Arizona State University
Ira A. Fulton Schools of Engineering
School of Electrical, Computer, and Energy Engineering
EEE 488: Senior Design Laboratory I

**To:**
Michael Kozicki (Professor)
Mehmet Balaban (Teaching Assistant)
Brian Skromme (Project Mentor) <skromme@asu.edu>

**From:**
Vanessa Martinez (Leader) <vimartin@asu.edu>
Paul Rayes (Software) <prayes@asu.edu>
Vu Truong (Software) <vu.truong@asu.edu>
Jossue Montes (Hardware) <montes.josh@gmail.com>

**Date:** 5 December 2014

**Project Title:** SLAM Air Quality Monitoring Robot: A Modern Implementation of environmental Monitoring Systems

Enclosed is our proposal in response to the Capstone Design Request for Proposals. We are proposing to continue work on a robot for monitoring cleanroom air particulate matter, using a modular add-on approach to potentially address multiple applications. This document describes the technical information, milestones, and budget. It also includes insights regarding the product currently in development and its role in a critical and fast-evolving industry.

# 1 Executive Summary

Cleanroom facilities must monitor the air quality in their clean rooms. However, research and interviews with clean room industry experts suggests that the methods of monitoring the quality of the air inside cleanrooms is not as effective as it could be. This includes the amount of time and labor required to monitor air quality, the amount of money being spent on sensors, and the ease of use of the monitoring solution.

In particular, current air quality monitoring solutions require one of two problematic approaches. First, a user may walk around the cleanroom with a hand-held sensor, which requires training that user and potentially disrupting cleanroom operations. Second, a large number of sensors may be installed at various locations within the cleanroom, which may be prohibitively expensive.

This proposal is to design a small modular robot to solve the issues listed above. The robot can detect air quality throughout an environment using a single sensor. It will build a map of the cleanroom environment with the air quality measurements presented in a color-coded "heat map," with additional details available if desired. This data is presented to the user on a web interface similar to many modern web applications. The modular approach means the robot could also have hardware add-on modules to potentially address other applications; this proposal focuses on air quality monitoring.

The project is divided into four milestones:

- Milestone 1: Robot drives like an RC car (Nov 20, completed)
- Milestone 2: Display various sensor data on web interface (Dec 30)
- Milestone 3: Hardware prototypes (Feb 26)
- Milestone 4: Software (Apr 29)

The cost of the robot will be under $700, plus the cost of the air quality sensing module, which will vary depending on the desired accuracy of the sensor.

# 2 INTRODUCTION

The existing air quality monitoring methods in clean rooms used by sensitive electronics manufacturers in semiconductor industry have areas where they can be improved. The two deficiencies this project will address are, air quality devices located at fixed locations and the hand-held requirement. The main problems with fixed air quality sensors are there has to be a large number of them in a clean room environment and a third party must be contracted to complete an air quality assessment. If not strategically placed, a large number of sensors can lead to ineffective readings of an environment. The sensors also have the potential to neglect areas at high risk for air contamination. The downside to hiring outside personnel to conduct air quality test is the cost. It is typically very expensive, which may lead to less frequent testing. This project proposes a more efficient and cost-effective way to address the surveillance inaccuracies and high costs semiconductor companies face in cleanroom air quality monitoring.

The industry-ready prototype of our product is affordable and fully automated. It is a small modular robot capable of simultaneous localization and mapping (SLAM) of its environment. It will map out its environment, remain aware of its location, and the air quality within the environment. The platform will support various hardware add-ons. This approach, however, will focus on air quality monitoring. The works-like prototype of the design will use an inexpensive sensor for proof of concept. The data obtained from the air quality sensor will be mapped in a 2-D plane and displayed the web interface. The web interface will also show the direction and speed. This project proposes an autonomous air quality monitoring robot to address the costly air quality assessment challenges faced by the semiconductor industry.

This proposal discusses the following:

- Technical details of the robot prototypes
- Technical details of the air quality sensor
- Technical details of the robot software
- Technical details of the web interface
- Analysis of ABET EC2000 criterion
- Project milestones and schedule
- Completed and planned tasks
- Available project resources
- Project budget

# 3 TECHNICAL NARRATIVE

This project involves several distinct parts: The robot hardware itself, the robot software, the air quality sensor, and the web interface. The robot contains the hardware, electronics, and sensors common to any application involving a SLAM robot. The robot software runs on the robot and controls the electronics and sensors. The air quality sensor measures the number of particles in the air, and for the works-like prototype will be an existing commercially-available sensor. The web interface runs on any existing modern computer, displays all relevant data to the user, and allows the user to give commands to the robot.

## 3.1 Air Quality

Consumer research, which involved scheduling field visits to several facilities, was conducted and analyzed extensively throughout the course of the project. All cleanroom facilities are issued an industry-standard Class rating that reflects the purity of their cleanroom environment(s). The Class rating is determined based on how many contaminant particles are detected in a set volume; for example, how many actual particulates per cubic inch. The usual culprits that contaminate a cleanroom environment are limited very small dirt or debris particles that measure on the order of tenths of micrometers in size.

Devices that test the air quality are priced according to their resolution, or the size of the smallest particle that can be recorded. Accordingly, devices with large particle resolutions cannot be used to issue high class ratings to a facility, since it is incapable of detecting very small or fine debris. The maximum environmental particle count requirement of a given Class rating is exacting[1]; however, the relationship between a desired Class rating and the required device resolution is relatively straightforward: i) Class 1000 ratings can only be issued using a device resolution of 0.5 microns or lower, ii) Class 100 ratings require a device resolution no larger than 0.3 micron, and iii) Class 1 ratings require 0.1 micron resolution device. The current SLAM robot will be equipped with a 1 micron resolution air sensor and all testing, simulation, and data-collecting will take place with this resolution in mind. Since the SLAM robot is intended to be modular by design, the end product can be custom tailored to the user’s needs and can be outfitted with a sensors of any resolution.

## 3.2 Robot Hardware

A block diagram with the hardware for the robot works-like prototype and air quality sensor is shown in Figure 1. A BeagleBone Black (BBB) running Ubuntu is the main computer, and connects directly to all the hardware. All the electronics are powered by a 2-cell Lithium Ion battery, either directly, through a 5V regulator, or through a 3.3V regulator internal to the BBB. A power LED indicates whether the robot is on and receiving power from the battery. The connection protocol used for each device is also shown. The connection to the user’s computer (labeled Laptop) is over standard WiFi, and thus requires a WiFi network to be set up. This will be the case in many, but not all, environments. A motor driver controls the speed and direction of the motors. This hardware is fully assembled.

The LCD and webcam are optional, and may be used to gain additional information about the robot’s operation or the environment. Both these devices are connected to the BBB through a USB hub. The gamepad is also optional, as all commands may be given using the computer’s keyboard, but in the non-autonomous version it provides an easier method of control. This hardware is connected, but the software for the LCD is not yet written.

