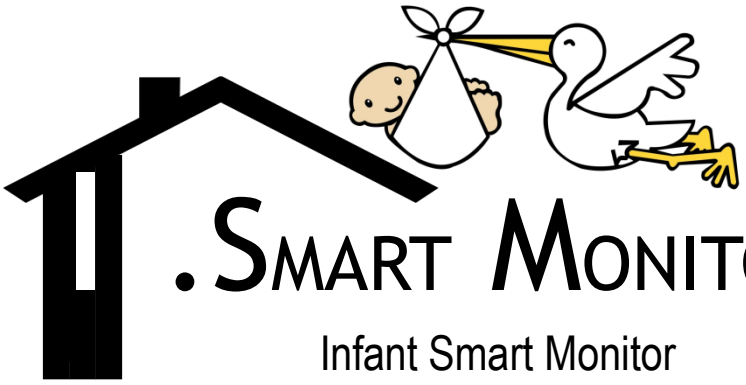


.SMART MONITOR

Infant Smart Monitor

Team 6

Fall 2013 – Spring 2014

End of Project Documentation
Infant Smart Monitor (I.Smart Monitor)
Joseph Cacioppo, Duaa Salah, Daniel Schmidt, Mahsa Shadmani, Vasiliy Warkentin
May 5, 2014

Abstract*—Monitoring the vital signs of a newborn is a fundamental component of modern healthcare. Monitoring can, however, bring its own problems that are very stressful for both infant and parents. These problems arise from the fact that a human is currently needed to do the monitoring. This individual must be well trained and diligent in order to be effective. Effective data collection is obviously vital to ensure that nothing important was missed and that the readings are accurate. Because detecting medically significant events is often a matter of comparing data values to certain thresholds, an electronic device could easily accomplish this. This paper documents the design of such a device, the I.Smart Monitor. It details the design process in two stages: First, the documentation supporting the development of a laboratory prototype that includes the motivation for the project as a solution to a societal problem. Secondly, the development of a deployable prototype and a discussion of what was learned during the first stage and how this was applied. The testing procedures that were performed in order to determine if the device meets established criteria are then outlined. How these criteria were established and what was learned from the testing is then discussed, feature by feature.

Keywords*—Sensor, Microcontroller, HIPAA, onboard storage, Arduino, C-language, EEPROM, Wi-Fi, Ethernet, I^2C bus, Modular sensors, Micro SD card, SPI, Remote Access, World Wide Web, ATmega328, hot-swappable

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

The birth of a child is often a long awaited life-changing event filled with anticipation and wonderment; however, not every birth proceeds as planned. The premature birth of a child leaves the parents filled with overwhelming anxiety that does not dissipate once the infant is released from the hospital. Compared to term infants, premature infants are more likely to suffer from jaundice, respiratory issues and Sudden Infant Death Syndrome (SIDS). Because of this, parents and caregivers of premature infants are often in a constant state of alert that is difficult to maintain. The ability to easily monitor the infant and facilitate communication between parents and doctors can go a long way in easing the parents' understandable anxiety.

From September 2013 to May 2014, Team 6 developed a device to reduce the stress during this time as much as possible. This is the story of the I.Smart Monitor.

II. SOCIETAL PROBLEM AND SOLUTION

Parents of newborns are concerned with the health of their children, especially parents of premature babies. In a study conducted by the Division of Reproductive Health, approximately one out of every eight babies born in the U.S. is preterm.[1] According to mayoclinic.org; being born too early can cause short-term and long-term health problems. Some examples of short-term complications are respiratory, heart, and temperature control problems. Long-term complications can include chronic health issues such as infections, asthma and feeding problems that, in some cases, require constant monitoring. In addition, these infants are at a greater risk of SIDS; approximately 4,000 infants die each year of SIDS in the United States.[2] In some cases, early intervention may have prevented some of these deaths. Some current methods used to detect these conditions involve monitoring the breathing, blood pressure and heart rate of an infant constantly. Unfortunately, this monitoring does not end when the baby is discharged from the hospital; premature infants need to be monitored even after coming home.

Taking care of premature infants is extremely difficult and challenging for the parents. Because of this, many families relish the return to the privacy and comfort of their own home. This is especially beneficial to the infants, as well. Infants in a warm, intimate environment, where they are spoken to and held by their parents, emotionally and physically thrive better than infants who spend the beginning of their life in a sterile environment, such as time alone in an infant incubator.[3]

By providing a way to monitor their baby's vital signs at home, parents can bring their baby home sooner to begin feeling like a 'real' family. Offering a home-use device that monitors vital signs increases the potential for parents to initiate early intervention by monitoring the infant's vital signs and activating an alarm when any vital sign falls outside of a range predetermined by medical industry standards.

A remote monitoring function can also alleviate strain on the health care system. When parents can view their infant's vital statistics in an easy and understandable format, excessive office visits will be reduced. Primary care physicians estimate that at least 10 percent of office visits are unnecessary and cutting these unnecessary visits in half would save overburdened doctors almost one half hour per day.[4] If the parents continue to be concerned, they can contact their pediatrician through the

website's messaging service. The pediatrician can view near real-time and historical vital sign data and recommend an office visit or assuage the parents' concerns about the health of their infant.

Figure II.1—Overcrowded Maternity Ward in China

The I.Smart Monitor will provide this service and transmit vital data to the pediatrician remotely. With over 10 percent of babies in the U.S. born prematurely, our monitor will provide peace of mind and has the potential to save lives, through both the alarm and the event recorder features. [5] The alarm feature will handle more immediate concerns, while the event recorder will help aid in diagnosis. The I.Smart Monitor team strongly believes this will help alleviate the stress and impact of this societal problem. To implement this solution, a device with multiple features corresponding to specific aspects of the mentioned societal issue, was designed.

Figure II.2—Overcrowded Maternity Ward in Los Angeles, CA

III. DESIGN IDEA

The I.Smart Monitor is comprised of a specialized central hub, which collects, stores, time stamps and transmits data wirelessly to the internet through a local area network. The monitor also consists of a variety of non-invasive sensors that are specifically designed to be safe and comfortable for the infant. Each sensor is a stand-alone unit that measures a particular vital sign and begins working when connected to the central hub. Each sensor includes a microcontroller, which allows the Smart Sensors to perform computations and processing. This means the hub itself is very simple; its processing requirements are constant, regardless of the number of sensors connected. The sensor controller allows for a "plug and play" type feature, which means ease of use for the parents. Also, the hub is able to send alarms to users through an audible indicator. Since the device uses the industry standard I^2C bus to communicate between the sensor controller and the hub, future sensor development and improvements will be easily compatible with the monitor. The modular design of the Smart Sensors allows the customer to purchase only the sensor type needed; however, should their needs change, they can simply purchase an additional sensor. There is no need for them to purchase a completely new system. This will result in cost savings to the customer. The modular design also allows the device to be adaptable to new monitoring capabilities and future sensor improvements, also saving the parents the cost of having to purchase the latest and greatest system while still providing them with the best new technology.

Moreover, the I.Smart monitor is providing the pediatrician with remote

access to near real-time and historical data. Ideally, this information can be viewed via the internet on our secure website for authorized users and can be displayed in two different formats: an easy to understand format for parents and a comprehensive format for their pediatricians and conforms to all HIPAA laws. Current devices only consist of a single type of sensor, which cannot be changed, and an audible alarm. Therefore, the novelty of our device is twofold. Our design allows a single device to be used in all situations with merely a different sensor module connected. The device will also allow for remote monitoring.

The initially stated design idea involved multiple sensors of different levels of sophistication. This has been discovered to be a misallocation of time. The innovation of the I.Smart Monitor lies in the method of delivering the sensor data to the parents and doctors, not in any particular one of the sensors itself. For this reason, the design idea has been modified to focus on this system, with only one type of sensor constructed multiple times in order to show the system works. The particulars of the feature set have not changed and are discussed in the following section.

IV. FEATURE SET

Making biometric measurements is inherently complex. Because of this, the following five features were developed to allow non-technical individuals to use the I.Smart Monitor:

A. Modular Sensors

The device is designed to be able to accommodate multiple sensors connected simultaneously. No configuration is required, to keep training to a minimum. Sensors can be connected or disconnected at any time or in any order, and the device will adjust accordingly without loss of functionality. This greatly improves the ease of use.

For this feature, the I^2C protocol is used. This protocol involves a master-slave dynamic in which the central hub (the master) requests and collects data packets from the sensors (the slaves). I^2C uses three wires, two lines for communication (SDA, SCL), plus a ground connection, as can be seen in figure IV.1

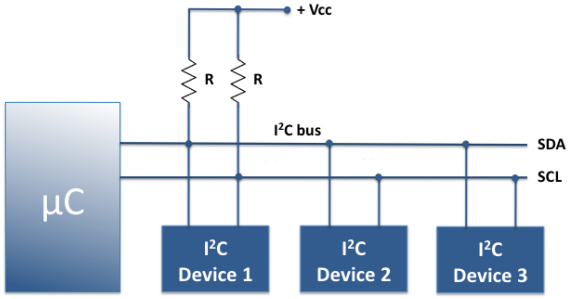


Figure IV.1—I^2C Bus Block Diagram Showing SDA and SCL Connections

The I.Smart Monitor will use the three standard I^2C wires as well as an additional wire to supply 5 VDC. This feature is implemented as follows: the hub constantly polls the bus for all possible address numbers, which are selected at random by each sensor when first connected. If the address number that the hub is currently polling is taken by a sensor, the sensor will return a confirmation. If a sensor selects an address that has already been selected by another sensor, no conflict will occur as the data will still be distinguished by its serial number. The hub will then request a data packet from the sensor. The data packets consist of five bytes: a one-byte ID which indicates sensor type, a unique two-byte serial number (15-bits for the number, with

the MSB as an alarm bit) which distinguishes one sensor from another of the same type, and two bytes of biometric data. The data is pre-processed by the slaves before transmission and so all packets are the same size, regardless of sensor type. The hub will then store this data to a storage device.

B. Event/Data Logger

The I.Smart Monitor will record data from each packet it receives. This is essential for any medical monitoring device and especially for one used in the home, away from medical experts. In addition, any device used to monitor a newborn should have the ability to record events of medical significance, if it is going to aid in diagnosis. For example, a baby's heart rate dropping below 120 beats per minute is not necessarily life threatening on its own, but a pattern could indicate a more chronic condition. An event recorder would give a clear indication of this and potentially assist in early diagnosis.

Upon receiving a data packet from the sensors, the device will write this data to a micro Secure Digital (SD) card.

Figure IV.2—MicroSD Card

The data stored on the SD card are: time-stamped data, sensor bus ID, sensor ID, sensor serial number, and sensor data. The SD card will act as both a recorder by itself, but also as a spool while data is uploaded to the internet—the rate of data coming in may sometimes exceed the rate at which it is uploaded. The SD card, with a capacity of 16 GB, is capable of storing sensor data for approximately five months in a worst case scenario for 127 simultaneous sensors. Because the device is intended for infants in the age range of only one to two weeks, 5 months is more than enough storage. Any class of SD card will work; the I.Smart Monitor uses class 10 and can store data at the rate of 10 MB per second. When it reaches 80% capacity, it will sound an alarm. When it is at 95%, it will stop recording continuously and only store alarming events of medical significance. When at 100% capacity, it will begin overwriting the oldest data. Table-IV.1 below illustrates SD capacity situations.

Table IV.1—SD Capacity Situations

SD Capacity Situations	Results
Reaches 80% capacity	Alarm will sound
Reaches 95% capacity	Record alarming data such as; time-stamped, sensor ID, sensor data, alarm bit.
Reaches 100%	Overwrite oldest non-alarming data on the SD card

The SD card is backed up by a 1 MB EEPROM which will not be removable but will prevent data loss if the SD card fails for any reason.

Figure IV.3—1 MB EEPROM

If the SD card fails or is removed, the device will begin writing to the EEPROM. Each data will be stored on the EEPROM.

In a worst case scenario situation, this can continue for 15 minutes with 127 connected sensors before the EEPROM reaches capacity; at which time data will be lost. If the SD card is replaced before this occurs, the data on the EEPROM will be immediately uploaded to the SD card. Because the data is time stamped, it will be placed in chronological order before being uploaded to the server.

If the SD card is not replaced, and the EEPROM gets to 90% capacity, an LED will blink. When the EEPROM is 100% full and the SD card is not connected, an alarm will sound and it will stop recording. Table-IV.2 below illustrates the EEPROM capacity situations.

Table IV.2—EEPROM Capacity Situations

SD Capacity Situations	Results
Reaches 90% capacity	LED will blink
Reaches 100% capacity	Alarm will sound and it will stop recording

After the data are recorded to the SD card, it will be transferred to the server. Before this occurs, however, the packet is checked for an indication of an alarm condition.

C. Alarm

The primary sub-feature of the alarm is the ability to indicate a medical emergency or other condition of interest in a way that is easily noticeable. This is, of course, dependent on the ability to detect said medical condition. The criteria for what is and is not a "medical emergency," or a "condition of interest," for a particular class of measurements (temperature, blood-oxygen saturation, etc.), has been established by the medical industry and a few examples can be seen in Table IV.3.

Table IV.3—Examples of Industry Standards

Measurement	Normal range for newborn
Axillary temperature	97.5 – 99.3 ^{o} F
Pulse rate	70 – 190 bpm
Respiratory rate	30 – 60 breaths-per-minute

As stated previously, the data packet sent from the sensors will have an alarm bit contained within the serial number. This bit will simply be set by an alarm condition and reset otherwise. The sensor determines whether such a condition has been met and relays this to the hub. Each type of sensor utilizes different criteria, set by medical standards. For example, an axillary skin temperature of less than 97 ^{o}F or more than 100 ^{o}F will trigger an alarm. Upon receiving a set alarm bit, the hub will activate an audible alarm and record the event as alarming data. As this device is designed for use in the home, it must be able to work with the home network of the user. The details of this ability are discussed in the next section

D. Home Network Connectivity

The Home Network Connectivity feature of I.Smart Monitor supports the ability of the device to connect to the local area network. This feature provides both wired and wireless connections to an existing home network. In the final prototype, the I.Smart Monitor is using an Arduino YUN

microcontroller which has built-in Wi-Fi and Ethernet support. The Wi-Fi and Ethernet features are able to provide IEEE 802.11b/g/n and IEEE 802.3 10/100Mbit/s, respectively. [6]

The design needs to follow the standard frequency defined and approved by the U.S. Food and Drug Administration (FDA), for the Radio Frequency Wireless Technology in Medical Devices. [7] According to fda.org, IEEE802.11 is one of the appropriate RF wireless technologies and is also supported by the Arduino YUN microcontroller.

Table IV.4—Wireless Standards & Frequencies

Standard	Frequency	Data Rate	Range
Inductive Coupling	< 1 MHz	1-30 kbps	< 1 m
Wireless Medical Telemetry System	608-614 MHz, 1395-1400 MHz, 1427-1429.5 MHz	250 kbps	30.60 m
Medical Device Radio Communication Service (MICS)	401-406 MHz	250 kbps	2-10 m
Medical Micropower Networks (MMNs)	413-419, 426-432, 438-444, 451-457 MHz		< 1 m
Medical Body Area Networks (MBANs)	2360-2400 MHz	10 kbps - 1 Mbps	< 1 m
802.11a Wi-Fi	5 GHz	54 Mbps	120 m
802.11b Wi-Fi	2.4 GHz	11 Mbps	140 m
802.11g Wi-Fi	2.4 GHz	54 Mbps	140 m
802.11n Wi-Fi	2.4 – 5 GHz	48 Mbps	250 m
802.11.1 Bluetooth Class I	2.4 GHz	3 Mbps	100 m
802.11.1 Bluetooth Class I	2.4 GHz	3 Mbps	10 m
802.11.4 (Zigbee)	868, 915 MHz, 2.4 GHz	40 kbps, 250 kbps	75 m
World Interoperability for Microwave Access (WIMAX)	2.5 GHz	70 Mbps (fixed) 40 Mbps (Mobile)	Several km

Also, based on the below table provided by mddionline.com, the standard frequency for 802.11b/g/n Wi-Fi technology is 2.4GHz, which is supported by the Arduino YUN as well.

Shown in Table IV.4 are common wireless standards and RF frequencies for wireless medical product designs targeted for U.S. medical devices. [8]

In addition to Wi-Fi, Ethernet is also supported. This allows the system to provide wired communication between the I.Smart Monitor and the local area network.

Above all, one of the most important criteria in the wireless communication is the security issue. The Health Insurance Portability and Accountability Act of 1996 (HIPAA) requires the U.S Health and Human Services to develop regulations protecting the privacy of certain health information. [9] To fulfill this requirement, the device needs to have WPA or WPA2 encryption support. The Arduino Yun is able to support both WPA, and WPA2. WPA, Wi-Fi Protected Access, and WPA2, Wi-Fi

Protected Access II, are security protocols developed by the Wi-Fi Alliance, an association that promotes Wi-Fi technology and standardizes Wi-Fi to make one device compatible with others.

To make the data available remotely, remote access is required. Hence, our next feature.

E. Remote Access

The sensor data, once uploaded to the fileserver, will be stored securely. The file names correspond to the sensor's serial number and are uploaded to the server using a secure SFTP protocol. A webserver interface reads the files from the fileserver and gives users with proper credentials, access to the data. These users would most likely be the parent or caregiver, and a doctor or member of the hospital staff. The latter will see a more detailed, technical presentation of the data, while the former will see a more intuitive, plain English display. The data will be accessible from a mobile device and formatted accordingly to make the data easy to read for users.

The five features discussed above were implemented in two stages. The first was the development of a laboratory prototype which served as a proof of concept and a platform to determine how the project could move toward the development of a deployable prototype, which is the second stage. The following two sections discuss these two processes.

