Electrical and Computer Engineering Capstone Project Proposal

Version 2.1

3/18/2015

**Erebus Labs**

**Open Sensor Platform**

Submitted By:

**Colten Nye**

**Steve Peirce**

**Golriz Sedaghat**

Supervisor:

**Dr. Lisa Zurk**

Sponsors:

**Dr. Mike Borowczak**

**Dr. Andrea Burrows**

Contents

[Abstract 3](#_Toc414486796)

[Introduction 3](#_Toc414486797)

[Sponsor 3](#_Toc414486798)

[Problem Description 4](#_Toc414486799)

[Previous Attempts 4](#_Toc414486800)

[Existing Products 5](#_Toc414486801)

[e-Health Shield 5](#_Toc414486802)

[NODE 5](#_Toc414486803)

[ArduIMU 6](#_Toc414486804)

[e-Health Shield 6](#_Toc414486805)

[NODE 6](#_Toc414486806)

[ArduIMU 6](#_Toc414486807)

[Project Statement 6](#_Toc414486808)

[Specifications 7](#_Toc414486809)

[Methodology 7](#_Toc414486810)

[Hardware 9](#_Toc414486811)

[Main board 9](#_Toc414486812)

[Sensor Boards 10](#_Toc414486813)

[Firmware 11](#_Toc414486814)

[Schedule 13](#_Toc414486815)

[Budget 16](#_Toc414486816)

[Revision History 17](#_Toc414486817)

# Abstract

In this project we are building an open-source sensor platform which targets K-12 students. The platform communicates with modular sensors, and stores the collected data by sensors on an SD card. The user is able to initialize, configure and program the sensors via a user interface. The critical feature which we need to minimize is the cost and power consumption of the system, as well as the usability of the device by K-12 students, their teachers and parents.

# 

# Introduction

An understanding of science and technology is necessary for anyone who wishes to make informed choices about issues in todays’ society. Recently there has been a significant focus on beginning technical education in Science, Technology, Engineering and Mathematics (STEM) at an earlier age. Efforts have been made to promote interests in STEM related fields among the students and encourage them in considering an STEM career in the future as well as providing the necessary background skills.

This document proposes the design and implementation of an open-source sensor platform; allowing a target audience of K-12 students and their teachers to collect and analyze data. This is accomplished by either employing the supported sensors associated with the platform or connecting their own sensor modules via an open-source interface specification should the desired sensors not yet be supported. Once collected, students are able to process and analyze the data using their preferred methods.

## Sponsor

The focus of our sponsor, *Erebus Labs & Consulting LLC*, is on the development of Secure Hardware/Software solutions with a focus on Educational Outreach and STEM integration in K20 classrooms. In this respect, one of the concerns has been regarding K-12 students using sensor platforms which often require programming/electronics experience which is an impediment to the use of such devices. Beyond of which there are virtually no affordable approaches to the sensor designs available for K12 students and their respective classroom budgets.

## Problem Description

Current sensor platforms are either completed, closed source, and expensive, or partially open, non-customizable, and trivial. The goal is to build an open source, fully contained sensor platform / collection system for $25-$50 USD for at-volume runs. The platform must support the ability to add modular physical sensors.

## Previous Attempts

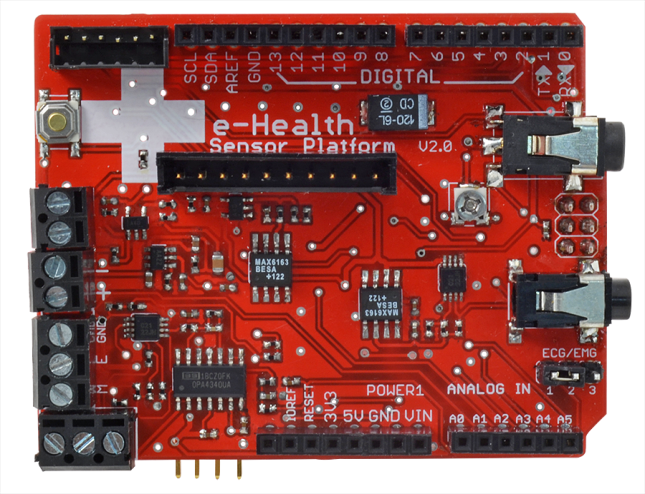
There has been previous related work done in the field by a Portland State University capstone team last year, 2013-14, which would be considered as a first revision proof of concept for this design. However, the implementation was highly complex and required experience in not only circuit and software design, but also the Linux operating system. Many students and teachers alike lack these prerequisite skills, therefore defeating the very purpose of the project in a usability sense.