There are two optical encoders on each wheel for a total of four sensors, which measure the speed and direction of rotation of the wheels. The inertial measurement unit (IMU) measures acceleration, angular velocity, and magnetic fields, which are used to determine the robot's relative motion. The LIDAR includes both a laser rangefinder and motor, and measures the distance to obstacles 360° around the robot approximately five times per second. The optical encoders and IMU hardware is assembled; the LIDAR is currently in the process of being purchased.

The air quality sensor will be integrated into the robot for the works-like prototype, and initially will be a commercially available sensor. A final product would have this sensor as an add-on module. A sensor has been chosen, and it will be purchased at the beginning of the Spring semester.

Many of these pieces will be either located on or connected to a custom circuit board (PCB). A schematic of this board is shown in Figure 3 in Appendix A. This board will connect directly to the BBB. The exact pins used to connect the hardware to the BBB (through this circuit board) are listed in Table 2 in Appendix B. This circuit board is currently on a prototyping board; it will be turned into a custom designed circuit board for the final works-like prototype.

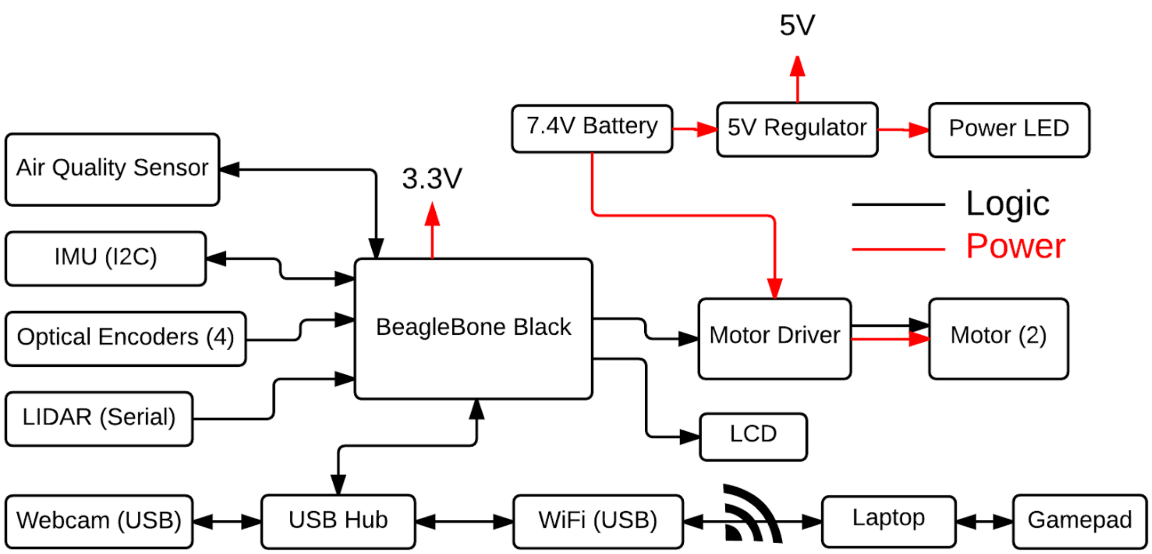



Figure 1. System Block Diagram for robot and air quality sensor hardware.

### 3.3 Software

The software is divided into two pieces: The robot software, and the web interface. The robot software handles navigation, localization, and mapping. It also works with the add-on hardware, which is the air quality sensor in this prototype. The web interface runs on any modern browser on the user's computer. It displays the robot's status, the map of the environment, sensor data, and allows the user to control the robot. A final product would have the software also be capable of high-level commands that the robot would execute autonomously and support software add-ons to work with the hardware add-ons, but the works-like prototype will have all the software in one package for simplicity.

The robot software is written in Node.js, which was chosen because it is based on Javascript, the same language the web interface is required to use, and its asynchronous design is ideal for robotics applications. It is composed of several modules, which are shown in Figure X along with with how they work together. The web interface software is organized in a similar fashion.

The current software version contains the IMU, Optical Encoders, Navigation, and Communication modules. The other modules will be written in the Spring semester. The modules of the web interface corresponding to the completed robot software modules are also completed.

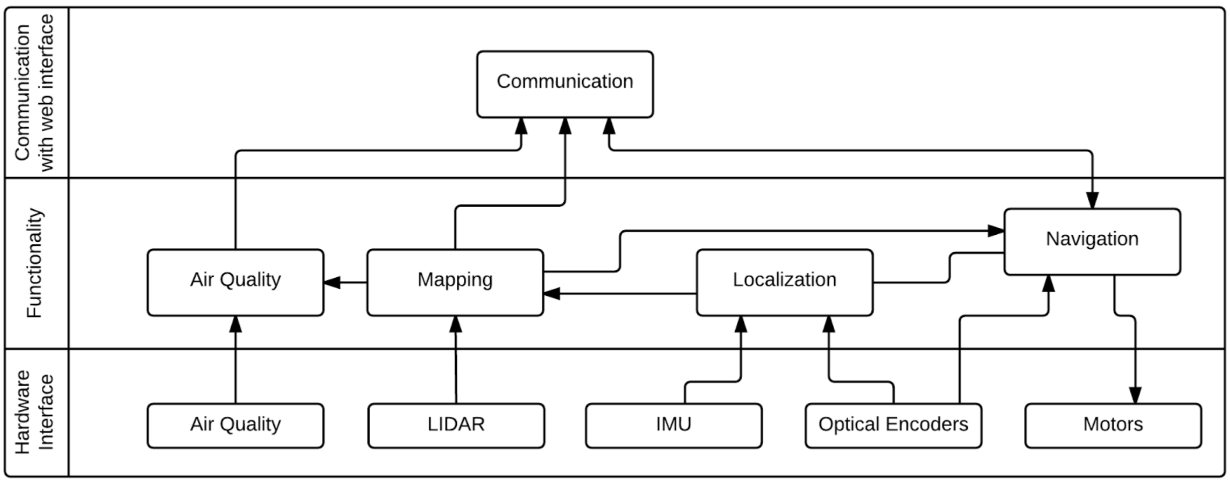



Figure 2. Block diagram of software modules.

## 3.4 ABET EC2000 Criterion

Per ABET EC2000[2], the project was designed with a sense of ethics and responsibility in mind. To this end, the following considerations were taken into account throughout the project.

### *3.4.1 Economic*

Depending on the specific needs of the consumer, air testing devices can cost from $1500 (on the low-end, educational facility side) to as far as $20,000+ (large-scale commercial). The consumer allocates further resources towards maintenance/repair costs, coupled with staff training in the case of sophisticated equipment. Evidently, reliable, accurate air testing is responsible for a significant chunk of the side costs to the consumer. The SLAM robot is designed with the end user’s expenses in mind. Its creators believe that its unique assembly and novel method of operation can significantly reduce side costs for the consumer, allowing the consumer to allocate a larger portion of resources towards main production.