V. CREATION DETAILS FALL 2013

Initially, this project was focused on a sensor; specifically, a sensor that would give an early indication of infant jaundice. This would have been a device that took a measurement of transcutaneous bilirubin levels, processed this data, and then displayed it in an easy to understand fashion. This idea was scrapped, as the team realized that it would be unsuitable for the intended scope of the project. This device would have been utterly dependent on the ability to make a proper transcutaneous bilirubin measurement. This involves complex physics concepts and fine calibration of LED emission wavelengths which were judged to be insurmountable problems, at least within the time allotted. After discussion with advisors, the project definition was modified and generalized to focus on the interface with the sensors rather than the sensors themselves. This has proven to be a very advantageous decision as the new project has been much more suited to the team members' skill sets.

In addition to the design of the I.Smart monitor, Team 6 was involved in the Idea-to-Product competition—a state-wide event for bio-medical engineering student projects. The competition itself consisted of a presentation of an idea and the marketing strategies involved in deploying the product. Upon entering, our team acquired two business students to assist in the presentation and the development of our marketing strategy. The competition took place in January 2014, in Santa Clara, California. Throughout the semester, Team 6 met with their College of Business student contacts, Terry Petlowany and Ravi Singh and received their assistance on research, especially where business skills were beneficial.

A. Funding

Table V.1—Funding Proposal Fall 2013

Item	Quantity	Unit Cost	Total Cost
Wi-Fi Shield for Arduino	1	$88.81	$88.81
Miscellaneous IC's		$23.70	$23.70
4066 chips	1	$2.40	$2.40
microSD Card	1	$19.99	$19.99
1 MB EEPROM	1	$6.73	$6.73
Hardware – Chassis, Serial connectors, IC sockets		$48.20	$48.20
RTC module	1	$15.00	$15.00
Arduino UNO + LCD + Ethernet	1	$86.68	$86.68
Arduino YUN	1	$85.00	$85.00
Pulse Sensor	2	$24.95	$47.44
LTE-302 IR Sensor	10	$0.33	$3.31
LTE-302 IR Emitter	10	$0.33	$3.31
Op-amps LM356	10	$0.35	$3.52
FTDI breakout board for ATmega328	2	$14.95	$29.90
16 MHz crystal oscillators	5	$0.95	$4.75
ATmega328	5	$5.50	$27.50
Office Supplies (Binder, paper, etc.)			$25.00
Total			$538.70

B. Project Milestones

Task Name	Start	Finish	Student
Fall Semester	**Tue 9/3/13**	**Thu 12/19/13**	
Modular	**Tue 10/1/13**	**Thu 12/12/13**	**Daniel & Joe**
Event/Data Logger	**Tue 10/8/13**	**Mon 12/2/13**	**Duaa**
Home Use	**Mon 10/14/13**	**Mon 12/9/13**	**Mahsa**
Alarm	**Fri 11/8/13**	**Mon 11/25/13**	**Daniel**
Project Management	**Tue 9/3/13**	**Mon 12/16/13**	**All**
Design Idea	Tue 9/3/13	Tue 9/10/13	
Breadboard Proof	Tue 9/24/13	Tue 10/8/13	
Work Breakdown Structure	Tue 9/10/13	Tue 9/17/13	
Project Timeline	Tue 9/17/13	Tue 9/24/13	
Create Laboratory Prototype	Wed 11/13/13	Tue 11/19/13	
Working Laboratory Prototype	Wed 11/13/13	Thu 12/12/13	
Team Leader 1 - Joe	Tue 9/10/13	Mon 10/28/13	
Team Leader 1 Report	Tue 9/10/13	Mon 10/28/13	
Team Leader 2 - Mahsa	Tue 10/29/13	Tue 12/10/13	
Team Leader 2 Report	Tue 10/29/13	Tue 12/10/13	
Team Member Evaluation	Mon 11/11/13	Mon 11/18/13	
Promote Speaking Proficiency	**Tue 9/3/13**	**Mon 12/16/13**	**All**
Problem Statement Presentation	Tue 9/3/13	Tue 9/10/13	
Feature List Presentation	Wed 9/11/13	Tue 9/17/13	
Breadboard Presentation	Tue 10/1/13	Tue 10/8/13	
Mid-term Technical Review Presentatic	Tue 11/5/13	Tue 11/12/13	
Laboratory Prototype Presentation	Tue 12/3/13	Tue 12/10/13	
Promote Writing Proficiency	**Tue 9/3/13**	**Mon 12/16/13**	**All**
Problem Statement Report	Tue 9/3/13	Tue 9/10/13	
Design Idea Contract	Wed 9/11/13	Tue 9/24/13	
End-of-term Documentation	Tue 11/12/13	Thu 12/5/13	

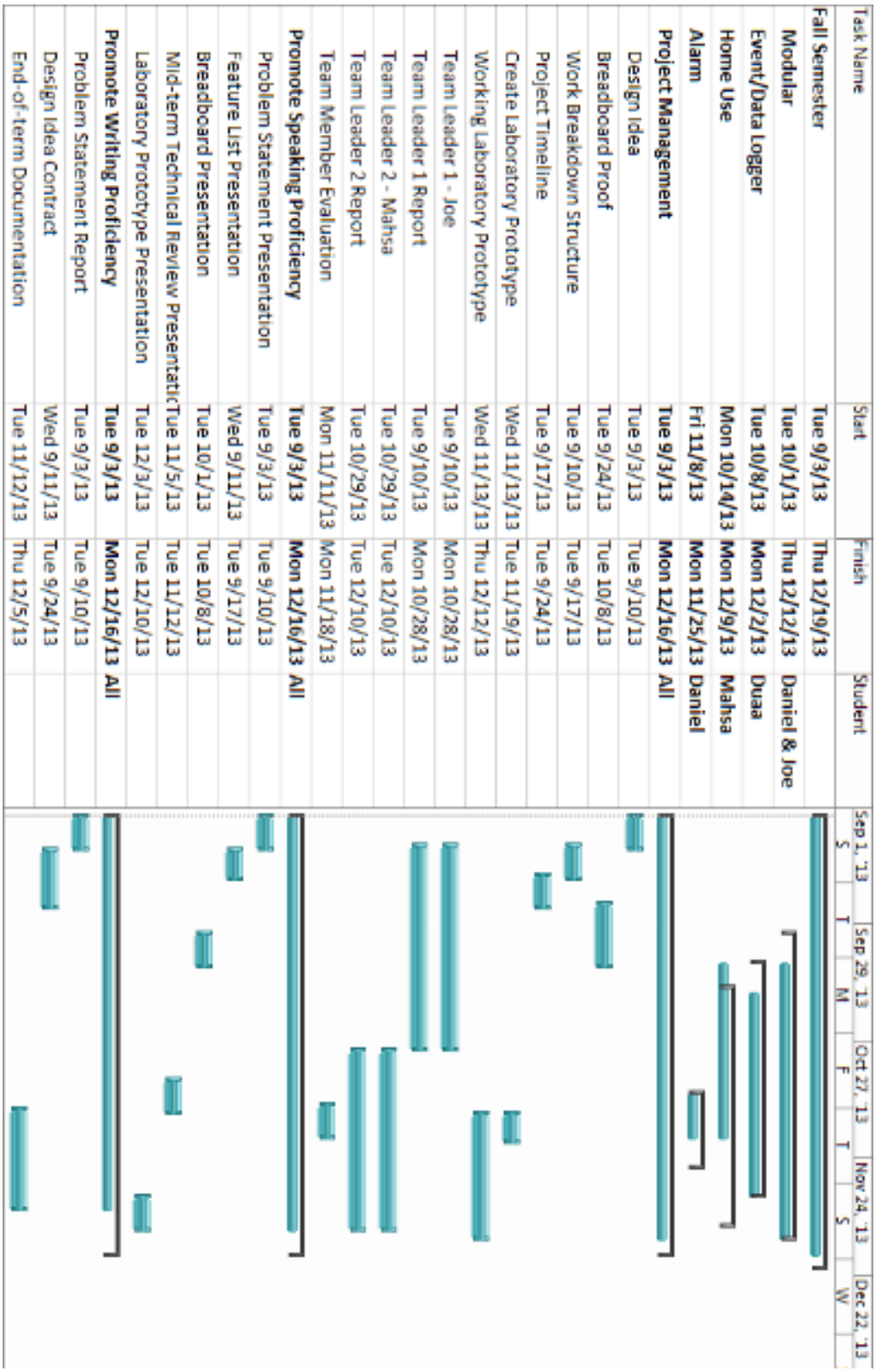


Figure V.1—Project Milestones Fall

C. Work Breakdown Structure

This section represents the work breakdown structure of the project over the fall semester. It explains how each feature was implemented, the time needed to complete each feature, as well as the scope of work, the budget, and the team member responsible for each part.

As said above, our project consists of five features: Modular Sensors, Event/data Recorder, Home Network Connectivity, Alarm, and Remote Access, as shown in figure V.2 below:

Modular Sensors: the I.Smart Monitor incorporates a modular sensor design. There were two tasks to implement modular sensors. The first task (2.1) dealt with multiple simultaneous sensors. To allow this, the sensors are smart, and each has its own identity. There were two types of sensors; basic sensor (2.1.1), intermediate sensor (2.1.2). Each sensor monitors and reads its own data. The second task (2.2) dealt with the microcontroller used to detect these sensors. The microcontroller was able to distinguish between these sensors and was be able to collect (2.2.1), interpret (2.2.2) and send data (2.2.3). The cost was approximately $194 and took around 194 hours to complete it. Daniel and Joseph were responsible to do the first task (2.1), Vasiliy was responsible with (2.2.1) task, Duaa did (2.2.2), and Mahsa did (2.2.3).

Alarm: the device has an alarm that will alert both parents and doctors if a medical emergency is detected. The tasks for the alarm feature were: receiving data (5.1), detecting threshold points (5.1.1) for each sensor, and then sending an alert signal (5.1.1.1). The cost of an audio alarm circuit is around $5, and this feature took around 10 hours to implement. Daniel worked on task (5.1) and Vasiliy worked on (5.1.1).

Event/Data Logger: There were two main tasks for the Event/Data Logger. The first task (4.1) was communication with the server, which included the ability to write to the server (4.1.1). There were two methods used for writing to the server: wirelessly (4.1.1.2) and wired (4.1.1.1). The second task (4.2) was dealing with onboard storage. The device can handle two types of storage: removable storage (4.2.1.), and embedded storage (4.2.2). Each one of the storage types is connected to the microcontroller in order to accomplish two activities: read data from and write data to the storage. The cost was around $112 for SD card, EEPROM and Arduino, and took around 35 hours to complete. Vasiliy worked on the server (4.1), while Duaa worked on (4.2)

Home Network Connectivity: this feature was implemented by breaking it into two tasks as shown in figure V.2a. The first task was to configure the microcontroller to be able to connect to a home network (1.1) and either communicates wirelessly (1.1.1.1) to send (1.1.1.1.1) and receive data (1.1.1.1.2), or wired (1.1.2) in case the caregiver has no wireless capability. This task (1.1.2) required dealing with two subtasks. The first subtask (1.1.2.1) was communication with the server which was implemented using the Arduino microcontroller to send (1.1.2.1.1) and receive data (1.1.2.1.2). The second subtask (1.1.2.2), was security. In this subtask (1.1.2.2), the data can be encrypted in order to follow the HIPAA (Health Insurance Portability and Accountability Act) standard and provide security for patient information. The last task is the power supply (1.2). The device is powered by an internal power supply (1.2.1) pulling power from a wall outlet (1.2.2) with a battery backup which prevents loss of data (1.2.3) The cost of this feature is around $176. This feature took around 19 hours to

complete. Mahsa worked on the first task (1.1) of this feature, and Daniel worked on the second task (1.2).

Remote Access: doctors and parents can access the data by website (3.1.1) which will appear in a web browser on either a mobile device or a PC (3.1.2). To allow for communication with the website and the web browser (3.1.2), the device will do the following tasks: read data (3.1.2.1) and write data (3.1.2.2). There were two activities for the reading task (3.1.2.1), both reading data that could be sent by interface (3.1.2.1.1) or data that could be sent from the device (3.1.2.1.2). For the writing task (3.1.2.2), the I.Smart Monitor can print collected data to a website page (3.1.2.2.1) that allows both doctors and parents to access the real-time readings. This feature took around 90 hours. Mahsa designed the website (3.1.1) and Vasiliy worked on the web connection (3.2.1)

In addition to these features' tasks, there are administrative tasks that have to be done. In general, all group members will participate in doing these tasks.

Project Management includes the following management tasks:

Design idea (6.1) which includes the design idea report (8.1) and presentation (7.1). All team members worked to implement this task.

Breadboard Proof (6.2): To implement this task, all team members participated. The CpE students worked on programming the microcontroller and the EEE students created the sensor circuit. All prepared the Breadboard Proof Presentation (7.3), and the task manager was Vasiliy.

Work Breakdown Structure (6.3): involved organizing the breakdown of all of the needed tasks that had to be done, creating the WBS diagram and writing a report to describe every task. Mahsa did the WBS diagrams, and Vasiliy and Duaa collaborated to write the WBS report.

Project Timeline (6.4): The team used Microsoft Project to create the project timeline for the whole project, and this was implemented by Duaa. Also, a separate timeline was created every two weeks.

Create Laboratory Prototype (6.5): All group members worked on getting the laboratory Prototype done, and the task manager was Daniel.

Working Laboratory Prototype (6.6): this was the result of our work of the entire fall semester. It consists of a Presentation (7.5) and documentation (8.3). The task manager was Joseph.

Team leaders (6.13-6.17): Team 6 has five members and each one led the group for a period of time, as determined in the project timeline. The leader has a big responsibility. He/she assigns tasks, make sure the assigned tasks are done, reviews documentation and lead the group meetings. Our first team leader was Joseph form the beginning of the fall semester until end of October. The next leader was Mahsa from the beginning of November until December.

Team Leader Report (6.18-6.22): Each leader wrote a report discussing his/her period leading the group, evaluating each team member, discussing difficulties and problems that need to be addressed, and giving advice and recommendations to the next team leader.

Team weekly reports (6.23): There was a weekly report every week that describes the last week's tasks and the next week's expected tasks. Each member documented his/her tasks with hours and current status. The leader discussed the overall team work and the status of the project.

Team member Evaluations (6.24-6.26): Each member wrote an evaluation on each other member based on their performance in the group.

Promote Speaking Proficiency: This task included all the group presentations during

the year. Each task needed 4 hours to be implemented.

Problem Statement Presentation (7.1): This was a presentation in which the team presents the Problem Statement and Elevator Pitch. All members prepared and participated in this presentation.

Feature List Presentation (7.2): The team presented the features of the project. Each member participated and briefly discussed a feature.

Breadboard Proof Presentation (7.3): This was a demonstration of the viability of the project. The team demonstrated the major elements of the I.Smart Monitor design.

Mid-Term Technical Review Presentation (7.4): The purpose was to demonstrate the integrated components of our design idea with real hardware and software.

Laboratory Prototype Presentation (7.5): The team presented the I.Smart Monitor project and team member tasks throughout the semester to the audience, and discussed what tasks have to be done in the spring semester to complete this project.

Promote Writing Proficiency: Each report needed approximately 6 hours to be done.

Problem Statement Report (8.1): The team wrote a report to define the societal problem that our design will address.

Work Breakdown Structure Report (8.2): In this report, all the tasks that have to be done were broken down into subtasks and a WBS diagram was created and a report written to describe every task. Mahsa did the WBS diagrams, while Vasiliy and Duaa wrote the report.

Design Idea Contract Report (8.3): This report stated our design idea and explained the feature set of our design.

End of Term Documentation (8.4): This was the last report of the fall semester; it consisted of the documentation of the working laboratory prototype.

The total cost for our design project in the fall semester was around $538.70, and the total time was 988 hours.

Figure V.2a—Work Breakdown Structure with Hours

Project Management		
6	Deisgn Idea	6.1
	Bread Board Proof	6.2
	Work Breakdown Structure	6.3
	Project Timeline	6.4
	Create Laboratory Prototype	6.5
	Working Laboratory Prototype	6.6
	Revise Timeline	6.7
	Device Testing	6.8
	Market Review	6.9
	Modify Prototype	6.10
	Mid-Term Technical Review	6.11
	Deployable Prototype Review	6.12
	Team Leader 1	6.13
	Team Leader 2	6.14
	Team Leader 3	6.15
	Team Leader 4	6.16
	Team leader 5	6.17
	Team Leader 1 Report	6.18
	Team Leader 2 Report	6.19
	Team Leader 3Report	6.20
	Team Leader 4 Report	6.21
	Team Leader 5 Report	6.22
	Team Weekly Reports	6.23
	Team Member Eval 1	6.24
	Team Member Eval 2	6.25
	Team Member Eval 3	6.26
	Team Member Eval 4	6.27
	Deployable Prototype Complete	6.28
Promote Speaking Proficiency		
7	Problem Statement Presentation	7.1
	Feature List Presentation	7.2
	Bread Board Presentation	7.3
	Mid-Term Technical Review Presentation	7.4
	Laboratory Prototype Presentation	7.5
	Revised Problem Statement Presentation	7.6
	Market Review Presentation	7.7
	Mid-Term Progress Review Presentation	7.8
	Feature Presertation	7.9
	Deploy Prototype Review	7.10
	Final Documentation Report Presentation	7.11
	Deploy Prototype Presentation	7.12
Promote Writing Proficiency		
8	Problem Statement Report	8.1
	Work Breakdown Structure	8.2
	Design Idea Contract	8.3
	End of Term Documentation	8.4
	Revised Problem Statement	8.5
	Device Test Plan Report	8.6
	Market Review	8.7
	Mid-Term Review - Testing Results	8.8
	Feature Report	8.9
	End of Project Documentation	8.10

Figure V.2b—Work Breakdown Structure with Hours Continued

D. Risk Assessment and Needed Mitigations

The largest risk for our project is equipment failure with no backup. Equipment could be damaged by accident (e.g. dropping the device or connecting it incorrectly, etc.) or by failure of another device causing overcurrent, etc. This risk can, however, be nearly eliminated by simply having backups of every part. Of course, this greatly increases the cost, and if that cost cannot be met then the risk cannot be mitigated. This risk, in addition to others, is tabulated below in Table V.2:

Table V.2—Risk Assessment Chart with Mitigation - Fall

Risk	Estimated Likelihood	Estimated Severity	Level of Project Impact Mitigation
Sensor component failure - no backup	10.00%	6	Severe, must then choose new sensor or buy new component. Time consuming and potentially expensive.
Microcontroller/system failure - no backup	5.00%	9	Severe, must order new microcontroller. Definitely expensive and time consuming.
Team decides to change part after time already invested in old part	20%	6	Could seriously halt progress. The benefit of the new part is assumed to be worth the risk and so mitigation is not an issue.
Team member needs to take a personal hiatus	50.00%	3	Not severe. The rest of the team will have to fill in with extra hours.
Code debugging issues	50.00%	6	Potentially severe, depending on the particular code and the time to complete. Very likely to happen to some code.

