User and Installation Manual

Desktop Enviroment setup

https://platformio.org/
It is recomended that you use PlatformIO for the development and deployment of software onto the microcontroller. The entire software developemnt process has used PlatformIO extensivly and is known to work well. PlatformIO can be used either through command line, or graphicially through third party IDE's. We did all of our software development in Visual Studio Code, which has a nice graphical extenstion for PlatformIO. Although using the Atom IDE is also an option that works well for platformIO.

Using Visual Studio Code

https://code.visualstudio.com/
Open up the directory that the `platformio.ini` file is located. If PlatformIO is installed, you should be displayed with a range of options for managing the software package. The main icons you will use ate the tick, for compiling the software, the arrow, for uploading the code to the board, and the plug, for connecting to the board over serial.

Launching

The following will describe the different states that the avionics package goes through throughout the duration of the launch

Timeline

1. **Power on**
   Package goes through each different hardware component enabled on the board and runs the initialisation code. This ensures that each component is physically working and able to communicate.
2. **Startup State**
   The package attempts to use each component as intended and confirms that each module is working as expected. The IMU is checked to report valid data. The GPS module is checked to have received a lock. The SD card is tested that it can write successfully. The RF module is tested that it can both send and receive information This state will only move on once all components have been verified.
3. **Idle State** This state is yet to be properly implemented, and currently just passes straight onto the prelaunch state. When implemented, this state waits for a wireless command from the base station to let the rocket know that it is in the launch position and ready to launch. Without any other interruptions to be expected. This state exists so that the rocket does not think it has launched after and sudden movements/being dropped while handling.
4. **Prelaunch State**
   In this state the package is effectively in idle waiting for the rocket to take off. The package continues to log both IMU and GPS data in this state. For this state to move one, at least one of the accelerator values must exceed 3gs of force.
5. **Launch State**
   During the launch state it is expected that the rocket is under power. The control code with be activated and will be actively controlling the gimbal. During this time both IMU and GPS data will be logged to

the SD card but there will be no wireless transmissions until the burn has finished. This is due to the need for the control code to run without blocking, and an rf transmission can take up to 150 milliseconds. In order for the package to move to the next state, all accelerometer values must be below 1.2gs

6. **Postlaunch State**
   This state is active after the engine has burnt out. So when it is in freefall and when it has hit the ground. During this state both IMU and gps data is being logged to the SD card, but the GPS data is also being broadcasted

Rocket Assembly

1. **Program Board** Program board and set up communication between Avionics Package and base station:

Ensure that PlatformIO CLI is installed and added to the PATH. If you can run the `platformio` or `pio` commands in the terminal then this is already done. Note that these two commands are interchangable. Navigate to the directory that contains the `platformio.ini` file. This is to let PlatformIO know which project it will be working with. In order to compile the package run `platformio run`, in order to upload to the board `platformio run --target upload` and in order to connect to the board over the serial connection `platformio device monitor`

2. **Power Electronics** Connect gimbal servos and battery to board.
3. **Attach Board** The H-shaped bottom PCB mount should be inserted into the main tube. Once past the first internal wall extrusions, rotate the board 90 degrees and continue insertion until the top PCB mount is flush with the top of the main tube, while ensuring wires are not snagged.
4. **Attach Nosecone** Ensure a sufficient amount of the provided foam is placed inside the cone to protect the battery.
5. **Pre-Launch Checks** Inspect the rocket airframe to ensure there are no gaps or loose parts. The gimbal must be oriented in a centred position, i.e. straight down prior to placing the rocket on a launch rail.
6. **Launch Set-up** This steps involves the placement of a D-E class rocket motor and ignition charge.
7. **Ready to launch**