### *3.4.2 Sustainability*

A small, efficient, economical 7.4V lithium ion battery powers the robot when it is not plugged in via USB to a computer. Plugging in enables battery charging, and the battery can of course be replaced at the end of its lifespan. So long as the robot is properly cared for, the lifetime of an individual unit is comparable to that of most consumer electronics.

### *3.4.3 Manufacturability*

The SLAM robot's workhorse is the small BeagleBone Black (BBB) single-board computer that serves as the main controller. It, along with the ensemble of electronic components mentioned earlier are assembled together through basic electrical soldering. The coding and software that drives the robot is completely open source (non-proprietary) and the electronic components are extremely cheap to buy in bulk; unfortunately, the BBB is not available for purchase in large quantities. This is a non-issue, however, since other single-board computers can be used – the BBB simply presented convenience. As a final note, all the components (including the BBB) can be recycled, provided that a non-lead based electrical solder is used – the current version uses lead.

### *3.4.4 Health and Safety*

The LIDAR component of the robot uses a laser to map the environment around it. There is potential for damage to human retinas if the user looks directly into this component. Since the robot is intended for no human intervention, it is unlikely that this is a threat. Also, the laser is emitted in brief, spaced-apart pulses so it isn't firing continuously, making even intentional contact with retinas exceedingly difficult. Additionally, there is a common concern from industry experts that the robot may accidentally bump into researchers/workers during its normal mode of operation. With sufficient coding this should not be a concern, as the autonomous model should not have difficulty recognizing objects in its path and altering its trajectory. Even at top speed, an impact with a human body could not possibly result in any serious injury. An impact at top speed might have just enough momentum to knock over a full glass of water, for example.

# 4 Management Plan

Problem: Existing air quality sensors primarily suffer from one of two disadvantages. Many are fixed to one location, and several are needed to give a broad picture of air quality in an environment. Hand-held models exist, but these require a person to carry them around and can be prohibitively expensive.

Our solution will improve on the existing solutions by being cheaper than a hand-held model and recording air quality data throughout the environment. Also, it will not require a human to carry it, will not require specialized training, and can be used instantly with little or no setup. The initial prototype will not be autonomous, but with that addition this will be an even more compelling solution as it could be set up and then simply left alone.

## 4.1 Tasks and Milestones

The milestones with their tasks are listed below, with the percent completed, start date, and finish date, sorted by finish date. If the percent completed is 100% then the finish date is the actual date, otherwise it is the scheduled date. See the Gantt chart in Appendix E for all scheduled tasks, or the timeline in Appendix D for the milestones and tasks listed below in a visual format.

- Ongoing tasks: Research, planning, and reporting
  - Research applications (100% Sept 5 - Sept 30)
  - Plan tasks, hardware, software, and goals (75% Sept 5 - Dec 18)
  - Research air quality requirements (23% Sept 5 - Feb 19)
- Milestone 1: Robot drives like an RC car (Nov 20)
  - Basic robot hardware and overhead (100% Summer - Sept 12)
    - Test hardware connection to BBB. Develop code for web interface to display desired and actual speed and direction. Develop Arduino code for motor drivers.
  - Robot drives like an RC car (100% Sept 12 - Nov. 20)
    - Hardware configuration of motors to BBB. Set motor speeds. Use gamepad to drive robot. Display feedback on web interface.
- Milestone 2: Display various sensor data on web interface (Dec 30)
  - Display data from optical encoders, IMU (36% Sept 30 - Dec. 16)
    - Optical encoders: Read the speed of the wheel from optical encoder. Convert data to RPMs. Use an interrupt instead of timer.
    - IMU: Obtain orientation and acceleration. Display orientation and acceleration.
  - Display LIDAR data (30% Oct 2 - Dec. 30)
    - Obtain environment data. Display environment data in a 2-D plane.
  - Display air quality data (30% Nov 12 - Dec 30)
    - Obtain air quality data. Display air quality data.
- Milestone 3: Hardware prototypes (Feb 26)
  - Design & manufacture looks-like prototype (0% Dec 31 - Feb 17)
    - Research, design, fabricate, and assemble parts.
  - Design & manufacture custom PCB (0% Dec 31 - Feb 26)
    - Research, design, fabricate, and assemble circuit board.
- Milestone 4: Software (Apr 29)
  - Calculate & display robot speed, orientation (0% Nov 26 - Feb 17)

        - Develop code to convert RPMs from optical encoders to speed. Develop code to use x, y, and z acceleration data from IMU. Set max motor speeds. Map environment will stationary. Map environment while rotating
    - Calculate & display robot position (0% Feb 18 - Mar 31)
        - Develop code to combine speed, acceleration, and orientation data to define position of robot.
    - Display map with robot, environment, and A.Q. data (0% April 1 - April 29)
        - Develop code to fuse robot orientation and acceleration data with environment. Develop code to display air quality data of environment.
- Optional/low-priority tasks
    - Display debugging data and robot status on LCD (57%)
    - Display streaming video on web interface (61%)

## 4.2 Resources

### *4.2.1 Facilities*

Facilities available for the project are TechShop Chandler; ASU libraries, study rooms, and conference rooms; as well as team members' personal work areas and equipment.

### *4.2.2 Capabilities*

Josh is chiefly tasked with the hardware designs of the SLAM robot, with responsibilities that start at the circuit diagram level and end with assembly of the robot itself. Chiefly, his duties include electrical soldering of components. He also has extremely relevant lab experience in a cleanroom facility at ASU's CSSER cleanroom.

Paul is primarily in charge of the software design of the robot and web interface. Paul worked on an autonomous robot in the past, and thus has experience with the types of problems involved in robotics. He has significant software experience in the languages used in the project, including web interface design. He also has some knowledge working with CAD tools to design both circuit boards and mechanical parts.

Vu is primarily focusing on the software designs of the robot, his role includes: writing code for the robot to perform various functions, researching, and debugging code errors. His knowledge in programming will be used to configure and port components to the BBB. He will manipulate data from both the IMU and LIDAR to assist in the code which in turn will be used in determining the location of the robot. Whereas, his researching and debugging skills are used for troubleshooting obstacles that show up upon coding.

Vanessa is the team lead of the SLAM Air Quality Monitoring Robot team. She is primary responsibility is in scheduling, organizing, and leading team and team-advisor meetings. She develops the content for the written and presentation aspects of the project. She handles team conflicts and concerns. She focuses on hardware testing and verification of electrical connections of components. She also supports the team in research.