E. Task Assignment to Complete Each Feature

This section will detail the tasks that each individual member completed, the general group tasks, total hours worked per team member and total hours spent to implement each feature over the fall semester.

1) Fall General Group Tasks for All Members

- Create Problem Statement Report
- Present our Problem Statement
- Create Design Idea Contract Report
- Create a Work Breakdown Structure
- Create a Project Timeline
- Create the end-of-term documentation
- Feature List Presentation
- Breadboard Presentation
- Mid-term technical Review Presentation
- Laboratory Prototype Presentation
- Weekly reports
- Team Evaluations reports

2) Individual Team Member Fall Tasks to Complete Assigned Feature

- Mahsa Shadmani: was assigned to work on the Home Network Connectivity feature and coding for microcontroller to detect connected sensors and interpret data.
- Daniel Schmidt: was assigned to work on the modular sensor feature's circuit

design, construction, coding, and troubleshooting for both sensors.

- Joseph Cacioppo: was assigned to work on the modular sensor feature's circuit design, construction, coding, and troubleshooting for both sensors.
- Duaa Salah: was assigned to work on the Event/Data Logger feature and coding of the microcontroller to detect connected sensors to send and display data.
- Vasiliy Warkentin: was assigned to work on the remote access feature, which included communication with the web server and coding for microcontroller to detect connected sensors and collect data.

3) *Total Hours Spent by Feature*

For the fall semester, 18.5 hours was spent to implement the home network connectivity feature; 97 hours was spent to implement modular sensor feature, 34.5 hours was spent to implement the event/data logger feature; 10 hours was spent to implement the alarm feature; and 90 hours was spent to implement the remote access feature.

4) *Total Hours Spent by Team member*

For the fall semester, Mahsa spent 178 hours, Duaa spent 174.5 hours, Daniel spent 246 hours, Joseph spent 216 hours, and Vasiliy spent 173 hours.

VI. CREATION DETAILS SPRING 2014

The spring semester included a continuation of the fall semester's tasks. Few things changed drastically during the two semesters and this is all outlined below. The most notable change was the reduction in types of sensors from two to one, as mentioned in the design idea section.

A. *Funding*

In this semester, we switched from using an Arduino UNO to an Arduino YUN. So in our final prototype, we didn't use the Wi-Fi and Ethernet shields or the Arduino UNO. Instead we used an Arduino YUN. We had difficulties trying to set up the wireless connection with the first Arduino YUN that we purchased at the end of the fall semester so we bought another one. Also, we purchased an additional micro-SD card.

Table VI.1—Funding Proposal Spring 2014

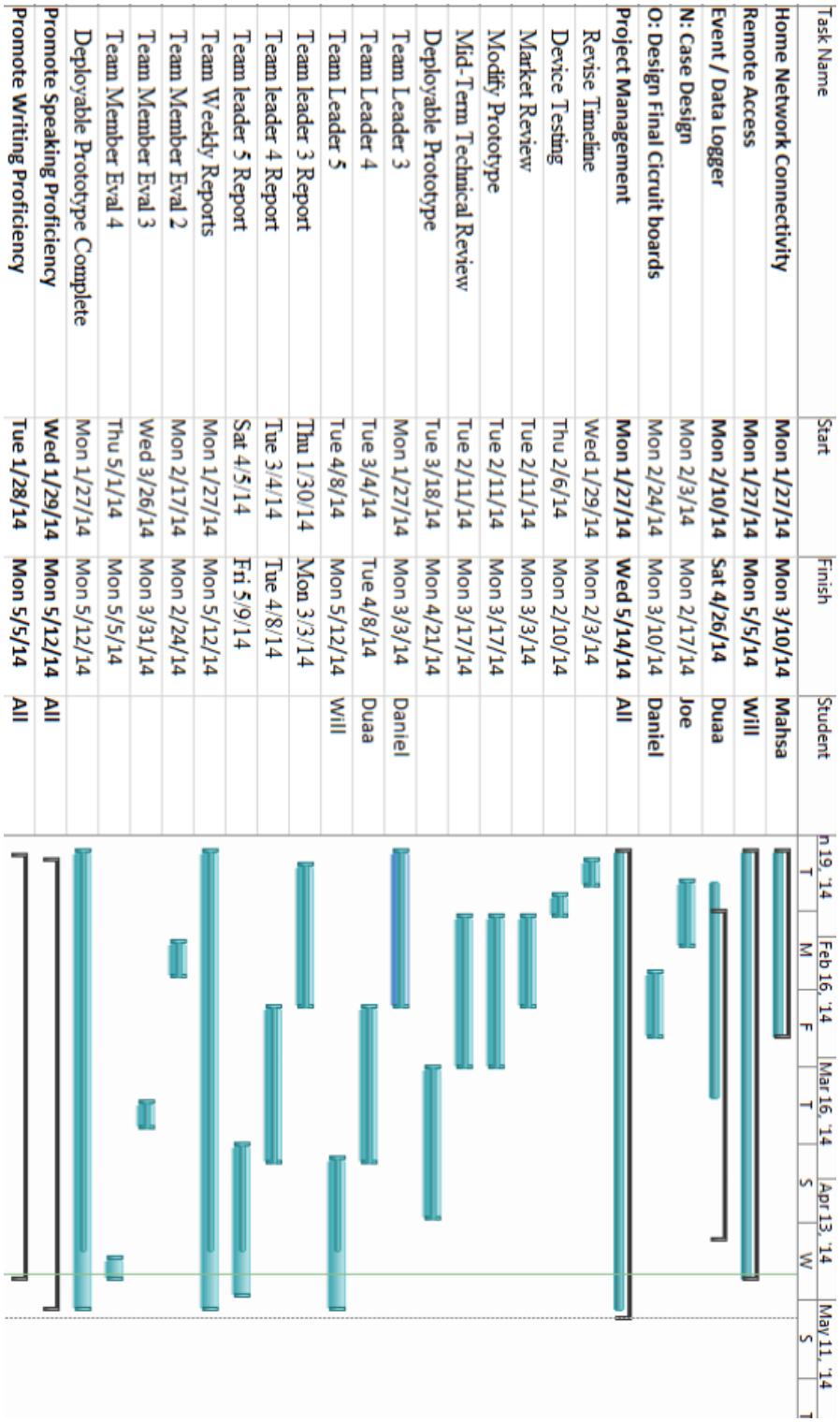
Item	Quantity	Unit Cost	Total Cost
Arduino YUN	1	$85.00	$85.00
microSD Card	1	$19.99	$19.99
Total			$104.99

In general we have used the below items in our final prototype which were purchased during both the fall and spring semesters.

Table VI.2—Funding Proposal for Final Prototype

Item	Quantity	Unit Cost	Total Cost
Wi-Fi Shield for Arduino	1	$88.81	$88.81
Miscellaneous IC's		$23.70	$23.70
4066 chips	1	$2.40	$2.40
MicroSD Card	1	$19.99	$19.99
1 MB EEPROM	1	$6.73	$6.73
Hardware – Chassis, Serial connectors, IC sockets		$48.20	$48.20
RTC module	1	$15.00	$15.00
Arduino UNO + LCD + Ethernet	1	$86.68	$86.68
Arduino YUN	1	$85.00	$85.00
Pulse Sensor	2	$24.95	$47.44
LTE-302 IR Sensor	10	$0.33	$3.31
LTE-302 IR Emitter	10	$0.33	$3.31
Op-amps LM356	10	$0.35	$3.52
FTDI breakout board for ATmega328	2	$14.95	$29.90
16 MHz crystal oscillators	5	$0.95	$4.75
ATmega328	5	$5.50	$27.50
Total			$358.81

B. Project Milestones



Task Name	Start	Finish	Student
Home Network Connectivity	**Mon 1/27/14**	**Mon 3/10/14**	**Mahsa**
Remote Access	**Mon 1/27/14**	**Mon 5/5/14**	**Will**
Event / Data Logger	**Mon 2/10/14**	**Sat 4/26/14**	**Duaa**
N: Case Design	Mon 2/3/14	Mon 2/17/14	**Joe**
O: Design Final Circuit boards	Mon 2/24/14	Mon 3/10/14	**Daniel**
Project Management	**Mon 1/27/14**	**Wed 5/14/14**	**All**
Revise Timeline	Wed 1/29/14	Mon 2/3/14	
Device Testing	Thu 2/6/14	Mon 2/10/14	
Market Review	Tue 2/11/14	Mon 3/3/14	
Modify Prototype	Tue 2/11/14	Mon 3/17/14	
Mid-Term Technical Review	Tue 2/11/14	Mon 3/17/14	
Deployable Prototype	Tue 3/18/14	Mon 4/21/14	
Team Leader 3	Mon 1/27/14	Mon 3/3/14	Daniel
Team Leader 4	Tue 3/4/14	Tue 4/8/14	Duaa
Team Leader 5	Tue 4/8/14	Mon 5/12/14	Will
Team leader 3 Report	Thu 1/30/14	Mon 3/3/14	
Team leader 4 Report	Tue 3/4/14	Tue 4/8/14	
Team leader 5 Report	Sat 4/5/14	Fri 5/9/14	
Team Weekly Reports	Mon 1/27/14	Mon 5/12/14	
Team Member Eval 2	Mon 2/17/14	Mon 2/24/14	
Team Member Eval 3	Wed 3/26/14	Mon 3/31/14	
Team Member Eval 4	Thu 5/1/14	Mon 5/5/14	
Deployable Prototype Complete	Mon 1/27/14	Mon 5/12/14	
Promote Speaking Proficiency	**Wed 1/29/14**	**Mon 5/12/14**	**All**
Promote Writing Proficiency	**Tue 1/28/14**	**Mon 5/5/14**	**All**

Figure VI.1—Project Milestones Spring

C. Work Breakdown Structure

This section represents the work breakdown structure of the project over the spring semester. It explains how each feature was implemented, the time needed to complete each feature, as well as the scope of work, the budget, and the team member responsible for each part.

Over the spring semester, we continued to work in implementing the features that were not completed by the end of the fall semester.

Alarm feature: to be able to complete this task, Daniel continued working on task (5.1) and Vasiliy was responsible to complete (5.1.1). This feature took around 22 hours to complete.

Event/Data Logger: to be able to complete this feature, there were two main tasks, as described in section V.2, Vasiliy continued to work on the first task (4.1) and Duaa continued to work on task (4.2) To implement this feature, it took around 51 hours to complete.

Home Network Connectivity: Daniel and Mahsa continued working to implement this feature over the spring semester. Mahsa continued working on the first task (1.1), and Daniel worked on the second task (1.2). It took around 12 hours.

Remote Access: to complete this feature, Mahsa designed the website (3.1.1) and Vasiliy worked on the web connection (3.2.1). It took around 32 hours to complete.

In addition to these features' tasks, there are administrative tasks that were done and all group members participated.

Project Management includes the following tasks:

Revise Timeline (6.7): This consisted of comparing the completed work over the fall semester and the remaining work, and revising the start time and finish time for each remaining task. The one who was responsible for this task was Duaa.

Device Testing (6.8): This task involved testing our current device, both hardware and software, to make sure it functions correctly. But before that, we wrote a device testing plan report (8.5). All group members worked on this task and the manager was Vasiliy.

Market Review (6.9): This task included reviewing the current devices on the market and making a market review report (8.6) and presentation (7.7). The manager for this task was Mahsa.

Modify Prototype (6.10): This task started from the beginning of the spring semester and was a project-level task involving all modifications based on device testing and the market review. The leader manager was Daniel.

Mid-term technical review (6.11): it included working to prepare for a presentation (7.8)) and writing a report about it (8.7). All members participated to get the work done and the leader manager was Duaa.

Deployable Prototype Review (6.12): This task consisted of the presentation (7.10) and documentation (8.8) of a completed deployable prototype. All group members participated in the presentation and wrote a one page handout of the feature list that the team presented. The leader manager for this task was Vasiliy.

Team leaders (6.15-6.17): the third leader was Daniel from the beginning of December until the end of March; the fourth leader was Duaa from the beginning of March until the beginning of April, and the last leader was Vasiliy from April until the beginning of May.

Team Leader Report (6.20-6.22): Each leader wrote a report discussing his/her period leading the group.

Team weekly reports (6.23): Just as the last semester, there were weekly reports due every week that described the last week's tasks and the next week's expected tasks. Each member documented his/her tasks with hours and status. The leader discussed the overall team work and the status of the project.

Team member Evaluations (6.25-6.26): Each member wrote two team evaluations during this semester.

Deployable Prototype Complete (6.27): This was the last task, which was a demonstration of a working deployable prototype. It included

preparing for a presentation, writing a one page report about the project's features, and demonstrating the project to an audience. The leader of this was Vasiliy.

Promote Speaking Proficiency: This task includes all the group presentations during the year. Each task needed 6 hours to be implemented.

Revised Problem Statement Presentation (7.6): This was the first presentation of the spring semester. It was intended to give a short overview of the revised problem statement/design idea, and the project timeline, based off the experiences from the fall semester.

Market Preview Presentation (7.7): the team gave a short overview of the market review to the entire senior design group.

Mid-Term Progress Review Presentation (7.8): to discuss and demonstrate the project after device testing and alteration.

Feature Presentation (7.9): Each member presented and discussed the feature that was assigned to them.

Deployable Prototype Review (7.10): We demonstrated the completed deployable prototype and presented the important features of the project.

Final Documentation Report Presentation (7.11): In this presentation, the team discussed and presented the I.Smart Monitor project documentation to the instructor.

Deployable Prototype Presentation (7.12): This was the last presentation, which was a demonstration of a working deployable prototype. It included preparing for the presentation, writing a one page handout about the project's features to give to all the visitors, and demonstrate the project to said audience.

Promote Writing Proficiency: Each report needed approximately 6 hours to complete.

Revised Problem Statement (8.5): We wrote a report about our review of the problem statement and design idea contract.

Device Test Plan Report (8.6): This reported our test plan of our device.

Market Review Report (8.7): We reported our market review after talking to experts and business managers.

Mid-Term Review-Testing Results (8.8): We documented the device testing results and how the test results impacted the project.

Feature Report (8.9): Each member wrote a report about their assigned feature.

End of Project Documentation (8.10): We documented all aspects of the project by providing all the required documentations.

The total time spent to complete implementing the feature set was 750 hours. Figures V.2a and V.2b both show the WBS from the fall semester and it still applied the same in the spring semester.

D. Risk Assessment and Needed Mitigations

The spring semester consisted of far fewer actual design elements than the fall semester. Because of this, most risks revolved around building and deadlines were less threatening. For example, all prototype components were soldered to perforated circuit boards. As the process of mapping a circuit from a protoboard to a perf-board is fraught with opportunities for error, it represents a risk. In addition, all team members are currently in their last semesters of college and so senioritis was a constant companion.

Table VI.3—Risk Assessment Chart with Mitigation - Spring

Risk	Estimated Likelihood	Estimated Severity	Level of Project Impact and Mitigation
Component failure - no backup	10.00%	6	Severe, must then choose new sensor or buy new component. Time consuming and potentially expensive.
System failure - no backup	5.00%	9	Severe, must order new microcontroller. Definitely expensive and time consuming.
Error discovered in soldered circuit	30%	6	Could seriously halt progress. The soldering must be done carefully and mindfully.
Team member needs to take a personal hiatus	50.00%	3	Not severe. The rest of the team will have to fill in with extra hours.
Code debugging issues	80.00%	7	Potentially severe, depending on the particular code and the time to complete. Very likely to happen to some code.

E. Task Assignment to Complete Each Feature

This section details the tasks that each individual member completed, the general group tasks, total hours worked per team member and total hours spent to implement each feature over the spring semester.

1) Spring General Group Tasks for All Members

- Create Revised Problem Statement Report and Presentation
- Create Device Test Plan
- Create Market Review Report and Presentation
- Create the End-Of- Project Documentation
- Mid-Term Progress Review Presentation
- Create Feature Report and Presentation
- Deployable Prototype Review
- Deployable Prototype Presentation
- Weekly reports
- Team Evaluations report.

2) Individual Team Member Spring Tasks

- Mahsa Shadmani: was assigned to continue working on Home Network Connectivity feature, design the website, and test the feature.
- Daniel Schmidt: was assigned to work on the power supply task, Alarm feature,

design Printed circuit board, design the case and testing

- Joseph Cacioppo: was assigned to design the case and to test the Modular Sensor feature.
- Duaa Salah: was assigned to complete the Onboard Storage feature and testing of the feature.
- Vasiliy Warkentin: was assigned to continue working on the Remote Access feature, code for the alarm and web communication.

3) Total Hours spent by Feature

For the spring semester, 11.5 hours were spent to implement the Home Network Connectivity feature; 97.1 hours were spent to implement the Modular Sensor feature; 51 hours were spent to implement the Event/Data Logger feature; 22 hours were spent to implement the Alarm feature; and 32 hours were spent to implement the Remote Access feature.

4) Total Hours Spent by Team member

For the spring semester, Mahsa spent 161.5 hours, Duaa spent 188.5 hours, Daniel spent 164.5 hours, Joseph spent 125.5 hours, and Vasiliy spent 112 hours.