## 

## Existing Products

We explored existing products that may solve the problem. The common issue with the devices we found is that they are prohibitively expensive. The other issues of the existing devices are listed in table 1.

### e-Health Shield

The e-Health Sensor Shield allows Arduino and Raspberry Pi users to perform biometric and medical applications by using 10 different sensors. Biometric information gathered can be wirelessly sent using Wi-Fi, 3G, GPRS, Bluetooth, 802.15.4 and ZigBee depending on the application. The complete kit is available for € 450 ($512 approximately).

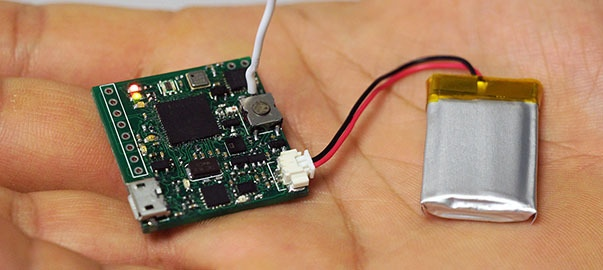
### 

### NODE

The NODE Sensor Platform allows students to explore concepts in science. Data can be collected using Vernier’s Graphical Analysis on iOS or Android devices. The NODE Sensor Platform includes accelerometer, gyroscope, and magnetometer. Two expansion ports allow students to connect additional sensor modules. The platform is available for $149. Not only is this product outside of the target price range, it is not open-source, as is required by the sponsor.

### 

### ArduIMU

ArduIMU V4, An Arduino Based Integrated Measurement Unit, is a fully open source complete wireless sensor module. That gives access to SPI, I2C, UART, analog input and PWM output, which would allow developers to design and develop their custom shields. ArduIMU V4 is available for $129.95, which is again outside the target price range.

|  |  |
| --- | --- |
| e-Health Shield | * High cost * Target audience (not necessarily useful for K12 students) |
| NODE | * High cost per unit * Closed design * Only accessible on devices with closed Vernier software |
| ArduIMU | * High cost per unit * Steep learning curve |

Table 1. Issues with current designs

# Project Statement

Our goal is to design and prototype an open-source hardware/software solution for collecting data from sensors that is both inexpensive and accessible to non-technical users. This solution will initially consist of the main device, a web application for programming the device, and two types of peripheral sensor boards.

The main device will be a board that we design that includes an STM32F205 microprocessor, an SD card slot for storing the collected data, four exposed ADC interfaces, one exposed I2C bus, one micro USB port for supplying power and establishing a serial connection to a host computer, various led indicators, a programming mode button, a power switch, and various circuit protection components.

The sensors boards will also be designed by us. Each sensor board simply carries one sensing component and any components necessary for it to operate and communicate. One will be a volatile organic compound sensor, which is an analog type device, and the other an accelerometer, which is a digital type device, meaning it is connected to the I2C bus.

The web application will be implemented as a Chrome App, which is platform independent and can be run locally (without an internet connection). The purpose of the app is to collect information about how the user intends to use the device, such as which types of sensors are attached to which ports, and how often they would like them sampled. After the information is collected, the user will transfer the configuration to the device by connecting the device to the host computer, then booting it into IAP mode (In-Application-Programming), and transferring the configuration via serial connection over the USB cable.

We will consider the project a success if we can procure the programming application, at least 2 working main boards, 4 working sensor boards (2 of each aforementioned type), and demonstrate successful system integration by May 6, 2015.

# Specifications[[1]](#footnote-1)

|  |  |
| --- | --- |
| **Category** | **Specification** |
| Configuration | Ability to communicate with:   * I2C sensors (I2C sensor boards are stackable) * Up to 4 analog sensors |
| Power | * 3.3 V operating voltage * USB Connector 5V * 3x1.5 AA Battery |
| Clock Speed | 16 MHz |
| Memory | * Micro SD Card connector (stores data collected by sensors) * Microcontroller has Up to 1 MB Flash * Microcontroller has 128+4 kB RAM |
| Communication | * I2C (Inter-Integrated Circuit) * ADC (Analog to Digital Converter) * USB (Universal Serial Bus) * SWD (Serial Wire Debug Interface) * USART (Universal Asynchronous Receiver/Transmitter) |
| User Interface | Web application |
| Supported Operating Systems | * Windows * Mac * Linux |
| Programming | JTAG In-Circuit Programming |
| Customizability | Main board can be customized over SWD port |
| Cost | $50 for main board |