#### *4.2.3 Expertise*

- Industry cleanrooms: Freescale Semiconductor, Particle Measuring Systems, UCT, Microchip (pending), ON Semiconductor (pending)
- Research cleanrooms: ASU CSSER and ASU MTW cleanrooms
- Individuals: Michael Kozicki, Arthur Handugan and Stefan Myhajlenko (ASU CSSER), Roy Camarena (UCT)
- MyRO PCB: PCB manufacturing company

### 4.3 Budget

#### *4.3.1 Equipment*

No equipment purchases are necessary for this project, as all necessary equipment is available for use at TechShop Chandler. Some equipment used is a soldering iron, 3D printer, multimeter, oscilloscope, logic analyzer, laser cutter, and other miscellaneous tools (screwdrivers, wire cutters, etc.).

#### *4.3.2 Robot Works-Like Prototype Supplies*

The hardware and electronics required for the works-like prototype are listed in Table 1. Some additional parts were also ordered for spares, or for testing before deciding on which part would work best. Prices are accurate as of 4 December 2014.

The LIDAR costs over $100, however for this project it is the best way to accomplish environmental mapping. Infrared sensors do not provide enough accuracy for the project, and ultrasonic sensors cost nearly as much as a LIDAR while still not providing as much accuracy and requiring additional code to use. The LIDAR is both highly accurate and easier to use. Most competing LIDARs with 360° sensing cost significantly more, making this one a good price.

The air quality sensor in the final product will need to be more accurate than this one. For a class 100 cleanroom, that will be between $300-500 for a 0.3um sensor; existing solutions for cleanrooms cost over $1k. For a class 1 cleanroom, that will be between $5k-10k for a 0.1um sensor; existing solutions for cleanrooms cost approximately $20k. For the initial works-like prototype the $15 sensor with 1um resolution will suffice, and is accurate enough for a class 1000 cleanroom[1].

#### *4.3.3 Robot Looks-Like Prototype Supplies*

The looks-like prototype will be manufactured using 3D printed or laser cut plastic at TechShop Chandler. The cost will depend on the material volume and time but will not be significant.

#### *4.3.4 Subcontracts*

Manufacturing of the custom circuit board will be done by a third party company, such as MyRO PCB. The cost for this will depend on the size of the board and is estimated to be under $100. All other manufacturing will be done using the resources available at TechShop Chandler.

Table 1: Supplies needed for works-like prototype

| **Item** | **Vendor** | **Part No.** | **Price/each** | **Quantity** | **Description** |
|---|---|---|---|---|---|
| Chassis and Motors | DFRobot | ROB0049 | 34.35 | 1 | Forms the mechanical base of the robot with two motors and a caster |
| Chassis upper deck | DFRobot | DFR0310 | 3.50 | 1 | Forms the mechanical top of the robot |
| Optical Encoders | DFRobot | SEN0116 | 2.90 | 2 | Tells how fast the wheels are spinning. |
| BeagleBone Black | Adafruit | 1876 | 55.00 | 1 | Central single-board computers |
| Proto Cape | SparkFun | DEV-12774 | 9.95 | 1 | Prototype for custom circuit board |
| IMU | Adafruit | 1714 | 19.95 | 1 | Provides info about the orientation and acceleration of the robot |
| LIDAR | RobotShop | RB-Rpk-01 | 398.99 | 1 | Environmental mapping |
| LCD | Sparkfun | LCD-09568 | 29.95 | 1 | Provide debugging information |
| Webcam | Amazon | B000Q3VECE | 13.97 | 1 | Streaming video |
| Dust Particle Sensor | Seeed Studio | SEN12291P | 15.90 | 1 | Detects airborne particles |
| Motor Driver | SparkFun | ROB-09457 | 8.95 | 1 | Powers the two motors |
| USB Hub | Amazon | B002FFT8Z6 | 5.34 | 1 | Gives more USB slots |
| Battery | BatterySpace | CU-LC-14430S2 | 14.95 | 1 | Powers the robot |
| Battery Charger | BatterySpace | CH-L7405 | 19.95 | 1 | Charges the battery |
| Miscellaneous small electronics (resistors, etc.) | | | | | |
| Miscellaneous supplies (solder, wires, etc.) | | | | | |
| **Total, not including custom PCB or air quality add-on module. Parts for final product will be cheaper due to bulk discounts.** | | | **636.55** | | |

# 5 CONCLUSION

The semiconductor industry powers the field of new and improved electronics, which enjoy a permanent, lasting place in our lives. One of the biggest operating costs facing the semiconductor industry arises from the need for high cleanliness and purity of their manufacturing facilities. Facilities allocate a significant amount of resources annually to meet the standards of cleanliness. They may sacrifice utility and accuracy for lower costs when it comes to monitoring the sterility of their environments. Devices that are manufactured in unclean environments will underperform or fall short of consumer expectations.

Existing solutions primarily suffer from one of two disadvantages. Many are fixed to one location, thus several are needed to give a broad picture of air quality in an environment. Hand-held models exist, but these require a person to carry them around and can be prohibitively expensive. The features that differentiate of our product from existing solutions are that it is less expensive and can record air quality data throughout the environment. The robot is fully automated and made for next to zero user intervention. Also, its hands-off mode of operation keeps facilities running smoothly without causing any kind of production delay.

Although semiconductor manufacturing facilities constitute the largest portion of the clientbase, the uses for the SLAM robot can extend far beyond environmental monitoring. The initial prototype will not be autonomous, however this will be an even more compelling solution as it could be set up and then simply left alone. It will not require a human to carry it, nor will it require any specialized training, and it can be used instantly with little or no setup. The most appealing aspect of this is the lower cost, which will allow industries to pull resources back towards main production. Given the existing solutions, we believe this robot will revolutionize the way air quality is surveyed in environments, provide the best method of verifying clean room air particulate levels, and create more sustainable and hygienic working spaces.

# 6 REFERENCES

1. Airborne Particulate Cleanliness Classes in Cleanrooms and Clean Zones, FED-STD-209E, 1992.
2. Criteria For Accrediting Engineering Programs, ABET, Baltimore, MD, 2012, pp. 3.
3. D. Molloy. *Beaglebone: GPIO Programming on ARM Embedded Linux*. [Online] Retrieved 2014, December 4. Available: http://derekmolloy.ie/beaglebone/beaglebone-gpio-programming-on-arm-embedded-linux/

