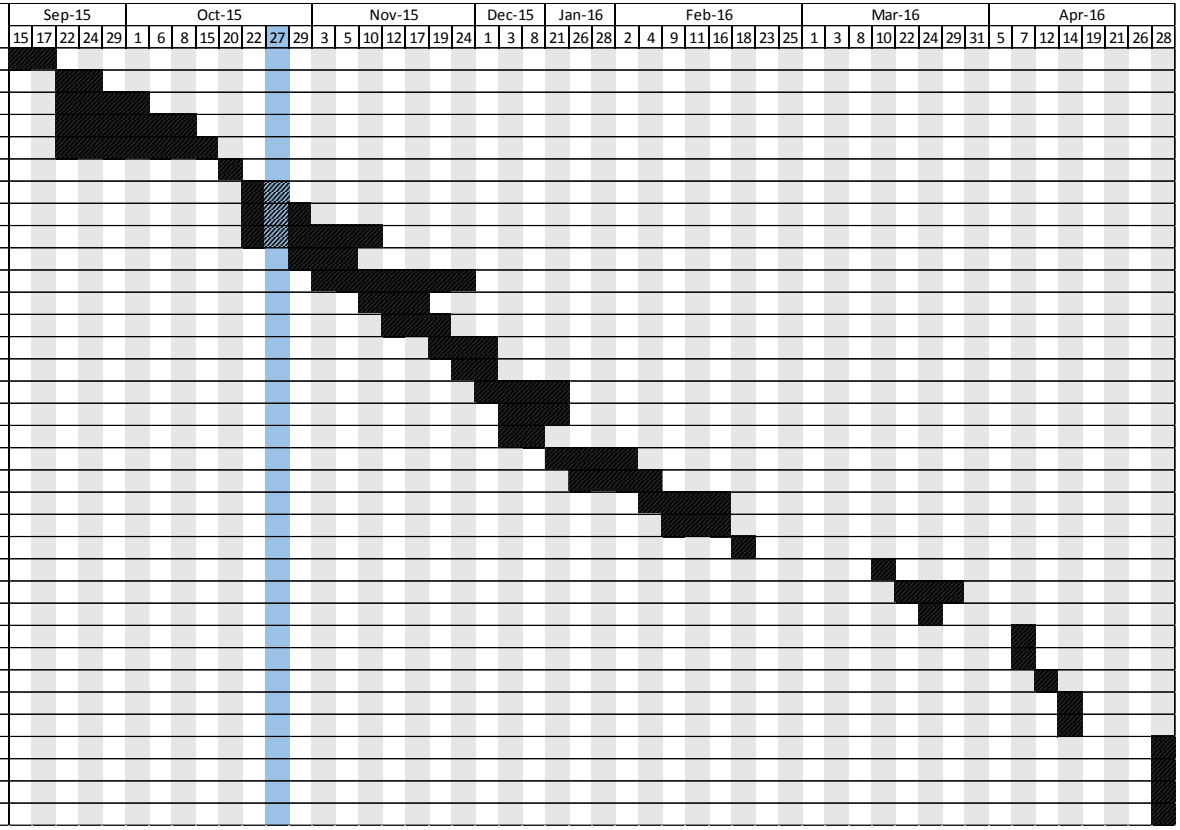
Updated Gantt Chart of Project Schedule (10/28/15)

