Project Journal

30/01/20

* Learned to solder, attached the speed controllers to the power distribution board
* Attached battery connector to PDB
* Attached frame to PDB
* Attached motors to frame
* Attached the Navio shield to the Raspberry pi and attached this to the top plate
* Attached the top plate to the frame

02/02/20

* Paired the Receiver with the transmitter
* Calibrated the Speed Controllers
* Corrected the Motor spin direction
* Attached the PPM encoder and connected the Navio board
* BROKE THE FUCKING NAVIO KILL ME NOW

07/02/20

* Dad saved the day by finding a Navio in stock within the UK, brilliant!
* Successfully connected the telemetry
* Installed the power module
* Attached the battery to the drone
* Built the GPS stand and attached the GPS module to the drone

08/02/20

* Configured the Navio2 – flashed the micro SD card with Raspbian + ardupilot software
* Did an SSH into the Raspberry and made some further configurations to the ardupilot software – selected version of ardupilot, selected type of vehicle, specified which port telemetry was connected to.
* Downloaded Mission Planner (ground control software, may use instead of QGroundControl, will have to compare and contrast)

09/02/20

* Calibrated the accelerometer, compass, gyroscope, RC transmitter and GPS module
* Troubleshooting – fixed pre-arm problems by setting all unused radio frequencies to a range between the minimum and maximum
* Successfully armed drone and performed a small test flight – the drone can fly with an RC transmitter
* Mention that the parts required took a long time to arrive – related to different parts coming from different parts of the world.
* Mention that one of those parts broke and needed to be replaced (got lucky, new supplier which shipped from the UK)
* IF EVERYTHING GOES TITS UP:
  + Present the set of tests you WOULD have set up, even if you can’t present the actual results themselves.
  + Show the test plan, show what you would have compared, show the expected results, show what the results would have meant if you could have compared the two
  + Show the test data based around the control
  + If it can’t work, then the technical complexity can be discussed and the REASONS WHY it didn’t work can also be discussed
  + Could mention the hardware difficulties, Linux architecture, Linux scheduling policies
  + Talk about setting up a virtual environment for testing purposes to help prevent a worst-case scenario from occurring
  + Talk about how the components are connected and how the power is distributed
  + Talk about how performance metrics are difficult to analyse, but mention what you DID analyse and compare between tests
  + Look at analysing the problem, and how you tried to solve the problem – don’t obsess over the actual prototype that you come up with
* Tell the story so far! Don’t worry about whether it’s working or not, just go through the process and talk about what is going on.
* Describe the DESIGN of the drone itself, the Linux environment that the flight controller will operate in, the test modules which will be used to monitor both the drone itself and the power consumption of the Raspberry Pi (flight controller) and finally the test case itself which will be used to monitor these metrics
* ‘Remind’ the reader of what the whole point of this project is. Go back over the problem statement and relate everything back.
* The next step will be evaluating and discussing what the results mean. Worry about this later.
* When analysing the results themselves DO NOT make massive claims – talk about what the results COULD mean and what future work needs to be done in order to get a larger view of what this line of research could mean. E.g. the environment which this experiment worked in is very specific – it won’t necessarily work with every MAV configuration, especially considering how modular/customisable Linux as an OS is.