# 7 APPENDIX A: SCHEMATICS

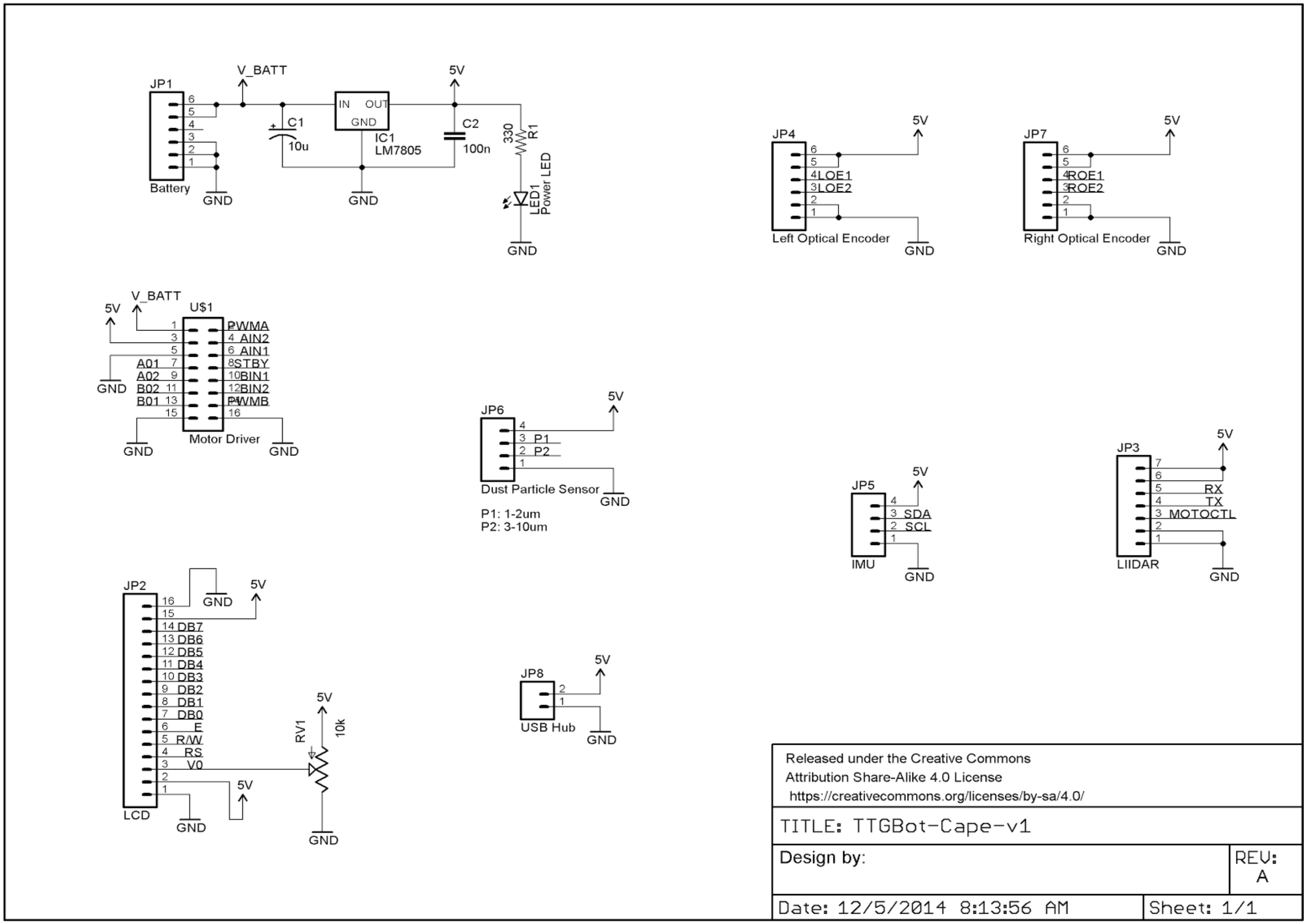



Figure 3. Schematic of robot PCB.

# 8 APPENDIX B: BEAGLEBONE BLACK PIN ALLOCATION

The default pin allocation for the BBB allocates pins to an HDMI interface, MMC flash storage, and user LEDs[3]. The HDMI interface has been disabled, thus those pins are available for use; the MMC interface and user LEDs are still enabled, thus those pins may not be used. The pins used to interface with the robot's hardware are listed in Table 2. Device pins that are not listed are not electrically connected to the BBB.

Table 2. BeagleBone Black pin allocations.

| Device | Device pin name | Device pin number | BBB pin name | BBB pin mode | BBB pin number | Description |
|---|---|---|---|---|---|---|
| **IMU** | SDA | | I2C2_SDA | 3 - I2C2_SDA | P9_20 | Serial data (I2C) |
| | SCL | | I2C2_SCL | 3 - I2C2_SCL | P9_19 | Serial clock (I2C) |
| **TB6612FNG Motor Driver** | STBY | | UART4_TXD | 7 - gpio0[31] | P9_13 | Standby (disable outputs) |
| | PWMA | | EHRPWM1A | 6 - ehrpwm1A_mux1 | P9_14 | Speed of motor A (PWM) |
| | PWMB | | EHRPWM1B | 6 - ehrpwm1B_mux1 | P9_16 | Speed of motor B (PWM) |
| | AIN1 | | GPIO1_28 | 7 - gpio1[28] | P9_12 | Direction of motor A |
| | AIN2 | | UART4_RXD | 7 - gpio0[30] | P9_11 | Direction of motor A |
| | BIN1 | | GPIO1_16 | 7 - gpio1[16] | P9_15 | Direction of motor B |
| | BIN2 | | GPIO1_17 | 7 - gpio1[17] | P9_23 | Direction of motor B |
| **Left Optical Encoder** | LOE1 | 1 | GPIO2_12 | 7 - gpio2[12] * | P8_39 | O.E. signal 1 for left wheel |
| | LOE2 | 2 | GPIO2_13 | 7 - gpio2[13] * | P8_40 | O.E. signal 2 for left wheel |
| **Right Optical Encoder** | ROE1 | 1 | GPIO2_10 | 7 - gpio2[10] * | P8_41 | O.E. signal 1 for right wheel |
| | ROE2 | 2 | GPIO2_11 | 7 - gpio2[11] * | P8_42 | O.E. signal 2 for right wheel |
| **HD44780 LCD** | DB4 | 11 | TBD | | | Data bus bit 4 |
| | DB5 | 12 | TBD | | | Data bus bit 5 |
| | DB6 | 13 | TBD | | | Data bus bit 6 |

| Device | Device pin name | Device pin number | BBB pin name | BBB pin mode | BBB pin number | Description |
|---|---|---|---|---|---|---|
| | DB7 | 14 | TBD | | | Data bus bit 7 |
| | RS | 4 | TBD | | | Register select |
| | E | 6 | TBD | | | Enable |
| | V0 | 3 | TBD | | | Contrast voltage |
| | R/W | 5 | TBD | | | Read/Write (tied to GND for write) |
| **RPLIDAR** | RX | | UART5_TXD | 4 - uart5_txd | P8_37 | Receive |
| | TX | | UART5_RXD | 4 - uart5_rxd | P8_38 | Transmit |
| | MOTOCTL | | UART3_CTSN | 2 - ehrpwm1A | P8_36 | Motor speed control (PWM) |
| **SM-PWM-01A Dust Particle Sensor** | P1 | 4 | GPIO1_13 | 7 - gpio1[13] * | P8_11 | 1-2um particle density |
| | P2 | 2 | GPIO1_12 | 7 - gpio1[12] * | P8_12 | 3-10um particle density |

* The optical encoders and dust sensor are connected to PRU-capable pins, as they involve real-time processing of data. Thus, we could use the PRU for either of these devices if necessary by changing the pin mode and using a separate assembly program. These pins also work as regular GPIOs.