Table 2. Main board design specifications

# Methodology

Figure 1 demonstrates a high level overview of the system. One of the main considerations in the design would be the sensor and main board interface. The system should be able to identify and support different types of communication via I2C or analog input via ADC. The main board contains a microcontroller which communicates with the sensor boards, reading the collected data from the sensors on a predetermined interval, via the browser configuration tool.

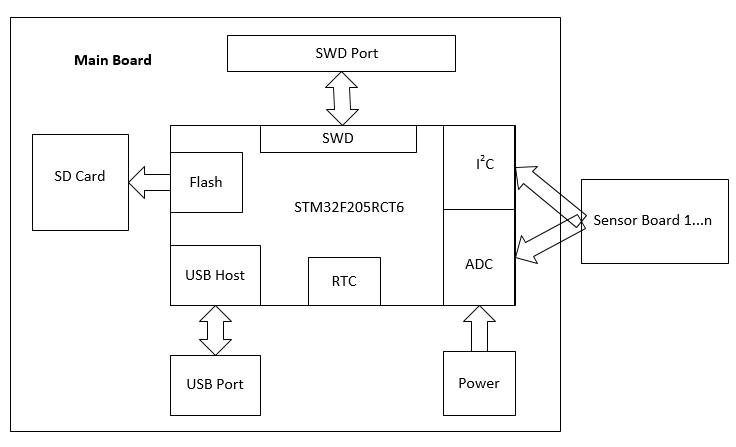


Figure 1. High Level System Design

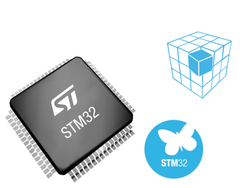
Sampled data is logged to an SD Card for safe storage after power-loss or in case of a physically damaging event. Sensor boards are designed to be “stackable” for I2C (Inter-Integrated Circuit) devices as they are uniquely addressable as per I2C specification. Should the user choose to employ multiple of the same sensor, a 3-bit DIP switch is provided on the I2C sensor board to set the appropriate address. If the respective IC supports this feature it would allow up to sixteen (16) uniquely addressable I2C devices to be connected via two (2) buses.

The ADC channels have a predetermined “snap-in” connector type thereby limiting the possibility of damage due to incorrect installation; the initial revision will support at least 3 ADC sensors. Another possible source of revenue is the production and sale of the “breakout” boards either by Erebus-labs or a licensed third-party.

## 

## Hardware

### Main board

The MCU for this revision is the STM32F205X by STMicroelectronics. The processor includes multiple capabilities useful for this application including multiple I2C, USB, and SPI (Serial Peripheral Interface) buses, low power consumption, on-board SDIO (Secure Digital Input Output), DMA (Direct Memory Access), and the STMCube software platform for configuring initialization code as well as adhering to the Cortex Microcontroller Software Interface Standard (CMSIS) industry standard. Being a more “powerful” processor, the MCU also allows for feature expansion as future revisions of the project ensue.

|  |  |
| --- | --- |
| STM32F205X | |
| Multiple communication interfaces | * Up to 3 I2C * Up to 3 SPI * On-board SDIO |
| Advanced connectivity | * USB 2.0 full-speed/high-speed device/host/On-The-Go controller with dedicated DMA * Ethernet |
| Large memory capacity | Up to 1MByte Flash Memory |
| Low power operating modes | * 1.8 to 3.6V operating supply voltage * Sleep, Stop and Standby modes * VBAT supply for RTC (Real Time Clock) * Power-ON Reset Circuitry * Power-Down Reset Circuitry * Brown-out Reset Circuitry |
| Different software platforms | * STM32Cube initialization code generator * Cortex Microcontroller Software Interface Standard (CMSIS) |
| Other peripherals | * Up to 17 Timers * Three 12-bit Analog to Digital Convertors /Two 12-bit Digital to Analog Convertors * Camera interface * Random number generator |

