Progress Review 5
Project Pegasus

Tushar Agrawal

Team A – Avengers Ultron
Teammates: Pratik Chatrath, Sean Bryan

ILR #6
January 28, 2015

1. Individual Progress

Odroid Backup

As our current SBC subsystem was working well, we decided to have another Odroid XU4 ready and set up with up an identical Ubuntu image. I was responsible to set it up for this PR.

We had ordered another Odroid XU4 before the winter break and had received it when Spring semester began. I tested the new ordroid with the old SD card image which was setup to our needs.

Next. I took the backup of the original image (Ubuntu 14.04 with OpenCV and ROS) and cloned in onto another card. Then, I tested the new Odroid with the new SD card. The board booted up perfectly and ran the Apriltag detection program. This concluded the setup. (refer Image 1)

To run the system offboard, I connected the Castle BEC to the gimbal power line (given by x8+), and used its output to power the Odroid.

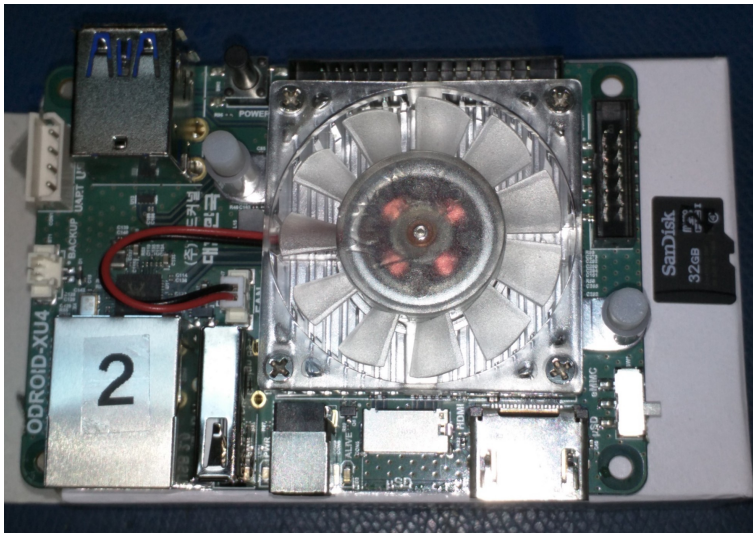

Image 1: Image 1: Backup Odroid and a memory card with Ubuntu image. A label with "2" marks the backup

Setup communication between Odroid and Pixhawk

For controlling the route of the UAV using the Odroid, we would need to setup the communication between the Odroid and Pixhawk. To do that, I analysed the 3 possible connection options at the pixhawk side:

1. USB: USB connection is not robust as the USB needs to be connected within the first few seconds after powering up the pixhawk, and hence, is not recommended.
2. Use TELEM2 port: TELEM2 port was being used by the Mission Planner Radio, which would have to be removed.
3. Use Serial 4 port: Serial 4 is free and configurable for any connection.

I initially tried setting up the Serial4 port so we can have the Mission Planner running.

But I hit various roadblocks. The Serial4 port does not send MAVLINK data (flight information data through MAVLink protocol) by default. Even after setting up the required parameters or executing the commands through the shell, the port did not start publishing the flight information. After looking at various online resources and understanding that Rohan of Team E was also facing the same problem without a resolution, I decided to switch to the TELEM2 port, which worked immediately without any problems.

I set up mavros, which is a ros node which communicates via MAVlink to the pixhawk and gives a framework for structured communication. Mavros could easily receive all flight related updates and even send simple commands like mode switches to the pixhawk (refer Image 2).