# 9 Appendix C: Code

All code is located in a private Git repository. Contact any of the team members with your BitBucket username for access to the repository.

Repository URL: https://bitbucket.org/the7thGhost/ttgbot

# 10 Appendix D: Timeline

Figure 4 shows the project milestones (not including planning), with the high-level project tasks. The “Today” marker indicates December 5.

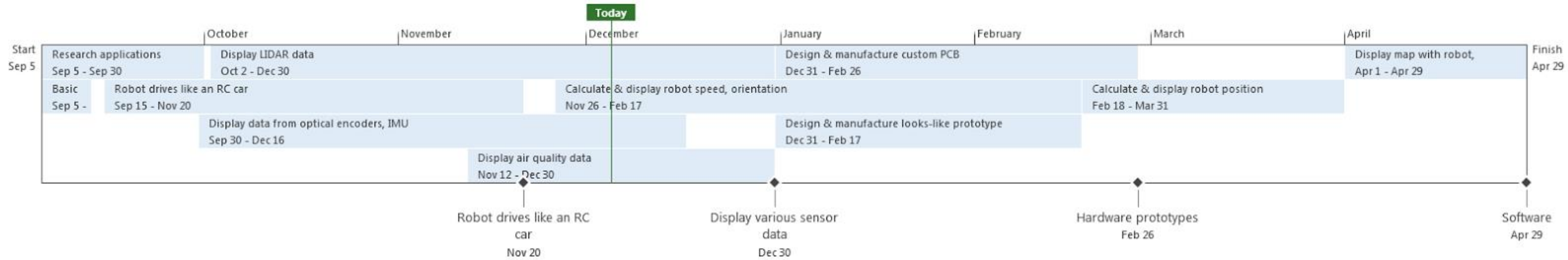
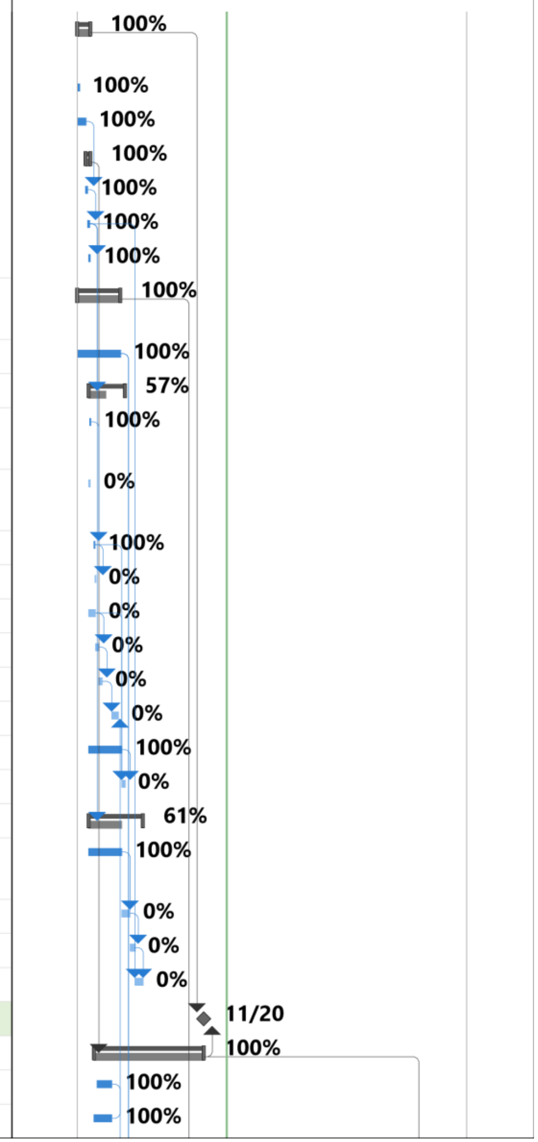


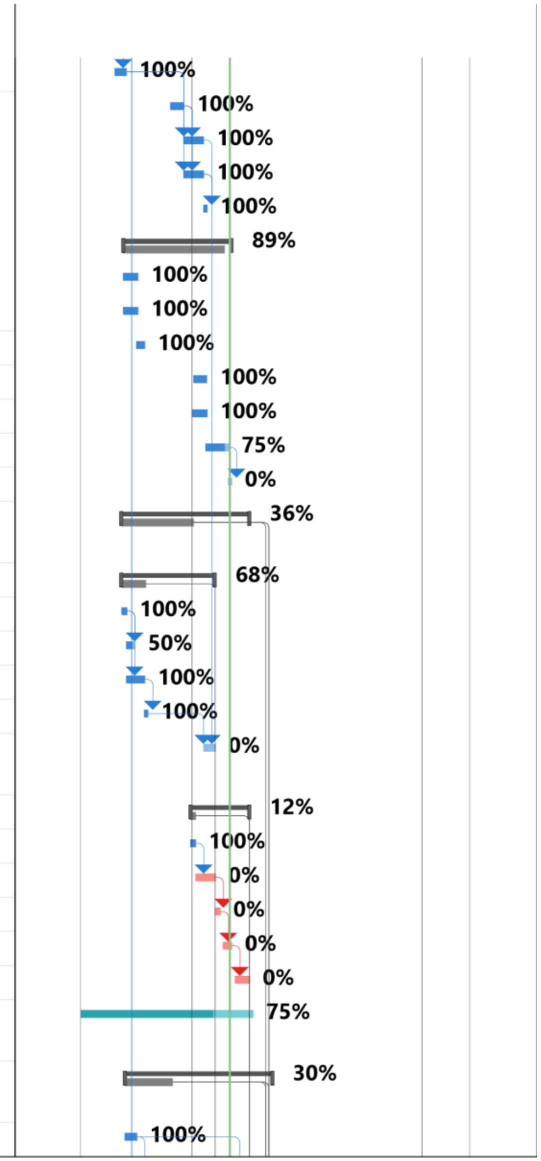

Figure 4. Timeline with project milestones and high-level tasks.