F. Market Review

Currently, there are several infant health monitors available to parents and healthcare providers. However, none of our competitors' devices offer smart sensors, which analyze and process the vital sign data for use by parents and pediatricians. These unique sensors consist of two components: the non-invasive sensor that attaches to the infant and a sensor controller that provides the processing functionality. These external sensors and sensor controllers can be developed independently of the I.Smart hub. This idea is unique to the I.Smart Sensor design—creating new accessories, without upgrading the hub. The burden on the hub is also independent of the amount of sensors connected to it. The hub itself has no additional tasks when an additional sensor is connected. Simply put, it is a Multi-Slave, Single-Master design that allows the I.Smart Monitor to be unique. Because of the I.Smart's modular design, parents can choose to monitor one or more different vital signs at a time. This is a marketable improvement over our competition; the other devices have specific monitoring capabilities that cannot incorporate additional functions as new technologies are developed.

Our unique contribution to the baby monitoring market is the integration of smart sensors, which actually process and interpret data, and the secure website, where the data is available remotely. The I.Smart Monitor takes data from the smart sensor controllers and sends the data, by a specific protocol, to the central hub where it is routed appropriately. This system of data acquisition, storage and transmission is our innovation.

The I.Smart Monitor system is also unique by providing the pediatrician with remote access to near real-time and historical data via the internet. This secure server provides data in two forms: an easy to understand format for parents and a comprehensive format for their pediatricians. In addition, it conforms to all current laws under the HIPAA.

The I.Smart Monitor team will apply for a series of patents for smart sensor technology. The first patent is a utility patent, which incorporates the way the sensor controllers communicate with the central hub by transferring data, including the algorithm. The second patent is a design patent, which patents the look of the product design.

1) Cost

A fair monetary estimate to fund the completion of the prototype, including the safety testing, development of the website and verification of the remote alarm system, approximately $150,000, which does not include any regulatory fees. The development of the hub and smart sensor controller technology is complete; the remaining steps use existing technology, final product design, and software.

2) Initial Market & Total Market Value

The initial customer group for this product is parents of premature newborns and other high-risk infants, with an emphasis on educating their pediatricians on the benefits of the I.Smart monitor. In the U.S. alone, approximately 4 million babies are born each year, making the market sustainable.[6]

The amount of money parents spend in the first year on products and services for the health and safety of their infant ranges from $8 to $14 thousand. Parents spend $200 million on infants in the Sacramento area alone.[7] The I.Smart device is an attractive option to our competitors', most of which monitor only one or two vital signs. While other companies send data to a phone app, the I.Smart monitor system is superior since it allows the data to be viewed by both parents and pediatricians via website and the smart sensor technology allows scalability of the device to user's current and future needs. To provide this coverage, the I.Smart monitor would sell for around $250, which would include the hub, 2 smart sensors, and lifetime access to the I.Smart website.

The initial test market is premature babies in Sacramento County. Sacramento County from 2002 to 2011 had an average of 20,762 births per year.[8] Premature infants account for approximately 2554 of those births. If each parent with a premature infant spends roughly $150 for an infant monitor (the average price of our closest competitor), this Sacramento test market would be worth approximately $383,100. If the I.Smart Monitor were to capture 5 percent of our test market, we would generate an average of $19,155 in sales revenue per year during the test market phase.

Once the initial test is complete, and the product seems viable, the intermediate market for the I.Smart monitor would be to sell it nationwide to parents of premature infants. Assuming that 500,000 babies are born prematurely each year, and each baby needs an infant monitor, we forecast that the overall premature infant monitor market is worth $75 million. A 5 percent market share for the I.Smart Monitor in the national premature infant market is worth $3.75 million.

The final market for this device will be a nationwide market to include all babies born in the United States, approximately 4 million births[9] and $600 million in baby monitor sales revenue per year. A 3 percent market share for the I.Smart Monitor in the national baby monitor market is worth $18 million in sales revenue. With the 4 million births per year in the U.S., the market potential for baby monitors is easily adequate to support the I.Smart Monitor business. The cost of the initial prototype is $117.21 per unit; however, with economies of scale and labor cost considerations, the projected cost of manufacturing per unit is $87.90. The wholesale price of the I.Smart Monitor will be $180 and the market price for the monitor will be $250.00; this will allow for a gross margin of around 40 percent. At the 3 percent market share, the estimated gross profit will be just over $9.18 million.

3) Window of Opportunity

Our product relies on parents and pediatricians choosing our product over those of our competitors. We believe that our product is affordable and offers benefits that are superior to other baby monitors. Our product is adaptable

to changing technology so will have a longer product life cycle than those of our competitors. With new monitors soon to reach the market, such as the Owlet Sock Monitor and the Baby Fairy Wrist Monitor, the market trend is for continued technological improvements in baby monitors. Even with new products entering the market, the I.Smart Monitor is still more advanced and comprehensive than the competition. We want to take advantage of our innovative product by introducing the I.Smart Monitor before other companies can develop similar technology.

4) Barriers to Entry

The United States Food and Drug Administration (FDA) requires an inspection of each new medical device and the completion of a rigorous application process. FDA Section 201 H states that a baby monitor is only a medical device when it claims to cure or prevent SIDS. Since we are not claiming to cure or prevent SIDS, our baby monitor is not a medical device; therefore, we are not subject to this rigorous application process.

Some other barriers to entry into this market are the initial costs of market penetration and the lack of product recognition. Other manufacturers have established their brands as safe, reliable and trustworthy. Our new product will not initially have those qualities; however, the superiority of our product will allow us to compete in this market.

5) Competitive Advantage

The I.Smart Monitor system is the only infant monitoring system that uses smart sensor controllers, which analyze and process vital sign data for use by parents and pediatricians. By using smart sensor controllers, our product is adaptable and upgradeable to new technologies while our competitors' products are not. The smart sensor controllers are "plug and play" and are easy to use for our customers. Our overall product package includes the user-friendly website that provides near real-time and historical data to parents and pediatricians and facilitates improved parent/pediatrician communication.

Any product released into a market must be purposefully designed to fit into that market in some way. In this paper, it has been shown what that market is for the I.Smart Monitor. As discussed above, data was gathered from the parents of infants to determine the best way to enter that market and the features of the I.Smart Monitor were designed with this in mind. Without taking into consideration how this product will actually get into the hands of people whom it would benefit, their hands will likely remain empty.

Once a customer has acquired this device, they, or a technician tasked with maintenance, will need a comprehensive guide to the using it and a summary of how it works. The following four sections function as this guide.

VII. USER MANUAL

Attention! This device should not be operated in extreme temperature. This device is for indoor use only. For best connection to the internet, it is recommended that the Hub is placed within 50ft of your home Wi-Fi router.

The I.Smart Monitor system consists of two major components. The first one is the Hub, this is the centralized device that processes all the data from different sensors, and then stores the data. The second is the Sensor Controller; this is the device that the sensors are connected to. The Sensor Controller communicates the data from the sensors on the baby to the Hub. The Hub communicates to the Sensor Controllers via the communication port, COM for short. The Sensor Controllers can be daisy chained together in any order, and only one Sensor Controller has to be connected to the Hub for

the Hub to see all the Sensor Controllers in the chain.

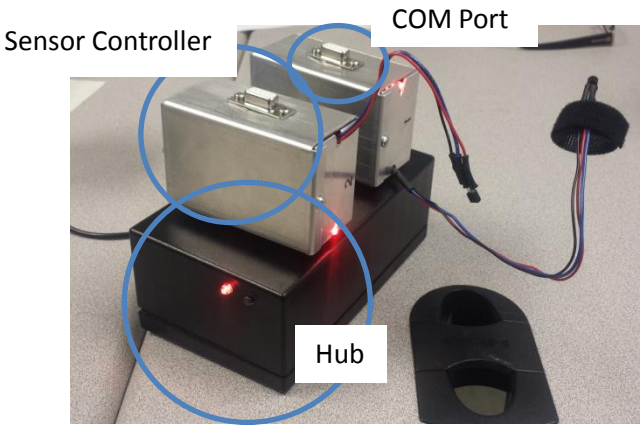


Figure VII.1—Different Parts of the I.Smart Monitor

A. *Initial Setup*

1) Place the Hub on an even, hard surface next to your baby's crib and make sure that it is within three (3) feet of a power outlet.
2) Insert a microSD card with at least 4GB of space into the microSD slot on the Hub.
3) Place the desired sensor controller next to the Hub and attach the male end COM port of the Hub to the Female end COM port on the sensor controller.
4) If you have more than one sensor, attach the sensor's COM port to the previously attached sensor.
5) Refer to the sensor's user manual on how to attach the sensor to the baby.
6) After all the sensors are connected to the baby, insure that none of the cables will cause entanglement.
7) Attach the Hub to the power outlet with the provided micro-USB cable and USB adapter.
8) The Hub will then start collecting data from the sensors controllers.

B. *Connecting to the Home Network*

The I.Smart Monitor has the ability to act as an Access Point, but it can also connect to an existing network. These instructions walk you through connecting your I.Smart Monitor to a wireless network. The I.Smart Monitor can connect to unencrypted networks, as well as networks that support WEP, WPA, and WPA2 encryption.

1) When you first power on the I.Smart Monitor, it will create a Wi-Fi network called *Arduino Yun-XXXXXXXXXXXX*. Connect your computer to this network.
2) Once you've obtained an IP address, open a web browser, and enter http://arduino.local or 192.168.240.1 in the address bar. After a few moments, a web page will appear asking for a password. Enter "Arduino" and click the *Log In* button.

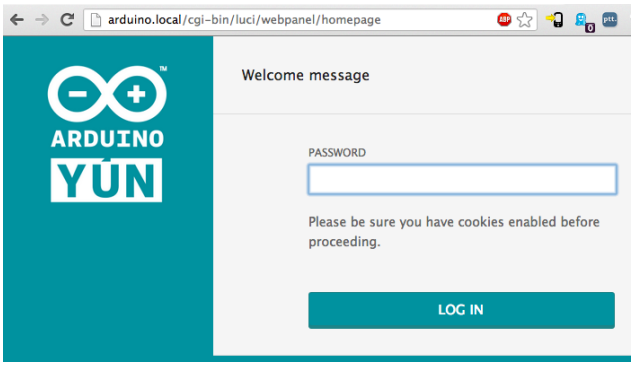


Figure VII.2—Login page for Arduino Yun

3) You will find a page with some diagnostic information about the current network connections. The first is your Wi-Fi interface; the second is your Ethernet connection. Press the *Configuration* button to proceed.

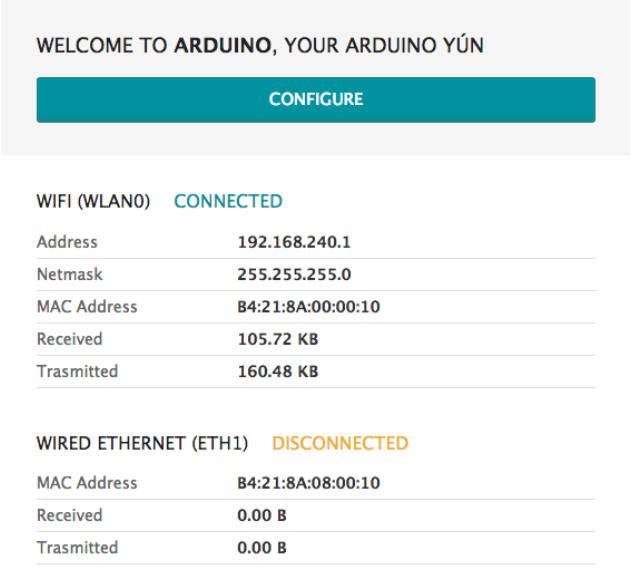


Figure VII.3—Example IP Configuration of the Arduino Yun

4) On the new page, you will configure your I.Smart Monitor, giving it a unique name and identifying what network you want to connect to.
5) In the *I.Smart Monitor NAME* field, give your Arduino a unique name and record it somewhere secure. You'll use this to refer to it in the future.
6) Choose a password of 8 or more characters for your Arduino. If you leave this field blank, the system retains the default password of Arduino
7) If you wish, you can set the time zone and country. It is recommended to set these options as it may help connecting to local Wi-Fi networks. Setting the local time zone also selects the country's regulatory domain.
8) Enter the name of the Wi-Fi network you wish to connect to.
9) Select the security type, and enter the password.

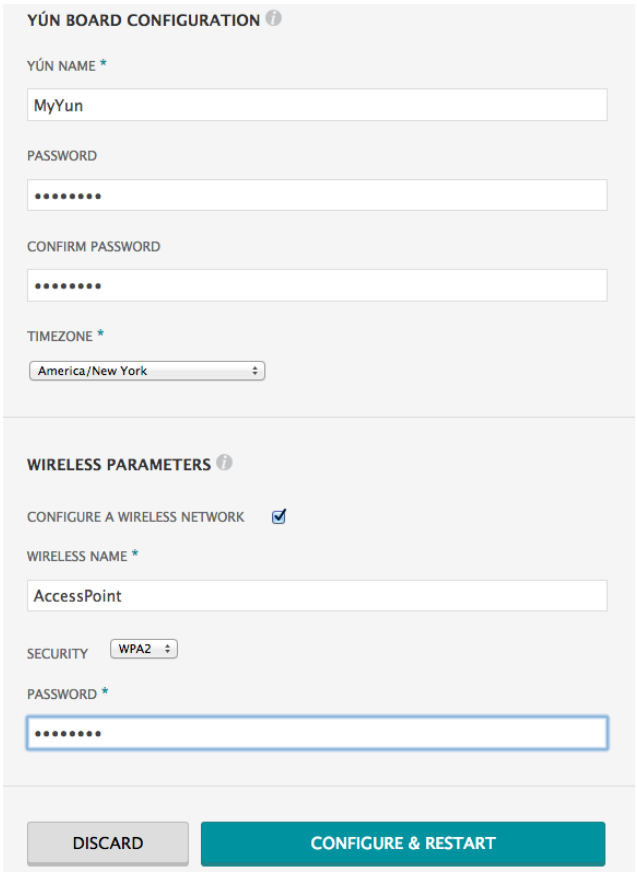


Figure VII.4—Home Network Parameters

10) When you press the *Configure & Restart* button, the Arduino will reset itself and join the specified network. The Arduino network will shut down after a few moments.

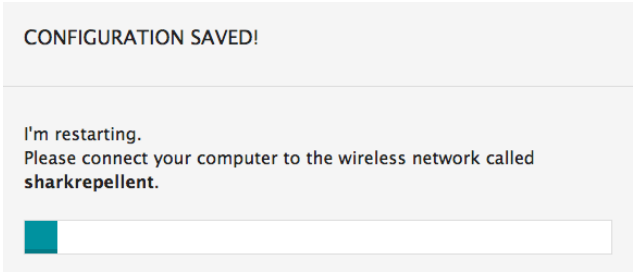


Figure VII.5—Arduino Yun Configuration Loading Screen

11) You can now join the network you assigned to the I.Smart Monitor.

C. Viewing the data:

1) Disconnect the Hub from power
2) Remove the microSD card from the Hub
3) Use a microSD reader to attach the microSD card to your computer
4) Open the microSD card in a file browser
5) Find the serial number of the sensor you want to view and open that file.

VIII. HARDWARE

A. Block Diagram & Documentation at Block Level

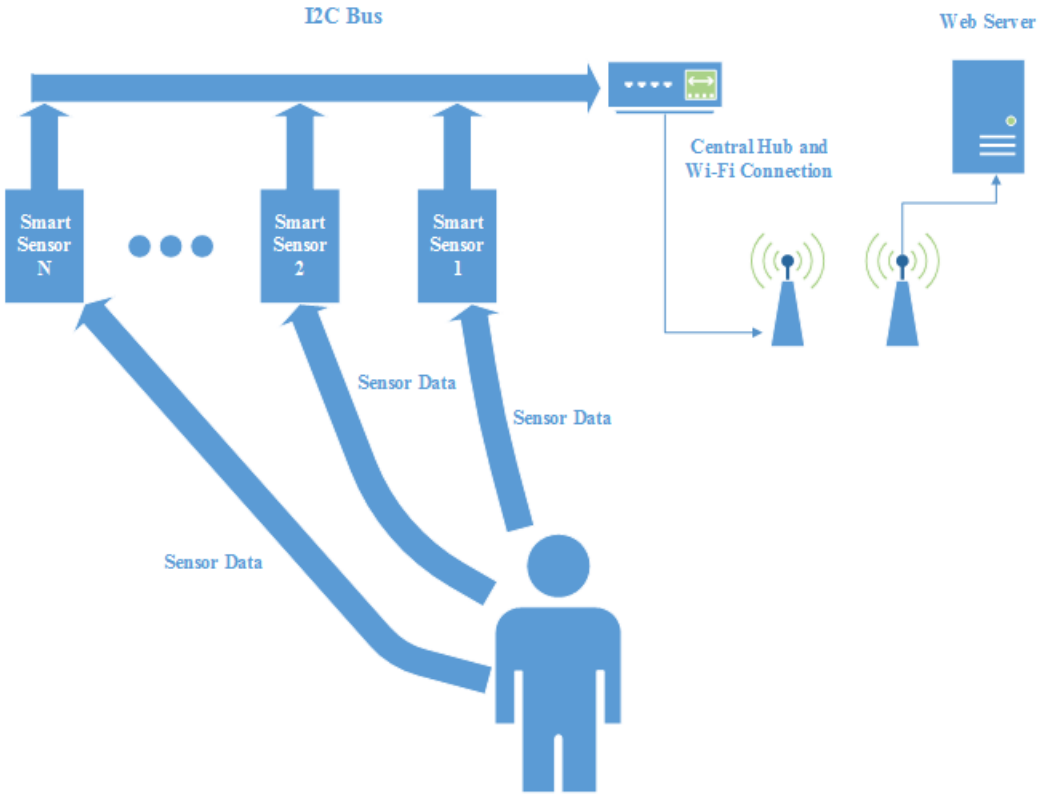


Figure VIII.1—Hardware Block Diagram of I.Smart Monitor

B. Schematic & Documentation to Component Level

In figure VIII.2, the sensor controller circuit can be seen. The 10 uF capacitor allows the ATmega328 chip to start up properly when power is first applied.

Since the device as a whole consists primarily of microcontroller platforms and integrated circuits, there is only one feature that was implemented using discrete components, the alarm.