ID	Activity	Start	Finish	Hours	Completion Percentage
1	Read Manual for Interpretation of data packet from scanner	9/15/2015	9/17/2015	3.33	100%
2	Research Image Registration Algorithms	9/22/2015	9/24/2015	4.33	100%
3	Purchase Camera	9/22/2015	10/1/2015	N/A	100%
4	Purchase Embedded Device	9/22/2015	10/8/2015	N/A	100%
5	Receive VLP-16	9/22/2015	10/15/2015	N/A	100%
6	Test Embedded Device	10/20/2015	10/20/2015	2.17	100%
7	Test Power supply to camera	10/22/2015	10/27/2015	5.83	100%
8	Implement Image Registration in MATLAB	10/22/2015	10/29/2015	7.17	0%
9	Implement Data Packet Read Function on Embedded Device	10/22/2015	11/10/2015	16	0%
10	Test image Capture Capability of Camera	10/29/2015	11/5/2015	7.17	100%
11	Implement Image Registration on Embedded Device	11/3/2015	11/23/2015	20.83	0%
12	Camera Installation	11/10/2015	11/17/2015	7.67	100%
13	Test Data Packet Read Function on Embedded Device	11/12/2015	11/19/2015	6.33	0%
14	Interface via operating system	11/19/2015	12/1/2015	8	90%
15	Test power supply to scanner	11/23/2015	12/1/2015	3.17	0%
16	Implement Image Registration for Single Frame Input	12/1/2015	1/21/2016	11	0%
17	Test Timing And Transmission of Data	12/3/2015	1/21/2016	6.17	0%
18	Test VLP-16 Scanner	12/3/2015	12/8/2015	3.17	0%
19	Process Data Packet From Scanner	1/21/2016	2/2/2016	11.33	0%
20	Image Registration For Live Video	1/26/2016	2/4/2016	11	0%
21	Camera Data Packet Transmission	2/4/2016	2/16/2016	11.5	0%
22	Orient/Install Scanner with appropriate scan angle (15 degrees)	2/9/2016	2/16/2016	6.17	0%
23	Progress Presentation	2/18/2016	2/18/2016	Deliverable	N/A
24	Student Expo Abstract	3/10/2016	3/10/2016	Deliverable	N/A
25	Test System Stability	3/22/2016	3/29/2016	Deliverable	N/A
26	Project Demonstration	3/24/2016	3/24/2016	Deliverable	N/A
27	Final Presentation (Last Lab Day)	4/7/2016	4/7/2016	Deliverable	N/A
28	Student Expo Poster Printing Deadline	4/7/2016	4/7/2016	Deliverable	N/A
29	Student Expo Poster Setup	4/12/2016	4/12/2016	Deliverable	N/A
30	Student Expo	4/14/2016	4/14/2016	Deliverable	N/A
31	Final Report (Draft)	4/14/2016	4/14/2016	Deliverable	N/A
32	Final Report	4/28/2016	4/28/2016	Deliverable	N/A
33	Final Web Page	4/28/2016	4/28/2016	Deliverable	N/A
34	Advisory Board Poster Printing Deadline	4/28/2016	4/28/2016	Deliverable	N/A
35	Advisory Board Poster Presentation	4/28/2016	4/28/2016	Deliverable	N/A



Progress Update (10/28/15)

Juan Vazquez

Currently I have been able to fully integrate a Linux environment on two different embedded devices. I have also been able to create an image capture program that interfaces both of these embedded devices with the Logitech C500 camera. Before the Odroid-XU4 was received, I was able to boot Debian on a Beaglebone Black, where I gained familiarity with the Linux environment. Once this was done, I modified an image capture c program using a "Video4Linux" library. This program is able to continuously read the video output from the Logitech C500 webcam while storing captured images in a created directory. Three images are stored and overwritten continuously with a current 10 second delay between each image capture.

Once the Odroid-XU4 was received, I began interfacing a Linux operating system. The first operating system, Xubuntu, was pre-installed onto the eMMC chip purchased with the Odroid. I was also able to successfully boot into Ubuntu Server from a microSD card. Once the interfacing of operating systems was completed, I started working on establishing an internet connection via Ethernet. This was done for both Xubuntu and Ubuntu Server by bridging an internet connection through Windows 7. With this current configuration, the Odroid is able to access the internet and download information. After an internet connection was configured, the next step was interfacing the previous image capture program used with the Odroid. After downloading the libraries for "Video4Linux" and creating the c file, the Odroid was able to successful capture and store images from the Logitech C500 camera. Future development will involve creating a similar program using "OpenCV" which will also be used for the image registration side of the project.

David Bumpus

After acquiring scan data and image data for testing and development, I am utilizing a modified SIFT approach to determine key features from the image and laser scan data. The key features from the image have been successfully obtained and can be registered to images of objects within the image. Moving forward, I will interpolate the lidar data and attempt to apply feature detection to register the lidar data with the image.

Daniel Kubik

Up until this week my main task had been to complete simulations of reading lidar data. Using sample data provided by Velodyne, and later our own data acquired from our VLP-16 Puck, I created MATLAB scripts that can parse the data to recover encoded distance, reflectivity, and time measurements. These simulations were a success, and I was able to replicate a 3D (three-dimensional) plot of the distance data in MATLAB identical to the view that can be attained using the Veloview software.

My next task is to implement these MATLAB scripts onto our Odroid XU4. To do this, I will need to rewrite these scripts in either C or C++ so that it is compatible with the Odroid. After a discussion with the team and our product advisor, we decided that C++ will be the better choice as this will allow for the use of Object Oriented Programming (OOP), which in the long run can save time. Unfortunately, while I am proficient in C, I have not had as much experience with C++ or OOP. This past week I have been reading textbooks and examples to help bring myself up to speed with OOP standards such as classes, constructors, deconstructors, etc.