# 11 Appendix E: Gantt Chart

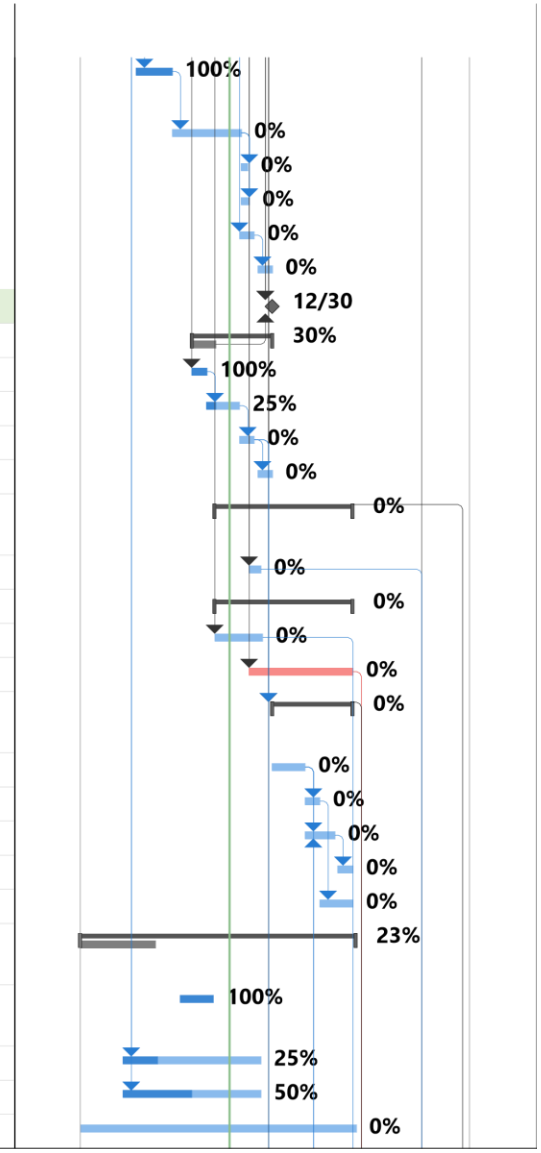
| | | | | | | |
|---|---|---|---|---|---|---|
| 12 | Basic robot hardware and overhead | 6 days | Fri 9/5/14 | Fri 9/12/14 | | |
| 13 | Summer tasks | 1 day | Fri 9/5/14 | Fri 9/5/14 | | |
| 14 | Install Ubuntu | 3 days | Fri 9/5/14 | Tue 9/9/14 | | Paul |
| 15 | Get WiFi @ ASU | 3 days | Wed 9/10/14 | Fri 9/12/14 | | Paul |
| 16 | Get WiFi working | 1 day | Wed 9/10/14 | Wed 9/10/14 | 14 | |
| 17 | Connect to ASU's WiFi | 1 day | Thu 9/11/14 | Thu 9/11/14 | 16 | |
| 18 | Write WiFi instructions | 1 day | Fri 9/12/14 | Fri 9/12/14 | 17 | |
| 10 | Research applications | 18 days | Fri 9/5/14 | Tue 9/30/14 | | |
| 11 | Decide on application | 18 days | Fri 9/5/14 | Tue 9/30/14 | | |
| 32 | Video Streaming-low priority | 16 days? | Fri 9/12/14 | Fri 10/3/14 | 17 | |
| 34 | Connect Webcam & take photos | 0.5 days | Fri 9/12/14 | Fri 9/12/14 | | Josh |
| 38 | Hack webcam and USB hub to fit | 1 day? | Fri 9/12/14 | Fri 9/12/14 | | Josh |
| 35 | Stream video | 0.5 days | Mon 9/15/14 | Mon 9/15/14 | 34 | Paul |
| 36 | Video on Website | 0.5 days | Mon 9/15/14 | Mon 9/15/14 | 35 | Paul |
| 37 | Attach servo | 2 days | Fri 9/12/14 | Mon 9/15/14 | | Josh |
| 40 | Wire servo | 2 days | Tue 9/16/14 | Wed 9/17/14 | 37 | Josh |
| 41 | Code for servo | 2 days | Thu 9/18/14 | Fri 9/19/14 | 40 | Vanessa |
| 42 | Control servo | 2 days | Fri 9/26/14 | Mon 9/29/14 | 41,26 | Vu |
| 33 | Buy USB hub | 14 days | Fri 9/12/14 | Wed 10/1/14 | | Paul |
| 39 | Attach to robot chassis | 2 days | Thu 10/2/14 | Fri 10/3/14 | 33,35,37 | Josh |
| 27 | LCD-low priority | 23 days | Fri 9/12/14 | Tue 10/14/14 | 17 | |
| 28 | Buy wires & connectors | 14 days | Fri 9/12/14 | Wed 10/1/14 | | Paul |
| 29 | Wire LCD | 3 days | Thu 10/2/14 | Mon 10/6/14 | 28 | Josh |
| 30 | Code for LCD | 3 days | Tue 10/7/14 | Thu 10/9/14 | 29 | Vanessa |
| 31 | Show info on LCD | 3 days | Fri 10/10/14 | Tue 10/14/14 | 30,17 | Vanessa |
| 1 | **Robot drives like an RC car** | **0 days?** | **Thu 11/20/1** | **Thu 11/20/14** | **19,12** | |
| 19 | Robot drives like an RC car | 49 days | Mon 9/15/14 | Thu 11/20/14 | 15 | |
| 20 | Buy connectors for motors | 7 days | Wed 9/17/14 | Thu 9/25/14 | | Paul |
| 26 | PS3 Ctrlr | 9 days | Mon 9/15/14 | Thu 9/25/14 | | Vu |



