# ILR 03
# Progress Review 2

Shivam Gautam

Team Daedalus

**Members: Pranav Maheshwari, Richa Varma, Mohak Bhardwaj, and Dorothy Kirlew**

**October 29, 2015**

# 1. Individual Progress

For the period following progress review 1, I undertook the following tasks:

1. Interfacing the Arduino Mega with Robot Operating System (ROS)
2. Installing libraries for Kinect and visualize the point cloud
3. Selection and procurement of the communication equipment
4. Locking down the mobile platform
5. Conceptual design for Power Distribution Board

## 1. Interface the Arduino Mega with ROS

We intend to use the Arduino Mega as our actuator control board. The Mega shall be used to control the motors of the mobile platform. The commands, however, would be generated by the Single Board Computer (SBC) which would be relayed to the Arduino through a serial link. Since our software subsystem is ROS based, it became imperative for us to find a way for the Arduino and the SBC to exchange information.

The ROS framework allows the Arduino to be interfaced as a ROS node. The 'rosserial' package is specifically designed to allow Arduino messages to be wrapped in standard ROS format. This ensures smooth data-transfer between the SBC and the Arduino.

The steps involved in setting up the Arduino with the ROS environment were-

1. Installing Binaries on the ROS workstation
2. Installing from Source onto the ROS workstation
3. Install ros_lib into the Arduino Environment

A simple "Hello World" program was uploaded to the Arduino. The Arduino, while behaving as a ROS node, published messages to the ROS topic called 'chatter'. The "Hello World" message was continuously published to this topic. This is depicted in figure 1.

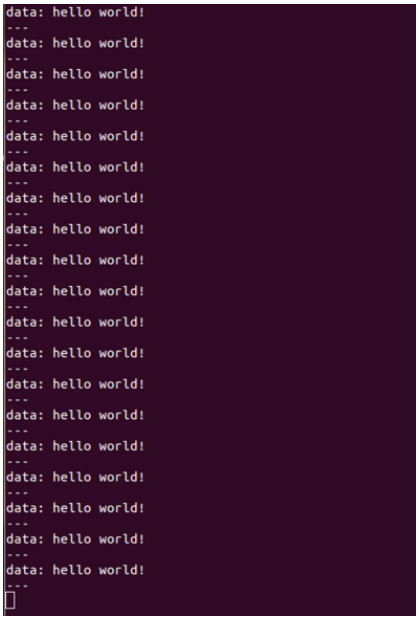


*Figure 1- Hello World Output by Arduino Mega*

## 2. Installing libraries for Kinect and visualize the point cloud

We have decided to move forward with the Kinect for accomplishing the object detection task. The first step in accomplishing this task was to install the relevant ROS packages for the Kinect. The following libraries exist for interfacing the Kinect with ROS.

- OpenNI
- Libfreenect
- Point Cloud Library (PCL)

All three libraries allowed the user to gather point cloud data from the Kinect. While the OpenNI library no longer supports the latest versions of ROS, the Libfreenect and PCL had support for ROS Indigo. The Point Cloud Library allows for greater manipulation of point cloud data and is widely used. The library can serve as a good reference point for implementing an obstacle detection algorithm.

All three libraries were installed on my workstation. The point cloud data generated was visualized using the RViz visualizer on Mohak's PC.

## 3. Selection and procurement of the communication equipment

The team had decided to implement a VANET based communication network and had finalized DiGi's DigiMesh protocol for the implementation of the network. The specific products that adhered to this protocol, as offered by Digi, were as follows-

- XBee DigiMesh/Multi-Point Adapters for 2.4 GHz and 900 MHz
- XStick USB Adapters

The XBee Multi-Point Adapter, depicted in figure 2, was selected based on their superior performance in terms of range and interference immunity. The components have been ordered and should arrive by early next week.

The selected XBee Adapter works in the 900 MHz frequency. This ensures greater range than the 2.4 GHz modules (~450 ft.) due to greater penetration power. The data we wish to transmit consists of a data structure containing the occupancy map in a string format and therefore the data rate supported (~250 Kbps) is sufficient for this application.


*Figure 2- Digi XBee Pro USB Adaptor*

## 4. Locking down the mobile platform

The Oculus Prime platform, depicted in figure 3, seems the most suitable platform for our needs currently.