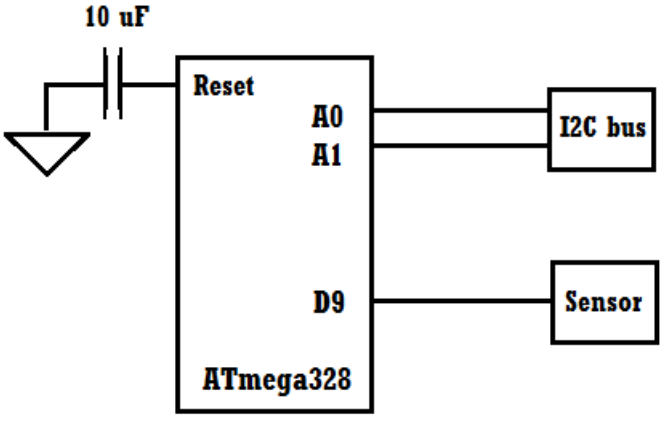


Figure VIII.2—Sensor Controller Circuit

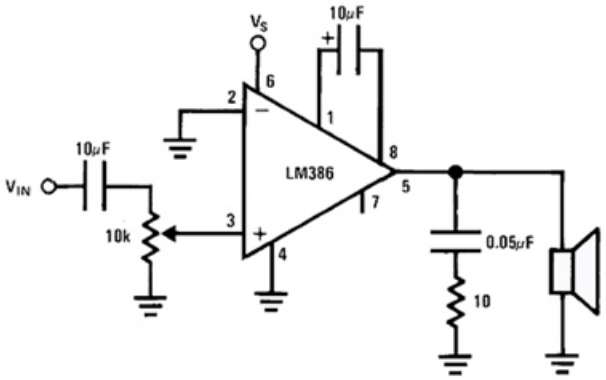


Figure VIII.3—Alarm Circuit Schematic

IX. SOFTWARE

A. Block Diagram & Documentation at Block Level

The main device had a simple task of detecting, retrieving information, and processing the information from the sensors that are connected to it. Then, it determines where to write the processed data.

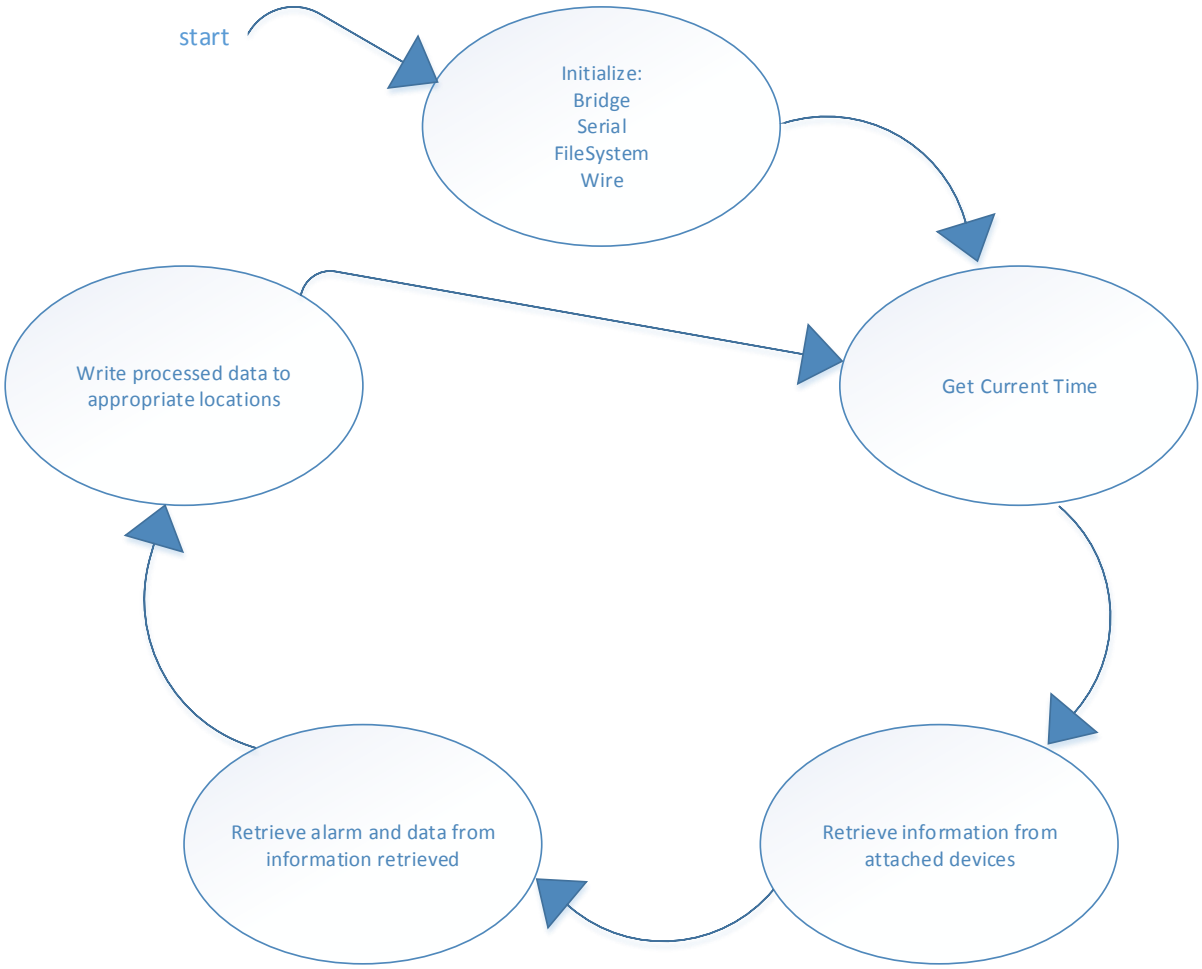


Figure IX.1—Flowchart of the Main Device Process.

B. *Flowchart, Pseudo-Code, & Documentation to Subroutine Level*

In this section the flowcharts show the basic processes of how the Hub device works. For the complete code for the Hub device, see Appendix B.

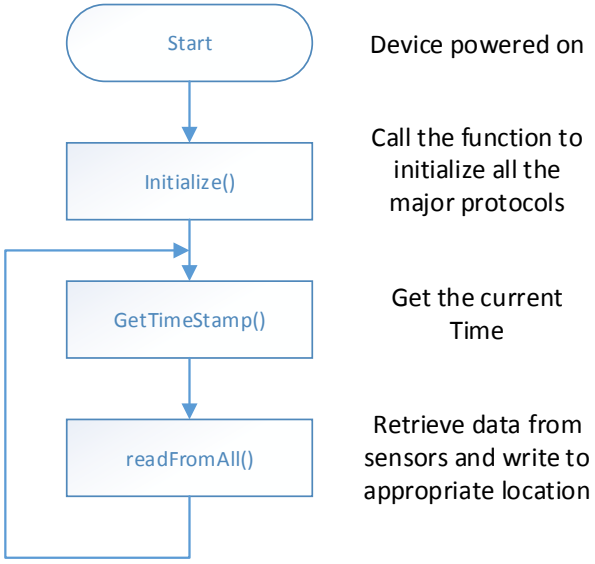


Figure IX.2—Flowchart for Main Hub Code

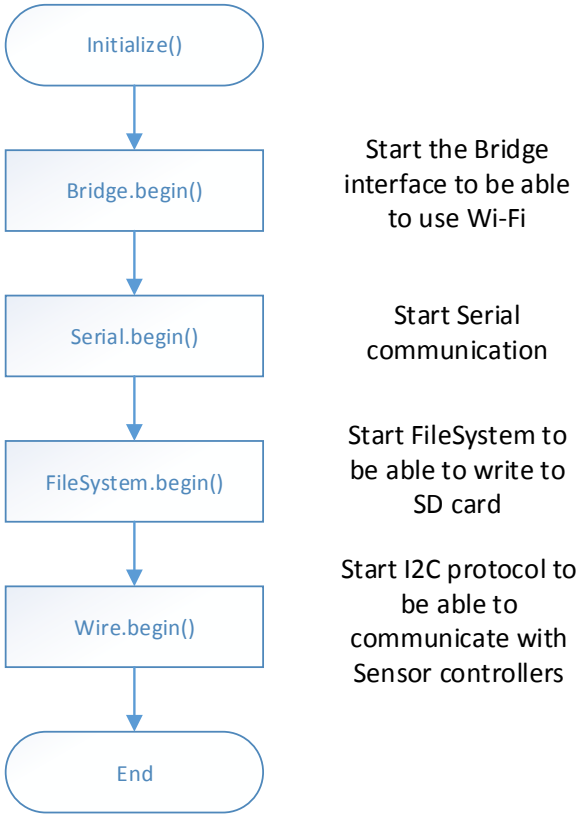


Figure IX.3—Flowchart of the Initialization Code

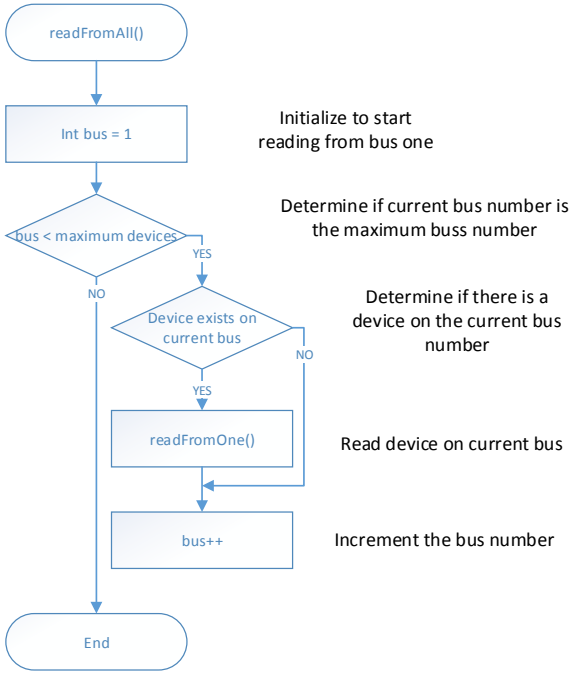


Figure IX.4—Flowchart of Sensor Data Collection – Hub Side

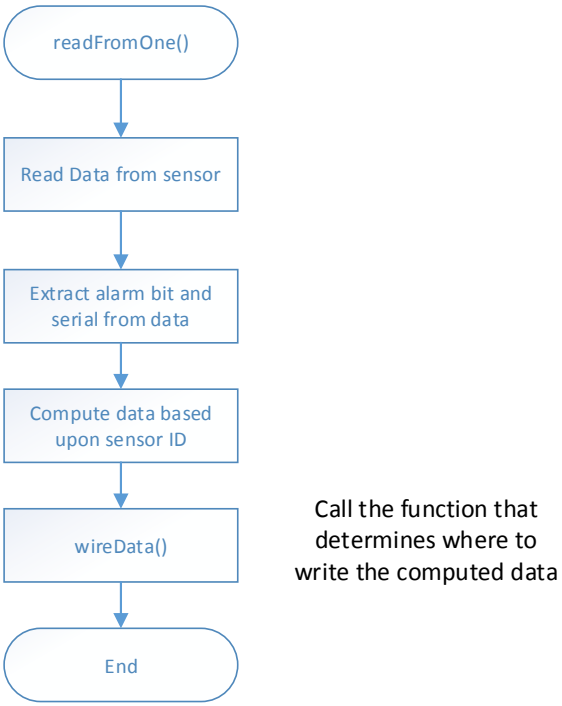


Figure IX.5—Flowchart of Sensor Data Collection – One Sensor

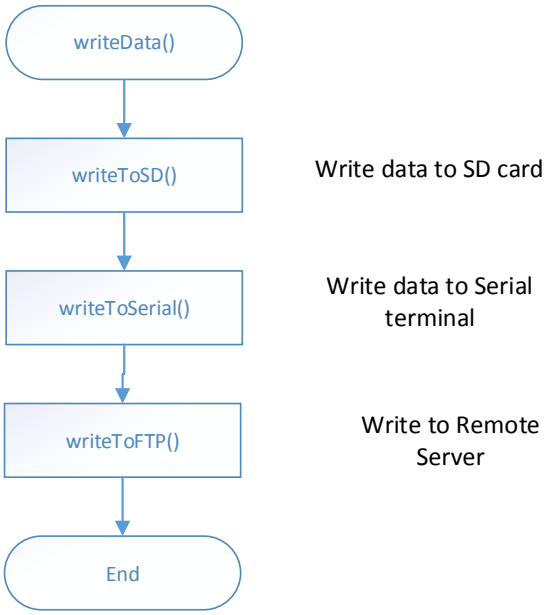


Figure IX.6—Flowchart for Sensor Data Writing

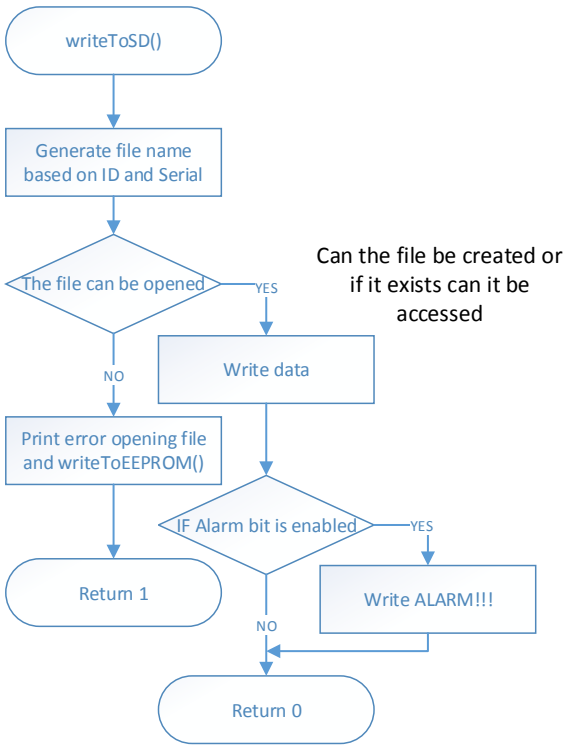


Figure IX.7—Flowchart of SD Card Data Writing

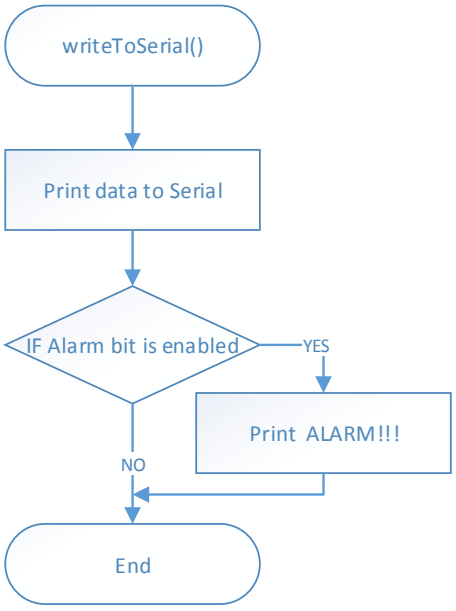


Figure IX.8—Flowchart of Sensor Data Serial Writing

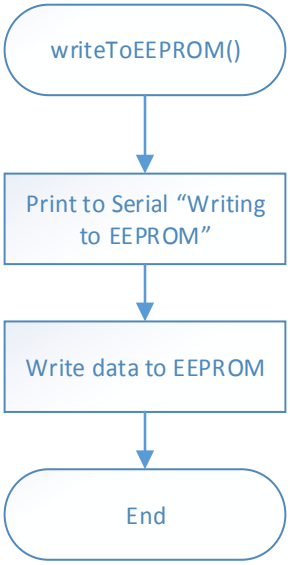


Figure IX.9—Flowchart for Writing Sensor Data to EEPROM

X. MECHANICAL: DRAWING AND DOCUMENTATION

For the prototype chassis, a proprietary electronic housing was selected. A perforated circuit board was likewise chosen. The device was placed into this housing at the end of the first semester and soldering began at the beginning of the second. Both can be seen below in figure X.1.

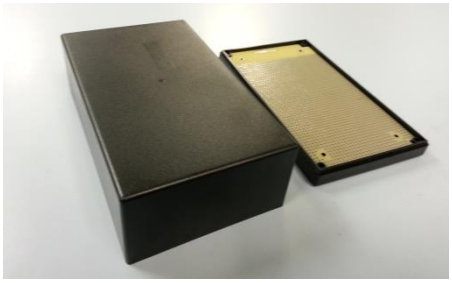


Figure X.1: Proprietary Housing Chosen for Main Hub

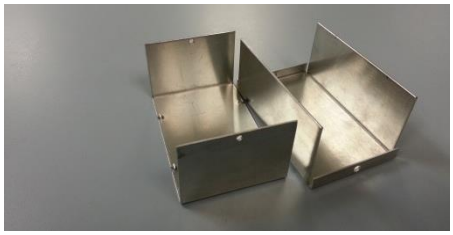


Figure X.2: Housing Chosen for Sensor Controllers

To allow the sensors to be easily connected to the main hub and each other, some RS-232 serial connectors and housings were purchased and wires soldered to the chosen terminals, as can be seen below in figure X.3.

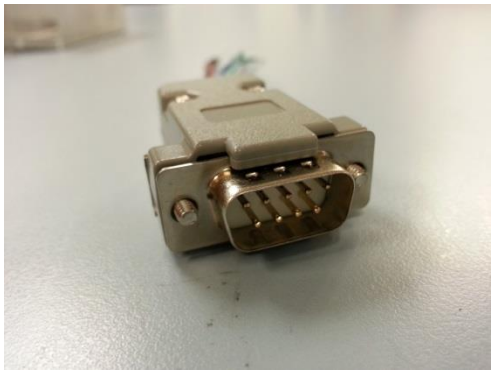


Figure X.3: RS-232 Serial Connectors for Sensor Controllers

The chosen pinout is pictured below in figure X.4.

```
male
Vcc       SDA GND
 0   0   0   0   0
   0   0   0   0
  SCL TX RX

female
GND SDA      Vcc
 0   0   0   0   0
   0   0   0   0
     RX TX SCL
```

Figure X.4—Pin-out for Sensor Controller Serial Connections

XI. TEST PLAN AND RESULTS

The elements of this test plan fall under two categories, safety and functionality. Safety testing is obviously important for the health of the user, but is also something that investors would be concerned with as a means of reducing future liabilities. Functionality testing is important for the reputation of the designer and the company producing the device, and in the case of a medical device, can be closely linked to safety testing. Below, a plan is detailed for testing the I.Smart Monitor in both fashions. The plan begins at the level of individual microcontrollers and extends to the device as a whole. First the hardware test plan is discussed, followed by software.