Table 3. Microcontroller Features



### Sensor Boards

|  |  |
| --- | --- |
| Accelerometer Sensor | ADXL345 (Analog Devices) |
| Communication Interface | I2C |
| Operating Voltage | 3.3V |
| Low power consumption | * Minimum of 23 µA in measurement mode * Minimum of 0.1 µA in standby mode |
| Application | To measure the static acceleration of gravity and dynamic acceleration resulting from motion |
| Other characteristics | * Availability of a DIP switch on ALTERNATE ADRRESS pin to configure two I2C addresses * User selectable, fixed and full resolution modes * Compatibility with Arduino accelerometer Breakout |

Table 4. Accelerometer Sensor Board Specifications

|  |  |
| --- | --- |
| Volatile Organic Compound (VOC) Sensor | TGS2602 |
| Communication Interface | ADC interface |
| Operating Voltage | 5V |
| Application | * VOC monitor * Ventilation control |
| Other characteristics | * High sensitivity to VOC’s * Low power consumption |

Table 5. VOC Sensor Board Specifications

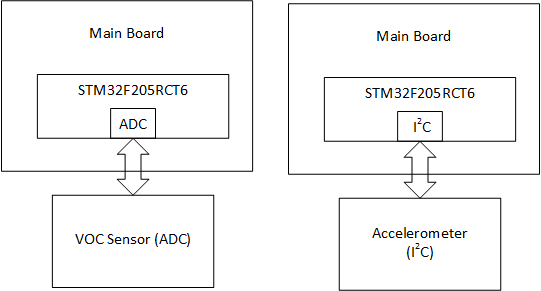


Figure 2 – Sensor boards connections to the main board

## Firmware

Figure 3 shows how the STM32F2X is programmed and debugged using JTAG ICP interface through CooCox CoIDE, OpenOCD or Keil uVision development environments. Although the sensor platform is an open source design, the choice of development tools is not all free of charge. It is up to the user what they would like to invest in more elaborate development suites. This programming is done “in-house,” verified prior to shipment and referred to as “In Circuit Programming.” It must be done at least once and requires additional equipment to do effectively, such as the Olimex ARM-USB-OCD-H ($70) or Keil ULINK Pro ($400+) JTAG units. This bootloader firmware allows the HID connection over USB to support the browser-based editing.

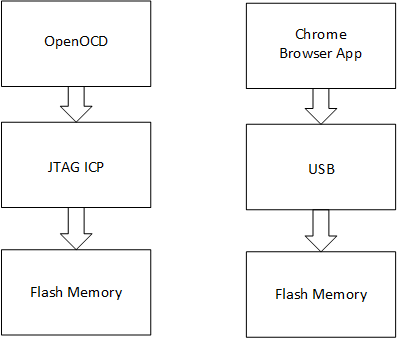


Figure 3. Main board design process (left), User interface design process (right)

However, a more advanced user is still able to customize the bootloader using a JTAG unit, but there is no need for the user to modify the bootloader while using the platform. This design allows the user to interact with the system and program the sensor types, intervals and values independent of the operating system through the browser application provided, (without charge) in the Google Play Store. Figure 2(b) demonstrates the user interface design of the Sensor Platform; user is able to connect to the system via USB, and program the sensor values through a Chrome based web browser application, via STM’s supported “In Application Programming,” hereafter referred to as IAP. Furthermore, user is able to save the collected data on a flash memory to prevent data loss, as well as extract the collected data for further analysis.

# Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Task Name** | **Duration** | **Start** | **Finish** |
| **Plan** | Teaming up and Meeting with Advisor | 3.33 days | Mon 11/10/14 | Thu 11/20/14 |
| Understanding the project and its requirements | 15.56 days | Mon 11/10/14 | Wed 1/7/15 |
| Meeting with sponsors | 0.33 days | Thu 12/4/14 | Thu 12/4/14 |
| Reviewing the existing design and gathering info about the requirements | 8.67 days | Fri 12/5/14 | Thu 1/8/15 |
| Meeting with sponsors | 0.33 days | Thu 1/8/15 | Thu 1/8/15 |
| Concept Generation and Evaluation | 0.33 days | Fri 1/9/15 | Fri 1/9/15 |
| Write up the preliminary project proposal | 0.67 days | Sun 1/11/15 | Tue 1/13/15 |
| Review project proposal | 1 day | Wed 1/14/15 | Fri 1/16/15 |
| Present Project Proposal | 1 hour | Tue 2/24/15 | Tue 2/24/15 |
| **Requirement** | Develop the general project requirement | 3.67 days | Thu 1/8/15 | Tue 1/20/15 |
| Write up the detail project requirement | 2 days | Sun 1/25/15 | Wed 1/28/15 |
| Review project requirement document | 1 day | Mon 2/2/15 | Tue 2/3/15 |
| Gain Sponsor approval of requirements | 0 days | Tue 2/24/15 | Tue 2/24/15 |
| **Design** | Determine the necessary functional blocks | 4.33 days | Fri 1/9/15 | Fri 1/23/15 |
| Review the block layout | 0.33 days | Sun 1/25/15 | Mon 1/26/15 |
| Determine the design concept of each block: problem to solve and available technology | 1.71 days | Sun 2/1/15 | Fri 2/6/15 |
| Write the test plan | 2 days | Wed 2/4/15 | Fri 2/6/15 |
| Review the test plan | 0.33 days | Fri 2/6/15 | Sat 2/7/15 |
| Design the preliminary circuitry | 2.67 days | Mon 2/2/15 | Wed 2/11/15 |
| Review the circuit design | 1 day | Thu 2/12/15 | Sat 2/14/15 |
| Generate the circuitry schematic and PCB layout on Eagle CAD | 1.67 days | Mon 2/16/15 | Fri 2/20/15 |
| Review and revise preliminary design | 0.33 days | Fri 2/20/15 | Sat 2/21/15 |
| Finalize the preliminary schematic and PCB design | 2 days | Wed 3/11/15 | Wed 3/18/15 |
| Review the design with the sponsors | 0.67 days | Sat 3/21/15 | Mon 3/23/15 |
| Specify the component for the design | 7 days | Fri 2/6/15 | Fri 2/27/15 |
| Place order for the components | 1 day | Mon 3/23/15 | Wed 3/25/15 |
| **Build** | User Interface | 28.38 days | Thu 3/19/15 | Thu 4/16/15 |
| Finalize configuration parameters | 6.38 days | Thu 3/19/15 | Wed 3/25/15 |
| Load sensor type definitions from file | 4.38 days | Wed 3/25/15 | Sun 3/29/15 |
| CSV | 7.38 days | Mon 3/30/15 | Mon 4/6/15 |
| write sensor -> CSV encoding | 5.38 days | Mon 3/30/15 | Sat 4/4/15 |
| write CSV -> sensor decoding | 3.38 days | Fri 4/3/15 | Mon 4/6/15 |
| Add FileSystem functions | 19.38 days | Sat 3/28/15 | Thu 4/16/15 |
| Save configuration | 3.38 days | Tue 4/7/15 | Fri 4/10/15 |
| Load Configuration | 3.38 days | Tue 4/7/15 | Fri 4/10/15 |
| Add serial functionality | 5.38 days | Sat 4/11/15 | Thu 4/16/15 |
| Hack the serial IAP driver | 10.38 days | Thu 4/16/15 | Sun 4/26/15 |
| Add a menu option for downloading a configuration | 2.38 days | Thu 4/16/15 | Sat 4/18/15 |
| Implement functionality of downloading the configuration | 2.38 days | Sat 4/18/15 | Mon 4/20/15 |
| Parse the configuration into an array of sensor structs | 2.38 days | Tue 4/21/15 | Thu 4/23/15 |
| Save to flash | 2.38 days | Fri 4/24/15 | Sun 4/26/15 |
| Program the MCU | 20.33 days | Thu 1/15/15 | Tue 4/7/15 |
| Send the boards to fab | 4 days | Mon 3/23/15 | Tue 4/7/15 |
| **Poster** | Working on the poster | 60.38 days | Wed 4/1/15 | Sun 5/31/15 |
| **Test** | Debug the codes | 5 days | Wed 4/8/15 | Wed 4/22/15 |
| Build the system on the PCB | 1.67 days | Fri 4/10/15 | Thu 4/16/15 |
| Test the hardware | 4.56 days | Fri 4/10/15 | Fri 4/24/15 |
| Test the Software | 4.33 days | Fri 4/10/15 | Fri 4/24/15 |
| Test the GUI | 1.67 days | Mon 4/27/15 | Wed 4/29/15 |
| Integration test | 1.67 days | Thu 4/30/15 | Mon 5/4/15 |
| Functionality test | 1.67 days | Tue 5/5/15 | Thu 5/7/15 |
| Design and build the device closure (box) | 6.33 days | Fri 4/17/15 | Sun 5/10/15 |
| Perform the final test | 17.96 days | Tue 4/28/15 | Sat 5/16/15 |
| Write the operation manual | 13.96 days | Tue 5/5/15 | Sat 5/23/15 |
| Review the operation manual | 3.17 days | Wed 5/13/15 | Tue 5/26/15 |
| **Present** | Complete the presentation documents (slides) | 5.67 days | Sat 5/9/15 | Fri 5/29/15 |
| Review the presentation documents | 0.33 days | Fri 5/29/15 | Mon 6/1/15 |
| Final presentation of the system | 1.33 days | Mon 6/1/15 | Thu 6/4/15 |

