ILR #2

Adam Yabroudi- Team A

October 23, 2015

Teammates: Tushar Agrawal, Pratik Chatrath, Sean Bryan

Individual Progress

This week I was in charge of setting up the hokuyo sensor on a raspberry pi. This task served two functions. It first taught us how to integrate with the Hokuyo on ROS. The second benefit was that we were able to set up a Raspberry Pi with ROS and see how its performance compared to the BeagleBoard –xM.

Setting up the Raspberry Pi with ROS

Setting up the board was mostly a timely process. I downloaded NOOBS with the Rasbian OS onto an SD card to get the Raspberry Pi in a brand new and clean state. I ran into issues with the SD card formatting but eventually was able to successfully set up Rasbian.

I think followed instructions online (http://wiki.ros.org/ROSberryPi/Installing%20ROS%20Indigo%20on%20Raspberry%20Pi) to install Indigo on the Rasberry Pi. Since the Raspberry Pi has a different architecture than laptops, it required a modified installation process that took into account the differences in dependencies. Running through the scripts was a multi-hour process and on my first try I was unsuccessful. I then had to format another SD card and restart the whole process and I finally had ROS running on the Raspberry Pi.

Hokuyo On ROS

Our original intention was to test out the Hokuyo on ROS on the Raspberry Pi to test the processing power of the Raspberry Pi. After installing ROS on the Raspberry Pi, it was functioning too slow to even attempt install the Hokuyo. Tushar, who was working near me, had installed ROS on the BeagleBoard-xM in a fraction of the time and his processing power was far greater. As a result we followed the following instructions (http://wiki.ros.org/hokuyo_node/Tutorials/UsingTheHokuyoNode) to get a Hokuyo node up and running and were able to print out the messages/data from the Hokuyo in real time. We then watched the values change as we put our hands nearer or farther from the Hokuyo.

Challenges:

SD Card

One issue that happened right away was with the SD cards. I formatted an SD card and loaded NOOBS onto it but my Raspberry Pi wouldn't recognize the SD card. After reformatting a few times repeatedly, I then changed SD cards and everything worked. They were both 8Gb SD cards so I'm not sure what happened there. The first card might have just been faulty.

Reinstalling Rasbian and ROS

As mentioned in the individual progress section, my first attempt of installing ROS on Rasbian failed. Instead of trying to fix a broken installation with lots of missing dependencies, I just repeated the whole process more carefully taking advantage of some of the information I had learned from the first installation.

Teamwork:

The project was split among the four teammates. Pratik worked on the obstacle avoidance system testing the ultrasound and IR rangefinders in various configurations. Sean helped build the mockup of the vehicle while working on the UGV, the website, and getting our FireFly6 sponsored by BirdsEyeView Aerobotics. Those projects didn't integrate with my progress from this week by design (we wanted to work on different subsystems in parallel).

Tushar was working on getting the BeagleBoard-xM up and running with ROS and OpenCV. His project was more closely related to my mine since we were comparing the Raspberry Pi to the BeagleBoard-xM in terms of performance. After this week we can conclusively say that the BeagleBoard-xM is the better board to use for vision processing. As mentioned previously, Tushar and I also worked together to test the Hokuyo ROS node. Tushar also worked on getting the camera integrated with the BeagleBoard-xM.

Future Plans

Going forward, Tushar is going to continue to work on the vision system. He will be performing more sophisticated OpenCV algorithms on board the BeagleBoard-xM and attempt to find a marker when it is within the field of view of the camera.

Pratik and Sean are going to pair up to make up for some lost ground on the obstacle avoidance system. Pratik will be handling more of the electronics and Sean will be handling more of the mechanics. As a combo they hope to gather more metrics about the sensors to make informed decisions about which sensors fit our application the best. These metrics are associated with functionality based on varying environmental conditions as well as varying performance as the baseline between two sensors is modified. Sean will also be handling the website and communication with BirdsEyeView Aerobotics.

For me, I will be mostly focusing on the Pixhawk this week trying to get the environment set up on Linux. As a quick proof of concept I will try to control a motor connected to the Pixhawk communicating with my computer as the base station. I will also continue to maintain our budget and anticipate ordering components for future tasks that are a week or two down the line. The most pressing of those orders is the NicaDrone electro-permanent magnet which we will be creating. I will organize all the appropriate files and compile the BOM and have that ordered ASAP. Lastly, I will get OpenCV installed and integrated with ROS on my laptop in anticipation to helping Tushar in between my other tasks.