A. Hardware

1) Electrical Properties

The electrical isolation of the main hub and sensor controllers was tested by measuring the resistance to circuit ground at several key nodes. These included the connection point between the sensors and sensor controller, the connection point between the sensor controllers and I^2C bus, and any other points where the resistance was expected to be high, such as on the 'north' (positive supply) side of any active devices. The resistance to ground at each of these points was over a meg-ohm.

In addition to electrical isolation, the voltage and current demands of the device must be determined for proper documentation to be possible. The changes in these values as more sensors are added or taken away are also important. This measurement was accomplished by connecting an ammeter in series with the main supply, and tabulating the current values over time as different numbers of sensors are attached. This could then be graphed or presented in some visual way for the data sheet.

Table XI.1—Electrical Properties of Hardware

Component	Current	Power
Alarm	5 mA inactive, 25 mA active	40 mW inactive, 200 mW active
Sensor	70 mA	420 mW
Main hub	100 mA	600 mW

The specifications of an I^2C bus contain several measureable quantities, such as maximum bus capacitance. This could be measured and recorded with a digital capacitance meter as several sensors are connected and disconnected. These data would then be compared and any relationship graphed.

2) Electromagnetic Properties

Testing the EM properties of the device would be done using an inductive antenna and an oscilloscope. A spectrum analyzer would be more effective however there is not one at the disposal of the group. As the device is running enclosed within its case, the probe, perhaps with a small coil attached to the end, could be moved around the outside and the oscilloscope checked for any signals. Conversely, a function generator could be used to attempt to send signals into the device while checking to see if they are being received anywhere inappropriately.

3) Microcontroller Testing

The testing of the sensor controllers involves both testing the correct transmission and reception of data from the sensors, but also testing the electrical properties of the circuit to ensure that it is properly isolated from the power source. The method of testing the former is to transmit some known constant value and check

if that value was received. The sensors themselves were tested in this way to determine if they required calibration. To test the controllers themselves, test code was written into another microcontroller that output a constant or functional binary value. This microcontroller was then connected to the sensor controller as if it were a sensor. The received value was then output to a serial monitor and compared to the expected value. In nearly all tests the values matched precisely over the entire range of temperatures that the sensor would be exposed to in this application. The exception was an intermittent problem that occurred only once.

4) Temperature

As the main hub and the sensor controllers are enclosed within a small package, it is important that none of the parts generate significant heat. If heat is generated regularly, than steps must be taken to dissipate that heat, which will inevitably increase the size of the affected part. Temperature can be easily measured using an infrared thermometer, such as those in many DMMs, or kitchen-supply stores. The device was left running for several hours with its temperature taken periodically at several key locations and the results recorded. Key locations included all power-control circuitry and any point where one conductor becomes two, etc. No part showed significant change in temperature with two sensors connected simultaneously. This is however to be expected since the device is already known to use little power during normal operation.

To verify the reliability of our temperature sensors, a device such as an Isotech Dry Block Calibrator would be used. This device is a metal block with pockets that can be set to a desired temperature, and into which a temp sensor can be inserted. Comparing the Isotech setting versus the output of the I.Smart temp sensor, will verify the integrity of the I.Smart Monitor.

5) Alarm

The main issue with the audio circuit was noise, mainly in the form of high-frequency "hiss." This was resolved with filter capacitors, as can be seen in figure VIII.3. Both the signal and power supply required high-pass filtering most likely due to the fact that the "power supply" is a microcontroller.

The second type of testing was focused on whether the circuit was as effective as it needs to be in an environment analogous to its place of operation; namely, a home with possible noise pollution. The main point here was to determine if the alarm could be easily noticed in another room with audio interference. The location chosen was two rooms with a closed door between them and plenty of chatter in the vicinity. From this, it was determined that the sound must be very loud and changing regularly so as to sound very different from ambient din.

The measured electrical characteristics of the alarm circuit can be seen in table XI.1 above.

6) Case & Chassis

The device is mounted onto a frame which is enclosed within a case. These are both designed to protect the electronics and as such have to be reliably sturdy. This could be easily tested through destructive means with a similarly built case. Here, however, both case and frame are proprietary items that have their own pre-determined specifications which could be simply acquired from the manufacturer, rather than wasting money destroying something that was purchased.

7) Reliability

It is important to ensure that the device is, in fact, easy to use and that the average user will have great difficulty in getting it to stop working correctly. This would be tested by giving the device to several non-technical individuals and

allowing them to play with it and try and use it. Their successes and failures would be carefully noted and considered as a possible reason to alter the design.

8) Wireless

The testing of the wireless system was accomplished inside several different environments with Wi-Fi networks. These include a home and various spots in the laboratory. The reception power was measured in different rooms in differing proximity to the router and the results tabulated. The measurement can be done using proprietary software. The floor plan of each location was mapped and compared to the table of values. This is discussed further in section B.2 below

B. Software

The software was written in a modular fashion, with the code for each feature written separately and then integrated. The software for the sensors is mainly concerned with processing the data into a form transmittable on the I^2C bus. This code was first tested with hard-coded values and upon successful reception was tested with actual real-time sensor data. The software for the data storage control system and Wi-Fi connection were tested in a similar manner. Other components were written separately and debugged; data logger, alarm, remote communication, each component was tested separately. Based on the different test results of each components code, revisions were made separately until the component as a whole could work separately with minimal problems. Then, the separate working components were put together into the main code of the hub and debugged to make sure that the component worked as desired with other components without obvious effects on other components. When all the components were put together, then all the different parts were retested to make sure everything as a whole was working properly.

Afterward, boundary conditions were tested and based on the tests, revisions were either made to the code, or new boundary conditions were observed and new code had to be written to cover those boundary conditions.

1) Event/Data Logger

To ensure that the Event/Data Logger feature worked correctly, testing was done to both the SD card and EEPROM. First, functionality testing was performed on the SD card by writing an Arduino test code that could write and read hard-coded values to and from an SD card simultaneously. Testing results were as expected. It saved the hard-coded values to the SD card, and it read the same values back. After that, the hard-coded values were replaced with actual real-time sensor data by connecting the sensors to the Arduino Yun and running the main Hub code that has a function for the SD card interface. Testing results were as expected. The correct sensor data was saved to the correct file in the SD card, and the saved data was retrievable. Both operational results were matching; the SD card functionality testing was successful.

Also, speed testing was performed to test the data transfer rate of the SD card to ensure it is storing real-time data. This rate was tested by writing Arduino code that sent hard-coded values to the card for certain amounts of time and measured how much it sent. The average rate of storing data with one sensor connected t is 0.265 kb/s, and with no sensors connected 5 Kb/s. Comparing these rates with the SD card capability of 10 MB/s, it is clear that the required speed can be easily accommodated. Each data packet coming from its sensor will take 2 seconds to be saved.

The same functionality testing was performed on the EEPROM. Its functionality was tested with an Arduino test code that could write and read hard-coded values to and from

different locations simultaneously on EEPROM to ensure it can be written to. After the test was successful, a sensor was connected to the Yun to test it with actual real-time sensor data. Testing results matched with the expected results. It saved correct sensor data to the right location; testing was successful.

While testing, two things were discovered. First, the correct time-stamp is only available when the device is connected to Wi-Fi. The solution to this problem would be to use a RTC (Real-Time Clock) module. Second, for a while, when the SD card was disconnected, it kept showing on the serial monitor that it was writing to it, when it should have been writing to the EEPROM; it was not working correctly. This was eventually fixed by trying a different procedure to implement it and retesting it.

2) Wired/Wireless Connection

The first step in testing for the Home Network Connectivity feature was to make sure that the central hub was able to connect to the internet both wirelessly and wired by using both Wi-Fi and Ethernet. At this stage of the testing, the I.Smart Monitor was checked for its ability to connect to both home network system as well as hot spot internet provided by a smart phone. The results of the testing are summarized in table XI.2 below.

Table XI.2—Test Results of Connection to the Network

Test ID	Description	Expected Results	Actual Results	Pass/Fail
1	Make connection to the internet provided at home	Ability to make connection	Ability to make connection	Pass
2	Make connection to the mobile hot spot network provided by cell phone	Ability to make connection	Ability to make connection	Pass

Next, because it is important for the device to be easy to use for the parents of a newborn, it must be easy to configure the internet connection with minimum training or interaction by an infant's parents. To test this, the device was given to different random people who were asked to try and connect it to the internet. We gave the device to six people with different levels of technological knowledge in connecting devices to the internet. The results are summarized in table XI.3 below.

Table XI.3—Test Results of Configuring the Connection to the Network Easily

Test ID	Description	Expected Results	Actual Results	Pass/Fail
3	Make connection to the Internet easily by person 1	Easily configure the connection	Not easy to configure connection	Fail
4	Make connection to the Internet easily by person 2	Easily configure the connection	Easy to configure connection	Pass
5	Make connection to the Internet easily by person 3	Easily configure the connection	Not easy to configure connection	Fail
6	Make connection to the Internet easily by person 4	Easily configure the connection	Easy to configure connection	Pass
7	Make connection to the Internet easily by person 5	Easily configure the connection	Not easy to configure connection	Fail
8	Make connection to the Internet easily by person 6	Easily configure the connection	Not easy to configure connection	Fail

Based on the above test result, making the connection to the internet was not easy for a few people. So it was decided to develop and write the user manual for configuring the device for home network connection part in more detail and adding visual instructions which makes following them easier.

As stated in section B.1 above, the rate at which data is saved is 0.265 kbps. Comparing this measurement to the transferred data rate of sending data from central hub to the internet it is clear that the rate of transferring to the internet is above the rate of saving data into the SD card.

The average of sending data to the internet at different distances from the router with wired and wireless connection is about 115 kbps which shows that we are able to transfer the saving data from the SD card to the internet with a speed far higher than the data transfer rate of the SD card. Below, figure XI.1 is a floor plan which shows different locations that were used to test the data transfer rate to the internet with respect to the router indicated. The distance of each location from the router is shown in the below figure and is indicated by d, and each location is shown by a dot and is market with A, B, C, or D.

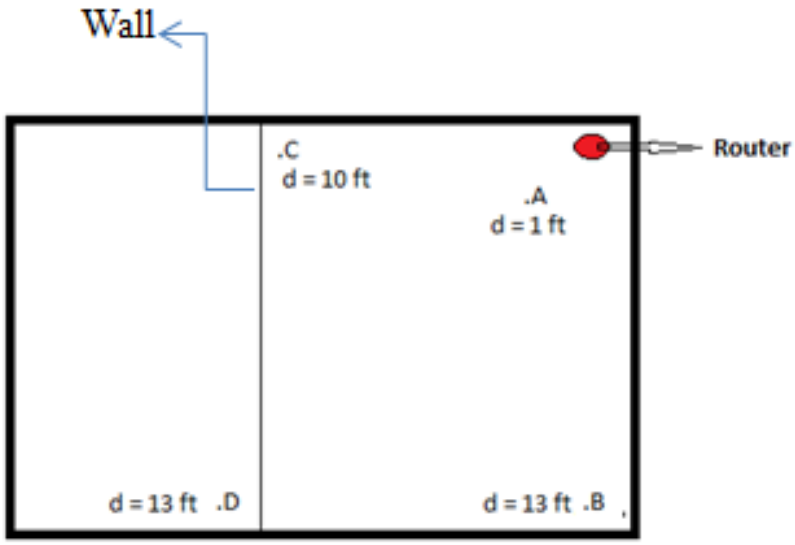


Figure XI.1—Floor Plan

The results of this stage of the test are summarized in table XI.4 below.

Table XI.4—Test Results of Data Transferred Rate

Test ID	Description	Expected Results	Actual Results	Pass/Fail
9	Measuring data with wire connection	Transferring data rate to the internet greater than saving data rate to the SD card	Transferring data rate is greater	Pass
10	Measuring data rate with wireless connection at A	Transferring data rate to the internet greater than saving data rate to the SD card	Transferring data rate is greater	Pass
11	Measuring data rate with wireless connection at A	Transferring data rate to the internet greater than saving data rate to the SD card	Transferring data rate is greater	Pass
12	Measuring data rate with wireless connection at A	Transferring data rate to the internet greater than saving data rate to the SD card	Transferring data rate is greater	Pass
13	Measuring data rate with wireless connection at A	Transferring data rate to the internet greater than saving data rate to the SD card	Transferring data rate is greater	Pass

3) Remote Access

The server was tested in two ways: communication from the hub, and to the website. The communication from the hub was tested both when there were low connection speeds and when there was a high ping rate. The device was connected to a network and the internet speeds throttled. A data delay was then introduced to test the time it takes to transfer data to the server successfully. It was then decided that the longest time gap in the data that could still be considered "near real-time" is about 10 seconds. These two scenarios determined the highest ping rate to be about 380ms and lowest speed possible for a

successful communication and data transmission to the server of about 128 kbs.

For the website communication, the website was checked on different operating systems and web browsers; it was found that the chrome browser worked best and is recommended to users for best results. The login feature of the website was also tested to ensure that permitted users only can view the required information. Without using sophisticated (and unsophisticated) hacking methods, the website was found to reject unknown passwords and allow entrance to registered passwords.

XII. INTEGRATION PLANS BASED ON TEST RESULTS

We will now discuss how each feature fits together with other features and integration with the device as a whole. This was very important to understand during the design process of the I.Smart Monitor in order to help mitigate and anticipate potential problems with the system as it began coming together. What follows is a discussion of this by feature.

A. Modular

The modular design of the I.Smart Monitor is really the heart of the system. The platform consists of multiple parts to accomplish its modular design. It consists of a central hub, and multiple sensors/sensor controllers. Medical sensors are directly wired to the sensor controllers. The connection between the sensor controllers and the hub is done through RS232 connectors. Each of the sensor controllers has an ATMEGA328 microcontroller. While the hub uses an Arduino Yun, due to its networking abilities. All of the other features use the modularity of the I.Smart Monitor to collect data and complete their tasks.

B. Event/Data Logger

The event/data logger functionality depends on the modular sensors. The sensor controller sends a five-byte data packet to the storage over the I^2C bus. The sensor data will be stored on an SD card. In addition, the name of the files on the SD card will be created based on the sensor type and the serial number. The serial number is the first byte and most of the second byte (15 bits) of the data packet. The MSB of the second byte contains an alarm bit that is used to activate the alarm when SD capacity reaches 80% and when EEPROM capacity reaches 100%. After data is stored to the SD card, it will be transferred to the internet. The remote access feature depends on the event and data logger feature. From this, we can see the integration of the event/data logger feature.

C. Alarm

The alarm is closely connected to both the Modular and Event/Data Logger features. Sensor-side, the alarm condition is detected by the sensor controller, which is the main component of the modular feature. The sensor controller sends a five-byte data packet to the main hub over the I^2C bus. Within these five bytes, there are two bits that contain both the serial number of the sensor (15 bits), and the alarm bit (MSB of higher-order byte). Because of this, the functionality of the alarm system is entirely dependent on the sensor controller working correctly, as well as the I^2C bus. The alarm was designed initially as a standalone system with a monotonic square wave audio pattern that was merely triggered and not based on hard-coded numbers. A function was then developed based on testing, discussed above, to create a more appropriate sound, and this function was inserted into the code for the main hub. Previously, the alarm code was simply sending a text indicator of an alarm condition to the Arduino IDE's Serial Monitor window, which was used for proof of concept. Instead of calling an internal function of the Arduino's

Serial class, the code now calls the "alarm" function developed by the I.Smart Monitor team.

In addition to a real-time indicator, the alarm affects the data being recorded as well. This ability is crucial and is obviously dependent on the functionality of the Event/Data Recording system. In fact, this is what is meant by "event recorder."

Ideally, the alarm could also give indications about the overall "health" of the monitor and to do this, it would need to be able to receive indications from other features, such as the SD card used in the Event/data logger.

D. Home Network Connectivity

As stated earlier in the Home Network Connectivity feature section, the I.Smart Monitor needs to follow the standards approved for medical devices. The Arduino YUN microcontroller fully supports all required specifications. However, based on our test results, making a connection to the internet with the chosen microcontroller was not very simple for people with a low level of technical knowledge. So we decided to prepare and write a more detailed user manual for the device to make the connection to the internet easier to understand for our users. Above all, and because the Home Network Connectivity feature is one of the most important features of our design, we needed to design our device with a microcontroller that could provide data transfer rates for us. This feature is important because it is highly engaged with other features, the event/data logger and remote access. It has to support the transferring of data from the SD card to the internet and then to the server. This required transferred data rate has been fully verified by our test results as stated above. So the choice of the Arduino YUN microcontroller for our device was very beneficial, as it fully supports all required specifications for the home network connectivity feature. The only area that we need to work on is the user manual in order to make it easier to follow and understand for parents with low levels of technical knowledge.

E. Remote Access

The remote access feature primarily integrates with both the home network connectivity and the event/data logger features. After testing, we found that the FTP connection with the server is achievable, but the time it takes to transfers large files every time a reading is made, is too long. This also stops the whole process. To fix this, the files will only be uploaded after ten readings, and only the ten new readings will be attached to the files already on the server; as opposed to uploading whole files over FTP. Another change that we integrated after testing was checking for alarms. A different file for alarms is created so that it will be easier for doctors to read alarming events, so it would not be lost in all the other data.

XIII. CONCLUSION

The design of the I.Smart Monitor has been a long and involved process. It began as a simple abstract solution to the societal problem discussed in the Societal Problem section of this document. This solution was designed to help alleviate the difficulties involved with the premature birth of a child. These include the stress for the parent and infant, as well as the lack of efficient communication between parent and doctor. By creating the ability to monitor a child from home, the parents and infants will be much more comfortable and the doctors will be able to help more patients with ease. Once this concept was realized, we began to develop the functional blocks of a system that would accomplish this, as discussed in the Design Idea section. These functional blocks were then specified as a feature set which is discussed at length in section with the same name.