# Budget

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part #** | **Part Name** | **Manufacturer** | **Vendor** | **Qty** | **Unit Cost** | **Cost** |
| TGS2602 | VOC Sensor | China(Mainland) | AliExpress | 5 | $5.95 | $29.72 |
| ADXL345BCCZ-RL7 | Accelerometer | [Analog Devices Inc](http://digikey.com/Suppliers/us/Analog-Devices.page?lang=en). | DigiKey | 5 | $7.55 | $37.75 |
| STM32F205RCT6 | MCU | STMicroelectronics | Mouser | 5 | $9.01 | $45.05 |
| ZX62R-B-5P | Micro USB, type B, male | [Hirose Connector](http://www.mouser.com/hiroseconnector/) | Mouser | 5 | $1.27 | $6.35 |
| 406C35E16M00000 | 16meg Oscillator | CTS Electronic Components | Mouser | 5 | $1.08 | $5.40 |
| STMPS2141STR | Current Limiter | STMicroelectronics | Mouser | 5 | $0.97 | $4.85 |
| RCLAMP0504SCT-ND | USB TVS Array | [Semtech Corporation](http://digikey.com/Suppliers/us/Semtech.page?lang=en) | DigiKey | 5 | $0.91 | $4.55 |
| TPS73233DBVR | 3.3V Voltage Regulator | Texas Instrument | Mouser | 5 | $1.46 | $7.30 |
| PRT-00127 | Micro SD Socket | SparkFun | SparkFun | 5 | $3.95 | $19.75 |
| HSMS-2824-BLKG | Schottky diode | Avago Technologies | Mouser | 5 | $0.89 | $4.45 |
| P4SMA7.5CA-E3/5A | TVS Diodes | [Vishay Semiconductors](http://www.mouser.com/vishaysemiconductors/) | Mouser | 15 | $0.219 | $0.219 |
| NDS352AP | P-Channel MOSFET | [Fairchild Semiconductor](http://digikey.com/Suppliers/us/Fairchild-Semiconductor.page?lang=en) | DigiKey | 5 | $0.60 | $3.00 |
| DSK-3R3H703T414-HRL | Super Capacitor | [Elna America](http://digikey.com/Suppliers/us/Elna-America.page?lang=en) | DigiKey | 5 | $0.75 | $3.76 |
| AYZ0202AGRLC | Sliding Switch | [C&K Components](http://digikey.com/Suppliers/us/C-and-K-Components.page?lang=en) | DigiKey | 5 | $1.33 | $6.65 |
| COM-00097 | Mini Push Button | SparkFun | SparkFun | 15 | $0.35 | $5.25 |
| TC33X-2-104E | Potentiometer | Bourns Inc. | DigiKey | 5 | $0.27 | $1.35 |
| PCB | N/A | OSH Park | N/A | 5 | $15 | $80 |
| $51.56 $265.40 | | | | | | |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Revision Date** | **Author** | **Comments** |
| 1.0 | 1/14/2015 | Golriz Sedaghat | Changes were made in Detailed Problem Description and Methodology, also figure 1 has been modified and figures 2 and 3 has been added |
| 2.0 | 2/27 | Colten Nye | Restructured and updated entire document |
| 2.1 | 3/18/2015 | Golriz Sedaghat | Information was added in abstract, specification, methodology, firmware, schedule and budget sections |
| 2.2 | 3/18/2015 | Colten Nye | Added to the Project Statement section |
|  |  |  |  |

1. More detailed Requirement and Specification document is available at:

   https://github.com/erebus-labs/sense\_platform/blob/master/Project%20Flow%20Documents/Project%20Documents/Requirements%20Specification.docx [↑](#footnote-ref-1)