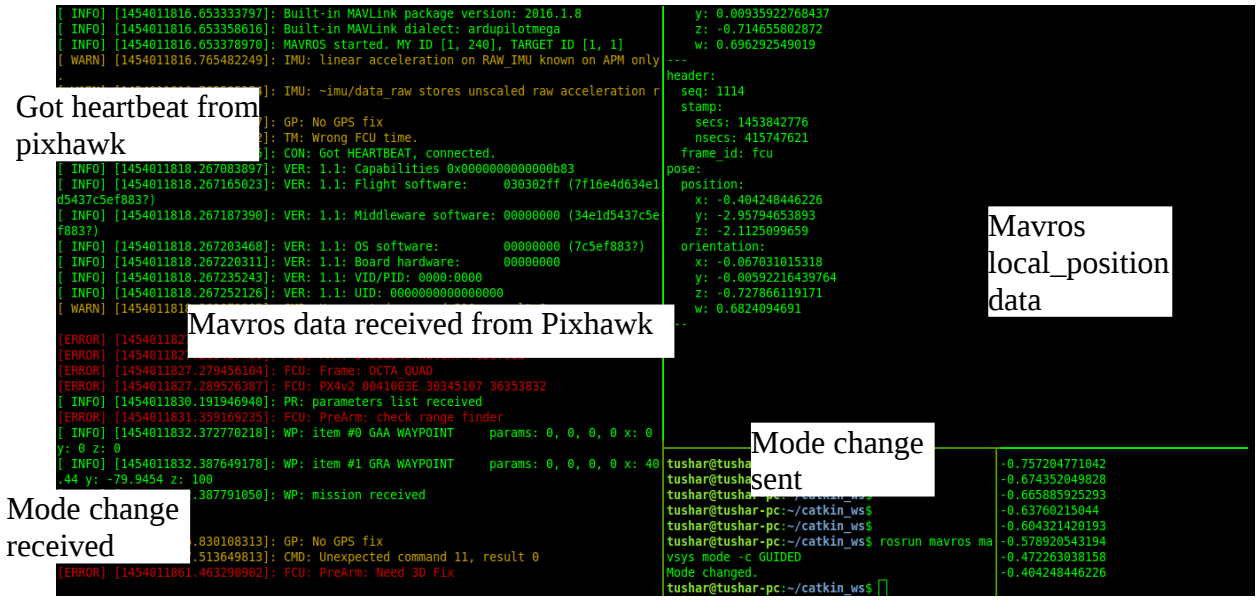

Image 2: Mavros updates from pixhawk and commands sent to the pixhawk (on laptop)

Until then, I was using the Laptop to finalize exactly what framework to setup. Once finalised, I set up the Odroid with mavros and tested its communciation with the pixhawk. It worked seamlessly as expected (refer Image 3).

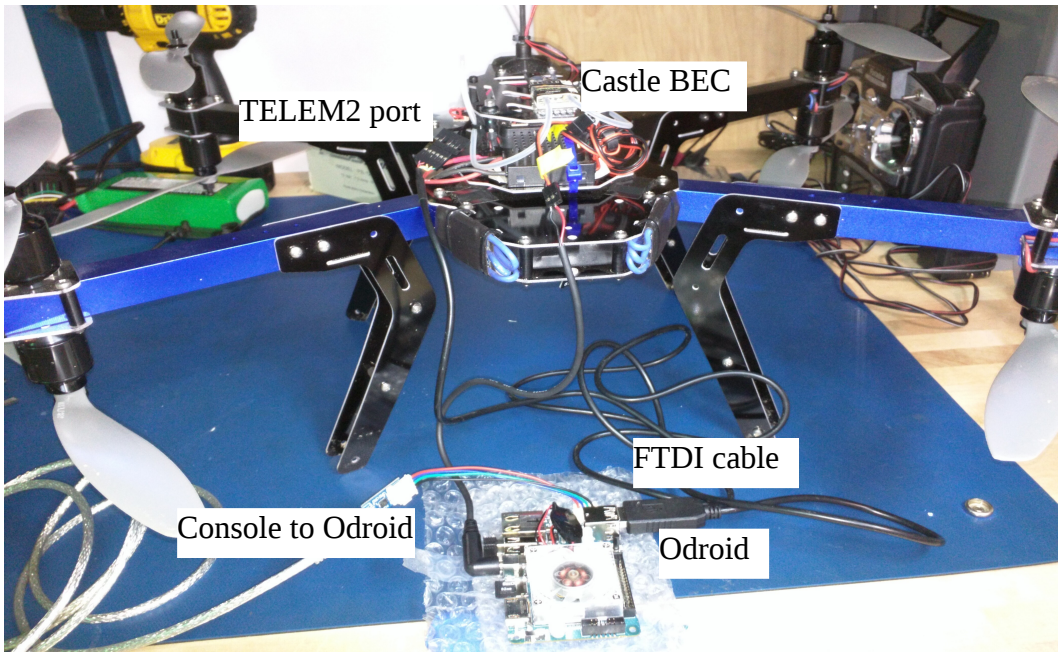

Image 3: Connections between Odroid and Pixhawk

Setup Lidar lite as range sensor (for altitude measurement)

As there was some error in holding the altitude, I decided to include the Lidar lite rangefinder to make the altitude measurement robust. I made the required cable to connect it to the pixhawk using the PWM connectors. I mounted it temporarily to the UAV and tried flying it in altitude hold mode. The flight was a lot more stable and robust. The flight demonstration shown as part of the PR was done in altitude hold mode with the Lidar lite attached.