The project went through the stages of development and the design revisions that were deemed appropriate based on what was learned or realized at each stage. For example, the planned timeline was constantly revised in the first few months of the project to reflect these realizations as they came. As time went on, we became more aware of realistic time frames for each feature and these revisions became fewer and further between.

The first phase of the project, the laboratory prototype, was an enlightening process and it is now clear to us just how much work goes into designing an electronic solution to a problem, particularly one that is intended for use by the general public. The design itself takes a great deal of time in order to be rigorous and mindful of potential risks.

The project consisted of more than just hardware and software design. It consisted of funding and even a market review. We not only had to prove the cost of our project, but we had to do research and test the market. The idea of not only having a device, but also knowing what it is worth is a very valuable tool.

Once the device design was complete, our team had to work to develop a test plan to ensure our product both helped with our societal problem, but also met the goals we set out for ourselves in the design idea contract. Once that criterion was met, it is finally safe to say this is our story for the I.Smart Monitor.

REFERENCES

[1] Center for Disease Control and Prevention (Oct 29, 2013) *National Prematurity Awareness Month* [Online] Available: http://www.cdc.gov/features/prematurebirth/

[2] Committee on Fetus and Newborn. American Academy of Pediatrics (April, 2003) *Apnea, Sudden Infant Death Syndrome, and Home Monitoring.* [Online] Available: http://pediatrics.aappublications.org/content/111/4/914.full.pdf

[3] American Medical Association. (October 9, 2012) *Infant Home Apnea Monitors* [Online] Available: http://www.anthem.com/medicalpolicies/guidelines/gl_pw_a053619.htm

[4] P. S. D. Lomdon, "Potential Reduction in Unnecessary Visits to Doctors from Safe and Appropriate use of OTC Medicines Could Save Consumer and Taxpayers Billions Annually," Paul A. London and Associates, 2011. [Online]. Available: http://www.yourhealthathand.org/images/uploads/London_Cost_Study_061711.pdf. [Accessed 12 2013].

[5] USA National Innovation Marketplace (September 9, 2009) *Benchmark Infant Cardiac/Apnea Home Monitor System (Medical)* [Online] Available: http://innovationsupplychain.com/innovations/report.php?id=2048

[6] Centers for Disease Control and Prevention (2013). Births and Natality. National Center for Health Statistics. [Online]. Available: http://www.cdc.gov/nchs/fastats/births.htm

[7] Agency for Healthcare Research and Quality, Center for Financing, Access and Cost Trends (2010). Health Insurance: Premiums and Increases. National Conference of State Legislatures. [Online]. Available: www.ncsl.org/research/health/health-insurance-premiums.aspx#Private Sector Premium Tables By State

[8] California Department of Public Health (2013). State of California. Table 2-18: Live Births, California Counties, 2002-2011 [Online]. Available: www.cdph.ca.gov/data/statistics/Documents/VSC-2011-0218.pdf

[9] Martin, J., Hamilton, B., Ventura, S., Osterman, M., Matthews, T.J. (2013). Births: Final Data for 2011. National Vital Statistics Reports. [Online]. Available: www.cdc.gov/nchs/data/nvsr/nvsr62/nvsr62_01.pdf

[10] "Arduino YUN," Arduino, [Online]. Available: http://arduino.cc/en/Main/Products. [Accessed 8 Feb 2014].

[11] "Radio Frequency Wireless Technology in Medical Devices - Guidance for Industry and Food and Drug Administration Staff," Food and Drug Administration 13 Aug 2013. [Online]. Available: http://www.fda.gov/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm077210.htm. [Accessed 8 Feb 2014].

[12] J. Hartford, "Must-Know Standards and Tests for Wireless Medical Devices," 20 Feb 2012. [Online]. Available: http://www.mddionline.com/article/must-know-standards-and-tests-wireless-devices. [Accessed 6 Feb 2014].

[13] "Wi-Fi in Healthcare," WiFi Alliace, Feb 2012. [Online]. Available: http://www.silexamerica.com/uploads/common/whitepaper-wifi-in-healthcare.pdf. [Accessed 8 Feb 2014]

[14] BroadbandDSLReports.com (Oct 7, 2013) *Google: 2% of our User Base is Using IPv6* [Online] Available: www.dslreports.com/shownews/Google-2-of-Our-User-Base-is-Using_IPv6-126109 Date Accessed: 11/26/2013

[15] Stork image taken from: http://wecanbeaoriginal.com/blog/2011/08/free-svg-download-stork-and-baby-scal-mtc/

[16] Image taken from:
http://blogs.cfr.org/coleman/category/topics/health/page/2/

[17] Image taken from:
http://onlyhousemusic.org/vbulletin/showthread.php?p=1236420

[18] Image taken from: http://embedded-lab.com/blog/?p=2583

[19] Image taken from:
http://www.frys.com/product/7824139

[20] Image taken from:
https://www.sparkfun.com/products/11287

[21] Image taken from:
https://www.sparkfun.com/products/11021

[22] Image taken from:
https://www.sparkfun.com/products/10524

[23] Image taken from
http://arduino.cc/en/Main/ArduinoBoardYun?from=Main.ArduinoYUN

[24] User's manual taken from http://arduino.cc/en/Guide/ArduinoYun#toc14

GLOSSARY

A

Arduino – a family of single-board microcontrollers featuring an open-source and peer-tested set of libraries. The hardware is also open-source

ATmega328 – An AVR microcontroller designed and produced by Atmel. This is the controller use in the Arduino UN and Arduino YUN platforms.

B

Bilirubin levels – the amount of bilirubin contained in tissue beneath the skin. An indirect measure of liver functionality, since the liver normally removes bilirubin from the blood.

Blood-oxygen saturation – the percentage of hemoglobin that is carrying oxygen to the total hemoglobin in the blood. Used as a measure of oxygen intake from breath rate.

Bradycardia – an abnormally low heart-rate. For adults it is anything below about 60 beats per minute (bpm); for infants the threshold is about 110 bpm.

Byte – Eight bits of binary information. The basic unit of computer data storage.

C

CAT5 cable - A twisted-pair cable for carrying signals.

C-language – the programming language used in the Arduino Integrated Development Environment (IDE)

CMOS tri-state – A buffer which has three possible states, High, Low, and High impedance, or High-Z state. In the latter state, the device is effectively removed from the circuit. Used to prevent loading effects by circuit blocks that are not currently being used.

E

EEPROM – Electrically Erasable Programmable Read-Only Memory; Here a DIP IC featuring Flash technology used as a redundant backup for data.

Email – text messages sent over the internet.

F

^{o}F – Degrees Fahrenheit; a unit of temperature measure. Equal to 9/5* ^{o}C + 32 for degrees Celsius.

G

GB (Giga Byte) – 1 Billion Bytes = 1,000,000,000 Bytes

H

HIPAA – Health Insurance Portability and Accountability Act; legal framework used to standardize online transmission of medical information. Here used as an external definition for security standards.

Hot-swappable – Able to be connected and disconnected without the need to switch power off.

Hyperbilirubinemia (jaundice) – the accumulation of subcutaneous bilirubin to toxic levels.

I

I2C bus – the physical connections used to send data from slave to master; notated SDA and SCL for serial data and serial clock, respectively.

I2C protocol – Inter-Integrated Circuit; A data transfer protocol using two data wires plus ground and a master-slave dynamic.

Internet – The Wide Area Network (WAN) consisting of all interconnected computers in the world that use the internet protocol suite TCP/IP.

IPv4 – Internet Protocol version 4; the fourth version of the internet protocol which routes most traffic on the internet.

K

kB/s – Kilobits per second; a data transfer rate

L

Local Area Networks (LANs) – Small networks of computers used in businesses and private homes. A home network is an example of one of these.

M

Master-Slave – the names for the roles played on the I^2C bus by different devices. The slave cannot send or receive data unless instructed to do so by the master.

MB (Mega Byte) – 1 Million Bytes = 1,000,000 Bytes

Microcontroller – an electronic circuit consisting of a microprocessor, memory and I/O circuitry. Often used as a single-program computer for controlling hardware.

Micro SD card – a smaller version of an SD card.

Modular sensors – Sensors that are interchangeable and self-contained. They merely require a hub to take data pre-processed data in a standard form from them

MSB – Most Significant Bit; the bit with the highest weighted value, or the farthest bit to the left when written in standard form.

O

Onboard storage – storage on the device itself, as opposed to that uploaded to the server. Implemented with flash memory devices.

P

Parallax Propeller – a multi-core microcontroller which was considered as a possible platform for the I.Smart Monitor.

R

Remote Access – The ability to access data without being physically connected to the source of the data, such as over the internet.

Respiration – the act of breathing and the associated circulation of oxygen and carbon dioxide in the lungs.

S

SCL – Serial Clock; One of the wires in an I^2C bus; used to synchronize the data

SDA – Serial Data; one of the wires in an I^2C bus; sends synchronous data

SD card - a storage device used for removable storage. Using Flash technology allows this card to be extremely small yet contain many GB of storage capability.

SMS - Short Message Service; a text messaging service used by mobile carriers.

SPI – Serial Peripheral Interface; a synchronous serial data link de facto standard. SPI uses a master-slave dynamic and a dedicated slave-select wire for every slave on the bus as well as three data lines. It allows for full-duplex communication.

Sudden Infant Death Syndrome (SIDS) – the sudden death of an infant that is not predicted by medical history and remains unexplained.

T

TTL 7400 series – Transistor-Transistor Logic; an architecture of Integrated Circuit logic using Bipolar Junction Transistors. Very fast switching capabilities but is being supplanted by CMOS which has a much lower power consumption rate.

W

Wi-Fi Shield – A circuit board designed to allow the Arduino to access a Wi-Fi hot spot and transmit or receive from the internet.

WPA – Wi-Fi Protected Access; a security protocol developed by the Wi-Fi Alliance to secure wireless computer networks.

Wireless – the transmission and reception of an electronic signal without the use of wires—usually with radio frequency or optical signals.

World Wide Web – The collection of web pages accessible on the internet through a web browser.

APPENDIX A-RESUMES

A. Joseph Cacioppo

Joseph S. Cacioppo

Objective

To develop a career as an Electrical Engineer that can make use of my proven abilities in Electrical engineering and management.

Summary of Qualifications

- System Design and Fabrication
- Current DoD Security Clearance
- Experience in malfunction analysis and troubleshooting
- Understanding of flight theory, sub-system tie in, digital logic, aircraft electrical and hydraulic system
- Ability to function well on a diverse team or as an individual
- Installation and removal of Line Replaceable Units and other avionics systems
- Excellent time management skills and punctuality
- Willing to travel and work rotating shifts

Education

California State University, Sacramento (Sac State), Sacramento, California
Bachelor of Science, Electronics Engineering, 2013-Graduation 2014, Current Cumulative GPA: 3.83

California State University, Fresno (Fresno State), Fresno, California
Bachelor of Science, Electrical Engineering, 2009-2013 GPA: 3.78

Fresno City College, Fresno, California
Associate of Arts, Liberal Arts with Highest Honors, December 2009, GPA: 3.65

Relevant Skills and Coursework

- Knowledge of various programming languages (C++, Microprocessor, MATLAB, Wolfram Mathematica)
- AC and DC Circuit Analysis
- Basic Solid-State Theory
- Integrated Avionics Systems Theory
- Microprocessor and Computer Architecture
- Multi-disciplinary engineering experience
- Electronics Analysis
- Knowledge of Digital Logic Design
- Signals and Systems
- Transmission/Receiver System Theory
- RADAR Homing/Warning System

Experience

Alpha Research and Technology El Dorado Hills, California, 2013
Systems Engineer Intern
- Acting project lead on Intelligent Display Panel
- Redesigned entire system interface while working closely with manufacturing, mechanical, and electrical engineers
- Reduced resources needed by 30% to complete project during system redesign
- Sought out, tested and implemented new product for redesign after recognizing system issues

US Air Force Edwards Air Force Base, California 2011
F-16 Link 16 and Tactical Data Link Engineer Intern
- Designed and fabricated multiple hardware interfacing devices for Tactical Data Link terminal and radio communication
- Tested the Integrated Data Modem and Link 16 system functionality in accordance with manufacturer's specifications and operational effectiveness
- Performed ground and flight tests using a Military Rugged Tablet, Battlefield Operational Support System, PRC-117 and ARC-210 Radios within the ground station and the control room

US Air Force Worldwide, 2003-2007
F-16 Avionics Systems Technician
- Maintenance, troubleshooting, upgrading, and programming the various avionics systems on the F-16
- Analyzed wiring diagrams to solve issues associated with system and sub-system tie-in
- Loaded and upgraded the current Operational Flight Program for the F-16 avionics systems
- Followed Technical Orders and safety practices to ensure proper maintenance was performed
- Maintained and recorded flight times, maintenance schedules and aerial refueling data for the entire squadron

Honors/Awards

- National Science and Mathematics Access to Retain Talent Grant 2010-2011
- Dean's List and President's List every semester, 2008-2013
- President of SIAM's Fresno State Chapter, 2012-2013
- Graduated with Highest Honors from Fresno City College, 2009
- Air Force Outstanding Unit Award, 2006

B. Duaa Salah

Duaa Salah

OBJECTIVE:

Seeking an entry level position in Computer Engineering

EDUCATION:

In Progress: Bachelor of Science, Computer Engineering, CSU Sacramento; expected graduation: May 2014

RELATED COURSES:

Advanced Logic Design	Intro to System Programming	Operating System Principles
Network Analysis	Programming Concepts and Methodology	Operating System Pragmatic
CMOS and VLSI Design	Micro-computer Assembly Language Programming	Computer Software Engineering
Computer Interfacing	Data Structures and Algorithm Analysis	Intro to Digital Signal Processing
Advanced Computer Org.	Computer Network & Internet	Signals & Systems
Computer Hardware Design	Organization Database Management & File Org.	Discrete Structures

KNOWLEDGE AND SKILLS

- ***Computer Languages:***
  C, Java, Assembly, Verilog, VHDL, Spin, ,MIPS, MySQL, JavaScript, HTML, Python
- ***Hardware/Software:***
  Xilinx ISE, FPGAs, ModelSim, MultiSim, Microsoft Office, Open Office, Microsoft Project, Microsoft Visio, Math Type L-Edit, PSpice, VNC Viewer, VMware Workstation, Arduino Software, Propeller Tool, Graphic Analysis, CircuitMaker, MATLAB
- ***Tools:***
  Oscilloscope, Waveform Generator, Multimeter
- ***Operating System:***
  Windows XP, Windows 7, Unix, Linux

WORK EXPERIENCE:

Student Assistant **California Department of Transportation** **Current**

Working as student assistant in the IT department. Perform routine maintenance tasks related to the database software and prepare technical assistance requests for the IT manager. Consult with staff to access additional database needs and improvements to database and reporting requirements. Perform data entry using word processing, spreadsheet or database commands. Create data directories and subdirectories for file and report generation retrieval purpose and maintain a disk file of entered data.

PROJECT EXPERIENCE:

Senior Design Project

Worked with a five-member team to develop a device to monitor an infant's health at home which can be remotely accessed by physicians and/or caregivers. The monitor is easy to use, monitors multiple vital signs, logs events and data, and activates an alarm when any vital sign is outside of a predefined range.

Computer Interfacing Project

Worked with a group to design PWM (Pulse Width Modulation) Fan Controller that control motor speed with basic system features in addition to some integration of advanced system features, such as: temperature dependence and smooth speed ramp.

CMOS & VLSI Project

Worked with another colleague to design and layout the control logic circuit for a 4-bit successive approximation analog-to-digital converter in 0.5 um CMOS. We designed a block diagram for the circuit, gate level, transistor level and layouts using L-Edit.

Logic Design Project

Designed a unique user generator feature which was Egyptian characters using VHDL language and displayed them by using LCD on Spartan 3E and VGA.

Asynchronous FIFO Controller

Implemented the design using hierarchical design methodology. There were three modules FIFO read, FIFO write and FIFO memory. Used ISE design tool to simulate the waveforms. Generated test bench for the Verilog code to verify the working of the desired design.

16-bit MIPS Processor

Worked with another colleague through the design, development, and implementation phases of a 16-bit MIPS processor with a 5-stage pipeline. Used Behavioral modeling in Verilog to implement load/store word operations, integer arithmetic, and branching. Also, implemented simple branch prediction, forwarding and hazard detection.

Hardware System Design

Worked with another colleague to design a 32-bit Target PCI Memory Card and a level-2 Cache controller.