*Figure 3- Oculus Prime platform*

The specifications of the platform are as follows.

- ABS Plastic Frame Set, including dock, with standard and Xtion mount upper
- LiPo battery, 5000 mAh
- AC/DC adapter
- Microsoft LifeCam Cinema Camera
- 4 x Motors, incl. 1x Encoder
- 4 x Wheels + Aluminum Hub Adapters
- 2x Cree XRE LED with Leads Attached
- Dock Marker Decal (pre-applied)
- Xaxxon MALG Motor Controller PCB
- Xaxxon Power Battery Charger PCB
- Cables: 2 x USB, SATA>MOLEX (PCB Power), 2.5mm Mainboard Power
- Speaker

- Wires, Fasteners, miscellaneous Hardware

The platform can support our electronic equipment and has ROS packages for locomotion and other functionality. This allows us to focus on the navigation and collaboration aspects of the project rather than developing the mobile platform. The motion of the platform, however, has to be constrained to best replicate the motions of an actual car.

### 6. Conceptual design for Power Distribution Board

I worked on the conceptual design of the PCB power distribution board for PCB Task with Mohak and Richa. I came up with the architecture for the power distribution board whih was focused on delivering power to the following subsystems-

- Arduino Mega
- Odroid XU4
- Microsoft Kinect

To meet the requirements of the board, I came up with a possible list of connectors and the number of pins required for each. I also worked on estimating the estimated current requirements of each subsystem and their peak ratings.

## 2. Challenges

One of the biggest challenges I faced was interfacing the Arduino Mega with the ROS framework. After installing the 'rosserial' library and uploading a simple 'Hello World' program on the Arduino Mega, I expected the Arduino to publish the 'Hello World' string. This, however, did not happen and I received the following error-

**[WARN] [WallTime: 1445556105.276560] Serial Port read failure: device reports readiness to read but returned no data (device disconnected?)**
**[INFO] [WallTime: 1445556105.277261] Packet Failed : Failed to read msg data**
**[INFO] [WallTime: 1445556105.277739] msg len is 8**

My initial instinct led me to believe that one of the probable causes for this problem could have been the code for serial communication in the 'rosserial' library. I edited the library to tinker with the baud rate but this did not yield any success. The problem was not well documented on the internet as well which proved to be deterrent in solving this problem.

Finally, after running the same program on a different version of the Linux kernel, the Arduino was able to publish messages. The problem lay with the latest kernel version of Linux which had trouble supporting external devices.

Another challenge that we faced as team was to design a power distribution board for a mobile platform that we currently did not possess. To solve this problem, we have decided to move forward with a generic power distribution board and work on revising the schematic once the platform arrives.

## 3. Teamwork

The team made good progress this week with respect to the mobile app, obstacle detection and interfacing the actuator control board. I worked with Mohak on visualizing the point-cloud data from the Kinect. Working with Pranav on interfacing ROS and Arduino Mega was a fruitful experience as this helped in eliminating the problem with the serial communication. I also worked with Richa and Mohak for designing of the power distribution board.

Dorothy and Richa successfully developed the back-end functionality for the Android App which is one of our requirements for this fall. Pranav and Mohak handled the interfacing of a Bluetooth device on ROS which would enable the app to connect with the single board computer.

## 4. Plans

The team would be working on accomplishing the following tasks for next week-

1. Mobile Platform- The team has ordered the Oculus Prime platform and is expecting it to arrive by later next week. Once the platform arrives, Richa and Mohak would be working on assembling the platform.
2. Communication System- I would be working on interfacing the XBee adaptors with the ROS framework along with Pranav.
3. Obstacle Detection- Mohak, Pranav and I would be working on creating dummy obstacle for the obstacle detection task. We would also begin implementing algorithms for obstacle detection.

- Mobile App- Dorothy and Richa would be working on adding functionality to the app which would allow it to exchange messages with a ROS node. Mohak and Pranav would be actively collaborating with them to implement this.

# References


1. http://wiki.ros.org/rosserial_arduino/Tutorials/Arduino%20IDE%20Setup
2. http://wiki.ros.org/rosserial_arduino
3. http://www.xaxxon.com/
4. http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI_tutorial_1:_Installing_and_testing