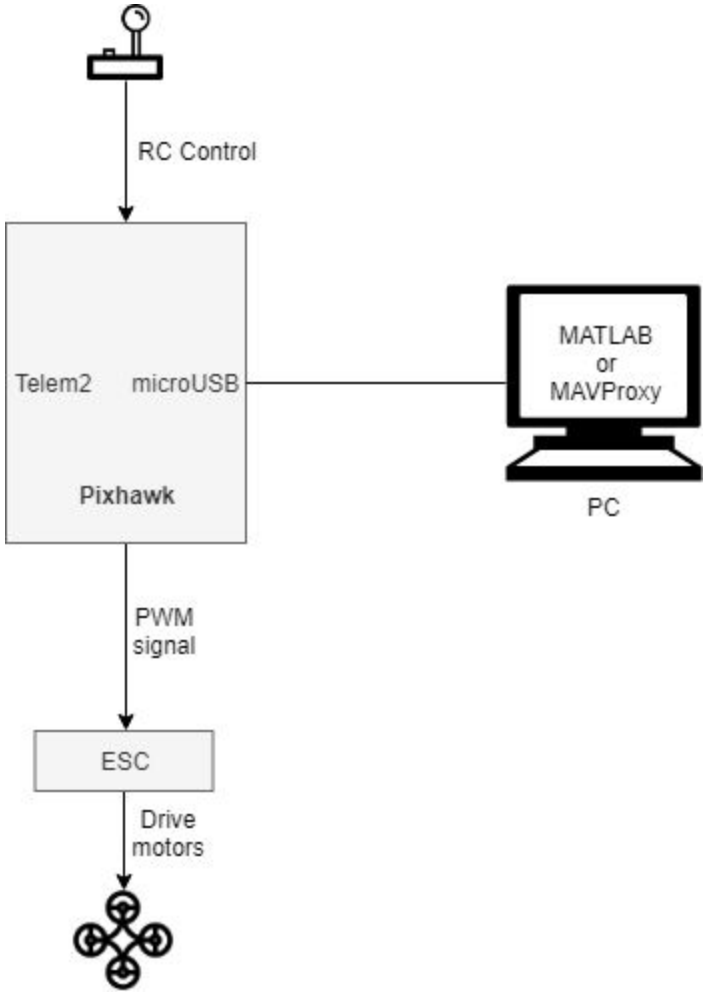


Figure 7 - Less Complex Network