ACIVITIES

Member of Institute of Electrical and Electronics Engineers (IEEE)
Member of IEEE Women in Engineering (WIE)

C. Daniel Schmidt

Daniel Schmidt

Objective: To become as skilled in the design of control systems and robots as I possibly can

Education:

California State University, Sacramento: 2012 to Fall 2014
Grade Point Average: 3.82
Degrees in Progress:
- B.S. Electrical Engineering – Controls

Sacramento City College: 2007 to 2012
Grade Point Average: 3.84
Degrees Completed:
- A.S. Electronic Facilities Maintenance Technician , Telecommunications Technician, Mathematics

Programs and Courses Studied:
- Computer Repair
- Operating Systems Experience (Windows & Linux)
- Soldering and high-tech assembly
- AC/DC theory and analysis
- Semiconductor theory
- Microprocessors and digital circuits
- Receivers and transmitters
- Mathematics
- General Chemistry
- General Physics
- Network analysis
- Transistor amplifier design
  - Bipolar Junction
  - MOSFET
- Logic design, HDL - Verilog
- Analog/Digital control system design
- Machine Vision
- Microcontroller programming
  - Atmel ATmega328
  - Parallax Propeller
- Robotics

Experience:
- Programming Languages:
  - C/C++
    - Structured and Object-Oriented – including recursive and polymorphic functions
  - Python
    - Structured and Object-Oriented
  - Intel Assembly
- Software:
  - Pspice, Multisim, ADS
  - MATLAB
  - MS Office Suite, Apache Open Office Suite

Projects:
- Senior Design:
  - Home infant monitor with hot-swappable, self-identifying sensors and Web-based interface
    - Designed hot-swappable 'smart sensor' modules which identify themselves as well as check sensor data for threshold levels indicative of a medical emergency and alert main hub
    - Designed interface between sensors and main hub using Inter-integrated Circuit (I^2C) Bus
    - Constructed all hardware components
    - Performed device and sensor hardware testing
    - Participated in Idea-to-Product competition for Biomedical design projects
- Robotics:
  - Laser-guided PID-controlled mobile robot
    - Wrote code for Proportional-Integral-Derivative control of robot's position
    - Tested robot to determine optimal value for PID tuning parameters
    - DC motors
  - Autonomous mobile robot with Infrared and Ultrasonic obstacle detection
    - ATmega328
    - Proportional Control of DC motors
  - Autonomous mobile robot with Infrared obstacle and edge detection
    - Parallax Propeller with C/C++
    - Continuous rotation servos
  - PID speed controller for DC motor using I^2C interface (Sensor to Controller)

D. Mahsa Shadmani

Mahsa Shadmani

OBJECTIVE: ENTRY LEVEL POSITION IN COMPUTER HARDWARE ENGINEERING

EDUCATION: Bachelor of Science Computer Engineering, 3.48 GPA.
CSU Sacramento, Graduation date Spring/2014

RELATED COURSES:

Operating System Principles	Signals & Systems	Advanced Computer Organization
Network Analysis	Advanced Logic Design	CMOS and VLSI
Data Structures and Algorithm	Computer Hardware System Design	Computer Networks and Internet
Software & Engineering Operation	Computer Interfacing	System Programming in Unix
Embedded Processor System Design (IP)	Operating System Pragmatics	Introductory Circuit Analysis

PROJECT EXPERIENCE:

Senior Project Design, I-Smart Monitor

Member of a group of five-student that design and develop a central hub by using Arduino YUN which able the parents of newborn to monitor the health of their infant at comfort of their home which can be remotely accessed by physician/caregiver via a secure website.

Home Automation

Member of a four-student team design and develop a Home Automation that are using wireless communication to transmit signals throughout a home. The propeller will transmit data through an XBee wireless transmitter. The XBee receiver module and send the data to the Arduino which controls a 120V device, a microprocessor, and an LED light

VLSI Design by L-edit

Design and Layout of a 3-Bit Serial Adder with Accumulator in 0.5 μm CMOS

16-bit MIPS Processor

Led a two-person team through the design and development and implementation phase of a 16-bit MIPS, processor with a 4-stage pipeline, Behavioral modeling in Verilog was used to implement load/Store word operation, integer arithmetic, and branching, Simple branch prediction, forwarding and hazard detection were also implemented

PCI Memory Card & Level2 Cache Memory

Led a two-person team through the design and development phase of 32 bit Target PCI Memory Card and also design a Level-2 Cache Memory

KNOWLEDGE AND SKILLS:

Languages: C, MIPS, ASCII, X86 Assembly, Verilog, VHDL, Java, JavaScript, HTML

Software and Tools: Unix/Linux, InetSoft, Xilinx ISE, Multisim, ModelSim, PSpice, L-edit, Oscilloscope, Auto-CAD

Communication/Organization/Leadership:

Excellent organization and time management skills, Effective leadership and team skills, Strong analytical and problem solving skills, Ability to adapt to new situations and technology quickly, Excellent technical report writing skill

WORK EXPERIENCE:

Student Assistant	**Legislative Data Center**	11/01-Present
Office Assistant	**Marin Eye Care**	8/08 – 10/13
Sales Associates	**Macy's**	10/07 – 5/08

AWARDS AND ACCOMPLISHMENTS: SRJC Dean's Honor List CSUS Dean's Honor List
Three years of studying in Chemical Engineering Field

VOLUNTEER WORK: Help earthquake victims of Bam 2003

WORK STATUS: **U.S Citizen with permanent unrestricted right to work in the U.S**

E. Vasiliy Warkentin

Vasiliy Warkentin

Education

California State University, Sacramento

Expected Graduation December 2014| Bachelor of Science, Computer Engineering

Blagovest Institute, Sacramento, California

August 2013| Bachelor of Church Ministry

Experience

Chief Executive Officer May 2013 – Present

Russian Baptist Church | 1000 Sacramento Ave. West Sacramento, CA 95605

- Director, decision maker, leader, manager and executor of the Church Board decisions

Network Administrator/Video Department manager January 2006 – May 2013

Russian Baptist Church | 1000 Sacramento Ave. West Sacramento, CA 95605

- Maintain Facility Network and Technology
- Manage the weekly video needs of the church
- Manage the 22 people video team

Skills

- Effective leadership and team skills
- Microprocessor and Computer Architecture
- Signals and Systems
- C Programing language

APPENDIX-B HUB MAIN CODE

```
#include <FileIO.h>
#include <EEPROM.h>
#include <Wire.h>
#include <Process.h>

const int SPKR_PIN = 13;
const int LED_PIN = 12;

//-------External EEPROM
const byte EEPROM_ID = 0x50; // I2C address for 24LC128 EEPROM
int TIMEUP_ADDR = 0;
int CLOCK_ADDR = 1;
int BUS_ADDR = 2;
int SERIAL_ADDR = 3;
int ID_ADDR = 4;
int DATA_ADDR = 5;

int kill = 0;
int rewritingInProress = 0;
unsigned long timeup;

//------I2C BUS
int devices = 127;
int id[127] = {0};

void setup() {
  initialize();
}

void loop() {
  int Internet = wifiCheck();
  String dataString;

  timeup = millis();
  timeup = timeup /1000;
  //delay(6000);

  dataString += getTimeStamp();
  dataString += " = ";
  byte dt =  dataString.toInt();
 // int dt =  getTimeStamp().toInt();
 // delay(1000);
  //Console.println();
  readFromAll(dataString);
}

int wifiCheck(){
```

```
  Process wifiCheck;  // initialize a new process

  wifiCheck.runShellCommand("/usr/bin/pretty-wifi-info.lua");  // command you want to run

  // while there's any characters coming back from the
  while (wifiCheck.running());
     int result = wifiCheck.parseInt();                    // look for an integer
     int result2 = wifiCheck.parseInt();                   // look for an integer
     int result3 = wifiCheck.parseInt();                   // look for an integer
     int result4 = wifiCheck.parseInt();            // look for an integer
     return result4;
}

void initialize(){
 // Initialize the Bridge and the Serial
  Bridge.begin();
  Console.begin();
  FileSystem.begin();
  while (!Console); // wait for serial port to connect
  Wire.begin();
  Console.println("iSmart Monitor\n");
}

void readFromAll(String dataString){
 int c = 1;
 while(c < devices){
   Wire.requestFrom(c, 5);
   if(Wire.available()){
     readFromOne(c, dataString);
   }
   if(c==79)c++;  //EEPROM on bus 80
   c++;
  }
}

void readFromOne(int c, String dataString){
  float data = 0;
  byte ID,
     SERIAL_HB,
     SERIAL_LB,
     alarm_bit,
     data_hb,
     data_lb;
  int serial;

      ID = Wire.read();
      SERIAL_HB = Wire.read();
      SERIAL_LB = Wire.read();
```

```
    data_hb = Wire.read();
    data_lb = Wire.read();

    alarm_bit = SERIAL_HB / 128;
    serial = (SERIAL_HB * 256 + SERIAL_LB) - alarm_bit * 32768;

    if(ID == 2){
      data = (float)((data_hb * 256 + data_lb) / 100.0);
    }
    else if(ID == 1){
      data = (float)((data_hb * 256 + data_lb) / 100.0);
    }
    else{
      data = data_hb * 256 + data_lb;
    }

    writeData(c, ID, alarm_bit, serial, data, dataString, timeup);
}

void writeData(int c, byte ID, byte alarm_bit, int serial, float data, String dataString,  long timeup){
    byte dt =  dataString.toInt();
    writeToSD(c, ID, alarm_bit,serial,data,dataString, timeup);
   // writeToFTP();
   if(wifiCheck()){         //there is internet connection
      writeToFTP();
      alarm_led();
   }
              // & LED   */
    writeToSerial(c, ID, alarm_bit,serial,data,dataString, timeup);
   // writeToEEPROM(c, ID,serial,data,dt, timeup);
    //Console.println();
}

int writeToSD(int c, byte ID, byte alarm_bit, int serial, float data, String dataString, long timeup){

   Process p;
   p.runShellCommand("df | grep dev/sda1");
   while (p.running());
   int result = p.parseInt();                    // look for an integer
   int SDsize = p.parseInt();                    // look for an integer
   int SDused = p.parseInt();                    // look for an integer
   int SDavailable = p.parseInt();            // look for an integer
   int SDpersent = p.parseInt();

   if (result == 1){
    String sFile = "/mnt/sda1/" + String(ID) + String(serial) + ".txt";
```

```
char file[100];
sFile.toCharArray(file, 100);

File dataFile = FileSystem.open(file, FILE_APPEND);
          ///////////////// clean up to represent right formating.
  if (DATA_ADDR > 5){                  //If SD is reconnected, upload from EEPROM to SD
     Console.println("Uploading Data from EEPROM to SD");
     for (int i=0; i<DATA_ADDR; i++)
     {
       int remainder = i%6;
       if(remainder==0){
        dataFile.print("Time Up: ");
        dataFile.print(I2CEEPROM_Read(i));
        dataFile.println(" Seconds");
       }
       else if (remainder == 1){
         dataFile.print("Real Time: ");
         dataFile.println(I2CEEPROM_Read(i));
       }
       else if(remainder == 2){
         dataFile.print("Sensor Bus: ");
         dataFile.println(I2CEEPROM_Read(i));
       }
       else if (remainder == 4){
         dataFile.print("Sensor ID: ");
         ID = I2CEEPROM_Read(i);
         dataFile.print(ID);
          if (ID == 1){
             dataFile.println(" Temperature Sensor");
           }
          else if (ID == 2){
           dataFile.println(" Pulse Sensor");
           }
       }
       else if (remainder == 3){
         dataFile.print("Serial number: ");
         dataFile.println(I2CEEPROM_Read(i));
       }
       else if (remainder == 5) {
         dataFile.print("Sensor Data: ");
         dataFile.print(I2CEEPROM_Read(i));
       if (ID == 1){
         dataFile.println("F");
         }
       else if (ID == 2){
         dataFile.println(" bpm");
        }
         dataFile.println();
```

```
    }

    I2CEEPROM_Write(i,0);
   }
   dataFile.println();
   dataFile.println();
   dataFile.println();
   dataFile.close();
   TIMEUP_ADDR = 0;
   CLOCK_ADDR = 1;
   BUS_ADDR = 2;
   SERIAL_ADDR = 3;
   ID_ADDR = 4;
   DATA_ADDR = 5;
 }
 //// SD Capacity options//////
 if(SDpersent == 80) {         //gets to 80%, alaram will sound
  Serial.println("SD is 80% full");
  dataFile.println("ALARM!");
  alarm_tone();
 }
 if(SDpersent == 95) {        //gets to 95%, recodard alarming data:time, id, alarm, data
  Console.println("SD is almost full");
  dataFile.println(timeup);
  dataFile.println(dataString);   //time
  dataFile.print("Sensor ID: ");
  dataFile.println(ID);
  if(alarm_bit){
   // alarm_tone();
    dataFile.println("ALARM!!!!");
  }
  dataFile.print("Sensor Data: ");
  dataFile.print(data);
  if (ID == 1){
    dataFile.println(" F");
  }
  else if (ID == 2){
     dataFile.println(" bpm");
  }
     dataFile.println();
     dataFile.close();
}
if(SDpersent == 100) {       //gets to 100%, overwrite oldest data
  Console.println("SD is full");
  if (rewritingInProress == 0){
     dataFile.rewindDirectory();  //will bring you back to the first file in the directory on an SD card
     dataFile.seek(0);      //seek to position 0 in dataFile
     rewritingInProress = 1;
```

```
    }
   }
          ////////////////////////////
     if(dataFile){
     Console.println("Writing to SD card");
     dataFile.print("Time Up: ");
     dataFile.print(timeup);
     dataFile.println(" Seconds");
     dataFile.print("Real Time: ");
     dataFile.println(dataString);   //real-time
     dataFile.print("Sensor Bus: ");
     dataFile.println(c);
     dataFile.print("Sensor ID: ");
     dataFile.print(ID);
     if (ID == 1){
        dataFile.println(" Temperature Sensor");
      }
     else if (ID == 2){
        dataFile.println(" Pulse Sensor");
     }
     dataFile.print("Serial number: ");
     dataFile.println(serial);
     if(alarm_bit){
      //alarm_tone();
      dataFile.print("ALARM!!!!");
     }
     dataFile.print("Sensor Data: ");
     dataFile.print(data);
     if (ID == 1){
      dataFile.println("F");
      }
     else if (ID == 2){
      dataFile.println(" bpm");
      }
     dataFile.close();
     dataFile.println();
     return 0;
     }
   }

  else{
     Console.println("Error opening SD....");
     byte dt =  dataString.toInt();
     writeToEEPROM(c, ID,serial,data,dt,timeup);    //If SD is disconnected, write to EEPROM
    return 1;
  }
}
```

```
void writeToFTP(){
  Process ftp;
  if (kill == 0){
  Console.println("Transfering files to FTP");
  ftp.runShellCommandAsynchronously("lftp -u ismart,Team6Monitor -e 'put /mnt/sda1/12.txt'
ftp://66.197.182.125");
  ftp.runShellCommandAsynchronously("lftp -u ismart,Team6Monitor -e 'put /mnt/sda1/11.txt'
ftp://66.197.182.125");
  }
  kill++;
  if(kill == 10){
    kill = 0;
    killFTP();
  }
  Console.println();
}

void killFTP(){
  Process kill;
  kill.runShellCommandAsynchronously("killall lftp");
  kill.runShellCommandAsynchronously("killall lftp");
  kill.runShellCommandAsynchronously("killall lftp");
  kill.runShellCommandAsynchronously("killall lftp");
}

void writeToSerial(int c, byte ID, byte alarm_bit, int serial, float data, String dataString, long timeup){
  Console.print("Time Up: ");
  Console.print(timeup);
  Console.println(" Seconds");
  Console.print("Real Time: ");
  Console.println(dataString);
  Console.print("Sensor Bus: ");
  Console.println(c);
  Console.print("Sensor ID: ");
  Console.print(ID);
  if (ID == 1){
    Console.println(" Temperature Sensor");
  }
  else if (ID == 2){
    Console.println(" Pulse Sensor");
  }
  Console.print("Serial number: ");
  Console.println(serial);
  if(alarm_bit){
    Console.println("ALARM!!!!");
    alarm_tone();
  }
  Console.print("Sensor Data: ");
```

```
  Console.print(data);
  if (ID == 1){
    Console.println("F");
  }
  else if (ID == 2){
    Console.println(" bpm");
  }
  Console.println();
}

void writeToEEPROM(int c, byte ID, int serial, float data, byte dt, long timeup){
    //Writing to EEPROM
    int i;
    Console.println("Writing to EEPROM");
    I2CEEPROM_Write(TIMEUP_ADDR, timeup);
    I2CEEPROM_Write(CLOCK_ADDR, dt);
    I2CEEPROM_Write(BUS_ADDR, c);
    I2CEEPROM_Write(ID_ADDR, ID);
    I2CEEPROM_Write(SERIAL_ADDR, serial);
    I2CEEPROM_Write(DATA_ADDR, data);
    TIMEUP_ADDR+=6;
    CLOCK_ADDR+=6;
    BUS_ADDR+=6;
    ID_ADDR+=6;
    SERIAL_ADDR+=6;
    DATA_ADDR+=6;

   ///EEPROM capacity options/////
   int ESIZE = (i/1024000000);
    if (ESIZE == 0.90) {      //when it's 90% full
       alarm_led();               //LED
       Console.println("ALARM: EEPROM is almost full & NO Internet Connection!");
    }
    else if (ESIZE == 1) {      //when it's 100% full
       alarm_tone();
       Console.println("ALARM: EEPROM is full & NO Internet Connection!");        //ALARM
       return;                        //it will stop recording
    }
}

void I2CEEPROM_Write( unsigned int address, byte data ) {
  Wire.beginTransmission(EEPROM_ID);
  Wire.write((int)highByte(address) );
  Wire.write((int)lowByte(address) );
  Wire.write(data);
  Wire.endTransmission();
  delay(5); // wait for the I2C EEPROM to complete the write cycle
}
```

```
byte I2CEEPROM_Read(unsigned int address )
{
 byte data;
 Wire.beginTransmission(EEPROM_ID);
 Wire.write((int)highByte(address) );
 Wire.write((int)lowByte(address) );
 Wire.endTransmission();
 Wire.requestFrom(EEPROM_ID,(byte)1);
 while(Wire.available() == 0) // wait for data
 ;
 data = Wire.read();
 return data;
}

String getTimeStamp() {
 String result;
 Process time;
 time.begin("date");
 time.addParameter("+%D-%T");  // parameters: D for the complete date mm/dd/yy
                               //             T for the time hh:mm:ss
 time.run();  // run the command
 // read the output of the command
 while(time.available()>0) {
  char c = time.read();
  if(c != '\n')
    result += c;
 }
 return result;
}

void alarm_tone(){
   for(int i = 500; i > 0; i-=50)
   {
      tone(SPKR_PIN, i);
      delay(50);
      noTone(SPKR_PIN);
   }
  for(int i = 0; i < 500; i+=50)
  {
     tone(SPKR_PIN, i);
     delay(50);
     noTone(SPKR_PIN);
  }
}

void alarm_led(){
   for(int i = 500; i > 0; i-=50)
```

```
   {
      tone(LED_PIN, i);
      delay(50);
      noTone(LED_PIN);
   }
  for(int i = 0; i < 500; i+=50)
  {
    tone(LED_PIN, i);
    delay(50);
    noTone(LED_PIN);
  }
}
```