

ILR07: Progress Review 8

Maitreya Naik

Team B: Arcus

Teammates: Logan Wan, Clare Cui, Angad Sidhu

ILR07

Feb. 15, 2016

1. Individual Progress

Since the last progress review, I was responsible for the PixRacer [1] integration with MAVROS [2], sourcing the MAVROS package into the project repository and testing out a demo code for verifying communication link with the drone.

1.1. PixRacer Integration

Similar to the custom PixRacer cable made in the last Progress Review, I had to make a Pixracer to Telemetry radio module cable for wireless data recording on ground station. The details of the cable and its connections are given in Table 1. To revise, this data includes the IMU readings, compass readings, battery voltage readings, and current readings. Upon referring to the hardware setup instructions at [3] and [4], a cable can be made with connections shown in Table 1.

TELEM2		FTDi	
1	+5v(RED)		NOT CONNECTED
2	Tx(out)	3	RX(Yellow) (in)
3	Rx(in)	4	TX(Orange) (out)
4	CTS(in)	8	RTS(Green) (out)
5	RTS(out)	9	CTS(Brown) (in)
6	GND	2	GND (black)

Table 1: PixRacer USB Integration [3]

After getting the PixRacer hooked up to all the sensors and communication modules, as well as testing for power, I calibrated the PixRacer sensors and assigned an RC controller switch for switching between manual control and “Offboard” control [5].

The next step was to source and build the MAVROS and MAVLink libraries into my branch of the project repository, and debug any issues that were encountered. This involved installing a few more ROS packages and dependencies.

1.2. Link Verification

A sample code is available on [6], the verification was carried out in 2 stages:

Stage1: Software

- a. First, the PixRacer was connected to the computer through the micro-USB port to be monitored in QGroundControl. The ESCs aren't powered at this point and the propellers are removed to avoid mishaps.
- b. A MAVLink launch file (under mavros->mavros->launch->px4.launch) [7] establishes the connection between ROS and our PixRacer MAVLink hardware. Next, the demo node is run affecting the TARGET thrust and orientation rates set on the PixRacer. The demo code instructs the PixRacer to switch to Offboard mode, and arm the motors, then

proceed to achieve a target altitude of 2 metres. The code was modified to get an altitude of 10metres for a visible change in TARGET values on the PixRacer.

- c. In QGroundControl->Widgets->Analyze, we scroll down to the TARGET thrust parameter. If observed before running the demo code, the thrust value should be 0. Upon running the demo code node and upon successful communication, this thrust should increase to a non-zero value. Again stopping the demo code node should reset the thrust to 0.

Stage2: System (Video available at: https://youtu.be/V_MwvGUzi2I)

- a. After Stage 1 is successful, and the PixRacer is mounted and calibrated on the drone, we power the ESCs but still don't put on the propellers.
- b. The demo code is changed such that we remove the code from forcing the controller into "Offboard" mode. Instead, the demo code waits for the drone to switch to "Offboard" mode. Upon switching into "Offboard" mode, the drone arms the ESCs and gives the target altitude of 10 metres.
- c. We assign an RC channel to switch "Offboard" mode ON/OFF. This can be done by going into QGroundControl-><Cogs Symbol>->Parameters->RC_MAP_OFFB_SW and assigning a channel value linked to one of the RC switches. This will allow us to use the RC switch to switch between RC control and offboard control.
- d. To verify, we switch on the power to the drone, verifying that the RC switch is set for manual control. Then, we connect the USART to USB cable between the Pixracer and computer intended to be used for MAVLink.
- e. We follow step (b) from Stage 1 and start the demo code node.
- f. Then we flip the RC switch to switch to "Offboard" mode. Immediately, we notice on our computer terminal that the demo code node detects "Offboard" mode, arms the motors and sets parameters (follow step (c) in Stage 1 to view) to make the drone reach an altitude of 10 metres, consequently making the motors run according to the set thrust and other parameters. We notice that moving the RC joysticks doesn't affect the drone anymore.
- g. When we flip the switch to manual control, we notice that the drone is still armed as we haven't commanded it to disarm yet. We also notice that we have regained RC control by moving the RC joysticks.
- h. We can verify the mode switch and working as many times as we want by flipping the assigned RC switch.
- i. To stop, switch back to Manual mode and disarm the motors.
- j. This culminates Stage2 verification.

2. Challenges

Challenges were faced while installing MAVLink and MAVROS due to dependency issues. Not many challenges were faced otherwise.

3. Teamwork

Clare and Logan were tasked with understanding the map_server and map_utils ROS packages to change our point cloud map into a voxel grid map and colorize it. They have familiarized with the packages and started development on the same. Clare is also responsible for achieving time synchronization of the different sensor data streams.

Angad was working on processing IMU data from older bag files and improving the state-estimate precision for reliable matching of RGB and LiDAR data and consequent colorization. He will now be focusing on working on an SUKF filter package from the RASL Lab

4. Plan

By the next Progress Review, we plan to have the new drone and PixRacer tuned for flight and make progress on to development of the obstacle detection pipeline. We also plan to have our dense colored voxel grid map generation ready and make our state-estimation more robust and in real-time. We will also be testing our navigation code in simulator and start developing on the trajectory planning pipeline.

References

- [1] PX4, "PixRacer," [Online]. Available: <https://pixhawk.org/modules/pixracer>.
- [2] O. S. R. Foundation, "MAVROS," [Online]. Available: <http://wiki.ros.org/mavros>.
- [3] PX4, "Companion Computer for Pixhawk class," [Online]. Available: <https://dev.px4.io/pixhawk-companion-computer.html>.
- [4] FTDi. [Online]. Available: http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232RG_CABLES.pdf.
- [5] PX4, "Offboard Control," [Online]. Available: <https://dev.px4.io/offboard-control.html>.
- [6] PX4, "MAVROS Offboard Example," [Online]. Available: <https://dev.px4.io/ros-mavros-offboard.html>.
- [7] MAVLINK, "MAVROS GitHub," [Online]. Available: <https://github.com/mavlink/mavros>.