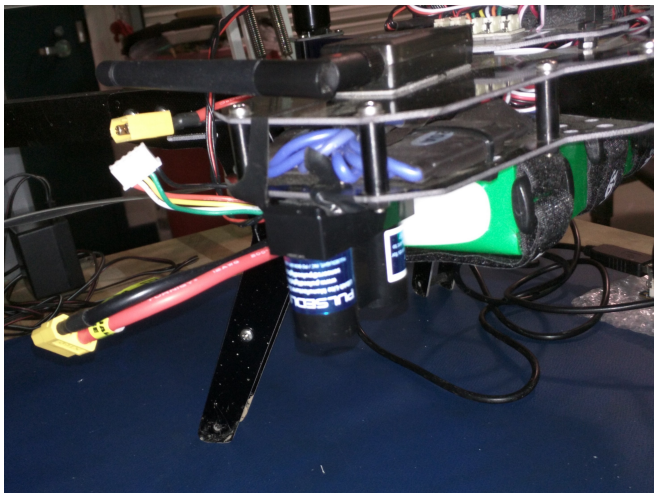
Image 4: Lidar lite rangefinder connected to the pixhawk, and attached to the UAV

2. Challenges

I faced major challenges while establishing communication between between pixhawk and the odroid. The initial method involved using the secondary interface, SERIAL4 for communicating to the Ordoid. The documentation for this was rather preliminary and informal. The x8+ shipped with the APM firmware, whose documentation was worse than the PX4 firmware used otherwise. All methods I could find were implemented and tested, but the flight information was not published over Serial4. Certain direct commands were sent through the NuttyShell to enable the interface, which too did not work.

I worked with Rohan from Team E to see if they had got the same working and realised they were facing the same issues. I finally decided to revert to the backup option of removing the Mission Planner Radio from TELEM2 port and using that to communicate with the UAV, which worked without any issues.

3. Teamwork

The team morale had been down after hearing Adam's decision to drop-out. After many critical discussions in the last week before the winter break, we finally decided to move onto a different UAV platform hoping to reduce a few problems. We all met up the first day of spring semester, picked up the new packages, and started assembling the UAV. The first version flew very well indoors, showing good promise. This motivated us to get started towards our application goals as soon as possible. Soon, the nicadrone was tested electronically and seemed to be satisfactory from initial tests. The UAV waypoint following mode was also confirmed to be working well. With a good initial backing we started our modules in depth.

Sean was responsible for the mechanical integration of components into the UAV. He made several prototypes to finalize the positioning and orientation of all sensors so they do not interfere with each other or the package.

In addition to changing the UAV platform, we were also looking forward to changing the obstacle detection sensors from ultrasonics to a Lidar as ultrasonics were not giving reliable data up to the end of the semester. Pratik started with setting that up with the navigation stack which we were hoping to use for our obstacle avoidance and navigation. I assisted him in a few software related issues which he was stuck with, but eventually had a good initial place with the entire stack set up and outputting control

velocities.

4. Future Work

Based on our current plan, for the next 2 weeks, I shall be working on getting offboard control entirely set up for the pixhawk. I shall start with the laptop and if possible, set it up onto the odroid as well. This would enable us to send control parameters (positions or velocities) to the pixhawk to control its behavior outside of the pixhawk.

Sean will mechanically mount all underbelly components onto the UAV and design and apply modifications as required for the package for being lifted and carried by the UAV.

Pratik will continue with setting up the navigation stack, now to get real odometry and obstacle data (possibly in simulation) and moving/planing accordingly. Once that is completed, if time permits, he will move onto see how to modify the mavros data received to odometry data format required by the navigation stack.