| | | | | | | |
|---|---|---|---|---|---|---|
| 21 | Wiring | 5 days | Fri 9/26/14 | Thu 10/2/14 | 20 | Josh |
| 22 | Disable HDMI Framer | 6 days | Thu 10/30/1 | Thu 11/6/14 | | Paul |
| 23 | Port Arduino code | 8 days | Fri 11/7/14 | Tue 11/18/14 | 22,21 | Vanessa |
| 24 | Get device tree working | 8 days | Fri 11/7/14 | Tue 11/18/14 | 21,22 | Paul |
| 25 | Set Motor Speeds | 2 days | Wed 11/19/1 | Thu 11/20/14 | 23,24 | Vu |
| 2 | Progress reports | 48 days | Wed 10/1/14 | Fri 12/5/14 | | |
| 3 | Presentation 1 | 7 days | Wed 10/1/14 | Thu 10/9/14 | | |
| 4 | Quad Chart 1 | 7 days | Wed 10/1/14 | Thu 10/9/14 | | |
| 5 | Memo 1 | 3 days | Thu 10/9/14 | Mon 10/13/14 | | |
| 6 | Presentation 2 | 6 days | Thu 11/13/1 | Thu 11/20/14 | | |
| 7 | Memo 2 | 7 days | Wed 11/12/1 | Thu 11/20/14 | | |
| 8 | Final report draft 1 | 10 days | Thu 11/20/1 | Wed 12/3/14 | | |
| 9 | Final report | 2 days | Thu 12/4/14 | Fri 12/5/14 | 8 | |
| 54 | Display data from optical encoders, IMU | 56 days | Tue 9/30/14 | Tue 12/16/14 | | |
| 55 | Optical encoders | 41 days | Tue 9/30/14 | Tue 11/25/14 | | |
| 56 | Test optical encoders | 3 days | Tue 9/30/14 | Thu 10/2/14 | | Vu |
| 57 | Wire Optical Encoders | 3 days | Fri 10/3/14 | Tue 10/7/14 | 56 | Josh |
| 58 | Get data from O.E. | 7 days | Fri 10/3/14 | Mon 10/13/14 | 56 | Paul,Vu |
| 59 | Convert data to RPMs | 2 days | Tue 10/14/1 | Wed 10/15/14 | 58 | Paul,Vu |
| 60 | Use interrupt instead of timer | 5 days | Wed 11/19/14 | Tue 11/25/14 | 59,24 | |
| 61 | IMU | 26 days | Tue 11/11/1 | Tue 12/16/14 | | |
| 62 | Wire IMU | 3 days | Tue 11/11/1 | Thu 11/13/14 | | Josh |
| 63 | Read IMU data | 8 days | Fri 11/14/14 | Tue 11/25/14 | 62 | |
| 64 | Get heading | 3 days | Wed 11/26/1 | Fri 11/28/14 | 63 | |
| 65 | Plot data | 5 days | Mon 12/1/14 | Fri 12/5/14 | 64 | |
| 66 | Calibrate heading | 7 days | Mon 12/8/14 | Tue 12/16/14 | 65 | |
| 97 | Plan tasks, goals, Gantt chart, schedule | 75 days | Fri 9/5/14 | Thu 12/18/14 | | |
| 43 | Display LIDAR data | 64 days? | Thu 10/2/14 | Tue 12/30/14 | | |
| 44 | Test short-range IR sensor | 5 days | Thu 10/2/14 | Wed 10/8/14 | | Vanessa |



| | | | | | | |
|---|---|---|---|---|---|---|
| 45 | Decide on Sensors-want to use LIDAR | 16 days | Thu 10/9/14 | Thu 10/30/14 | 44 | |
| 46 | Buy Sensor | 30 days? | Fri 10/31/14 | Thu 12/11/14 | 45 | Vanessa |
| 48 | Attach sensors | 2 days | Fri 12/12/14 | Mon 12/15/14 | 46 | Josh |
| 47 | Wire sensors | 3 days | Fri 12/12/14 | Tue 12/16/14 | 46 | Josh |
| 49 | Read distances | 7 days | Thu 12/11/1 | Fri 12/19/14 | 44 | Vanessa,Vu |
| 50 | Map while stationary | 7 days | Mon 12/22/1 | Tue 12/30/14 | 49 | |
| 67 | **Display various sensor data** | **0 days** | **Tue 12/30/1** | **Tue 12/30/14** | **54,43,90** | |
| 90 | Display air quality data | 35 days | Wed 11/12/1 | Tue 12/30/14 | | |
| 91 | Determine necessary data | 7 days | Wed 11/12/1 | Thu 11/20/14 | 10 | Josh |
| 92 | Research and buy sensors | 14 days | Fri 11/21/14 | Wed 12/10/14 | 91 | Josh |
| 93 | Test sensors | 7 days | Thu 12/11/1 | Fri 12/19/14 | 92 | |
| 94 | Get data from sensors | 7 days | Mon 12/22/1 | Tue 12/30/14 | 93 | |
| 68 | Calculate & display robot speed, orientation | 60 days | Wed 11/26/14 | Tue 2/17/15 | | |
| 72 | Determine orientation | 5 days | Wed 12/17/1 | Tue 12/23/14 | 61 | |
| 69 | Determine speed | 60 days | Wed 11/26/1 | Tue 2/17/15 | | |
| 71 | Convert RPMs to speed | 21 days | Wed 11/26/1 | Wed 12/24/14 | 55 | Vu |
| 70 | Get speed from IMU data | 45 days | Wed 12/17/1 | Tue 2/17/15 | 61 | |
| 79 | Design & manufacture looks-like prototype | 35 days | Wed 12/31/14 | Tue 2/17/15 | 43,54,93 | |
| 80 | Design chassis | 14 days | Wed 12/31/1 | Mon 1/19/15 | | |
| 83 | Design A.Q. module | 7 days | Tue 1/20/15 | Wed 1/28/15 | 80 | |
| 81 | Fabricate chassis | 14 days | Tue 1/20/15 | Fri 2/6/15 | 80,75 | |
| 82 | Assemble chassis | 7 days | Mon 2/9/15 | Tue 2/17/15 | 81 | |
| 84 | Fabricate A.Q. module | 14 days | Thu 1/29/15 | Tue 2/17/15 | 83 | |
| 85 | Research air quality requirements | 120 days | Fri 9/5/14 | Thu 2/19/15 | | |
| 89 | Research clean room standards | 14 days | Wed 11/5/14 | Mon 11/24/14 | | Josh |
| 86 | Air quality competition | 60 days | Wed 10/1/14 | Tue 12/23/14 | 11 | Josh |
| 87 | Talk with A.Q. experts | 60 days | Wed 10/1/14 | Tue 12/23/14 | 11 | Josh |
| 88 | Research AQ patents | 120 days | Fri 9/5/14 | Thu 2/19/15 | | Vanessa |



| | | | | | |
|---|---|---|---|---|---|
| 74 | Design & manufacture custom PCB | 42 days | Wed 12/31/14 | Thu 2/26/15 | 43,54,93 |
| 75 | Design PCB | 14 days | Wed 12/31/1 | Mon 1/19/15 | |
| 76 | Fabricate PCB | 7 days | Tue 1/20/15 | Wed 1/28/15 | 75,80 |
| 77 | Populate PCB | 7 days | Thu 1/29/15 | Fri 2/6/15 | 76 |
| 78 | Test PCB | 14 days | Mon 2/9/15 | Thu 2/26/15 | 77 |
| 95 | **Hardware prototypes** | **0 days** | **Thu 2/26/15** | **Thu 2/26/15** | **74,79** |
| 73 | Calculate & display robot position | 30 days | Wed 2/18/15 | Tue 3/31/15 | 70,71 |
| 51 | Display map with robot, environment, and A.Q. data | 21 days | Wed 4/1/15 | Wed 4/29/15 | 73 |
| 53 | Set Max Speed | 14 days | Wed 4/1/15 | Mon 4/20/15 | |
| 52 | Map while driving | 21 days | Wed 4/1/15 | Wed 4/29/15 | 19,72 |
| 96 | **Software** | **0 days** | **Wed 4/29/1!** | **Wed 4/29/15** | **68,73,51** |