2011-2012 Global Clinic: Non-Intrusive Monitoring System for Isolated Elderly

Delivered: May 4, 2012

工学院大学
KOGAKUIN UNIVERSITY

BROADCOM®

Spring Semester Project Team

Harvey Mudd College Group
Benjamin Liu (Fall Team Leader)
Noah Duncan (Winter Team Leader)
Edward Wang (Spring Team Leader)
Samantha Ipser (Spring Junior)
Shinsaku Nakamura (Spring Exchange)

Kogakuin University Group
Masahiro Kato (1st Team Leader)
Yohei Katayama (2nd Team Leader)
Kazuma Ouchi (3rd Team Leader)
Kyosuke Kusaka (4th Team Leader)

Harvey Mudd College Advisors
Professor Lisette dePillis
Professor Patrick Little
Professor Okitsugu Furuya

Kogakuin University Advisors
Professor Yukio Nakajima
Professor Masanori Shiomi
Professor Yoshihito Yasaki
Professor Sebastian Brooke

ABSTRACT

The Global Clinic team at Harvey Mudd College (HMC) in Claremont, California is collaborating with Kogakuin University (KU) in Tokyo, Japan to develop a non-intrusive monitoring system for isolated elderly individuals in Japan. The proportion of elderly people in Japan is growing steadily. This growth, along with the fact that more and more Japanese elderly live alone and geographically distant from their families, is driving an increasing need to provide these isolated elderly individuals a sense of physical security, along with a way to stay connected to family members.

Currently available monitoring systems are not fully meeting the needs of the elderly and their families, mostly because existing solutions require users to alter their lifestyles. In response to the need for an improved monitoring and communications system, the Global Clinic team is creating a suite of user-friendly instrumented form factors that, once equipped with a variety of sensors, can measure and transmit important health data about an elderly person to family members. The current collection of sensors the team has chosen can measure heart rate, body temperature, body position, ambient temperature, humidity and geographic location. These particular sensors were chosen as a result of surveys carried out in Japan, and consultations with medical experts. The Global Clinic team has created a demonstration-ready prototype in the form of a sensor-embedded walking cane. An accessory clip form factor and a remote-control form factor are currently under development. A user interface that can read the wirelessly transmitted sensor data has been built on top of an online web-based data base. This report provides the full details on the Harvey Mudd College Global Clinic 2011-2012 Project.

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

Japan is facing the phenomenon of an aging population. Elderly individuals in Japan are becoming increasingly isolated, with no one to look after them as their health deteriorates. In addition to health deterioration, the Japanese government has also identified a gradual deterioration of mental health, due to factors such as feeling lonely from a lack of contact with family and friends. To address this issue, the Japanese government has been introducing various ways to make sure the safety and health of elderly individuals is monitored. The end goal of the monitoring is to bring happiness and reassurance to individuals in Japan by connecting the elderly individuals with remote family members through a user-friendly monitoring system.

At the local prefecture level, many projects have been funded to develop monitoring devices for elderly individuals. However, the existing systems require the user to adjust his or her lifestyle to incorporate the new device into their daily routines. The inconvenience not only makes it hard for users to remember to use the device, it also poses a problem with wide spread adoption.

Countering existing systems, which address only functionality, the Global Clinic team decided to approach the problem from a user centric perspective. Instead of designing for monitoring functionalities, the project's main goal was to develop monitoring devices that non-intrusively integrate into a person's current lifestyle. To achieve this, the team chose to convert items that an elderly individual would commonly use on a daily basis into monitoring devices. A well-chosen form factor streamlines the process of health monitoring so that elderly individuals can have their well-being measured and analyzed without being constantly reminded that they are using a monitoring device.

This project was a joint collaboration between student groups at Harvey Mudd College (HMC) in Claremont, California and Kogakuin University (KU) in Tokyo, Japan. In August 2011, the HMC group met with the Japanese group at the KU campus to produce the team charter and decide on preliminary project goals. For the duration of the 2011-2012 academic year, the Global Clinic team divided the project work between the groups at both institutions. The HMC group focused on the analog/software design and the KU group focused on market surveys and the physical user interface.

This report covers the background information for the project, form factors, sensor hardware implementation, and software implementation for data acquisition, storage, and display. Section 3 will provide background information about the project. This information includes the project's motivation, the research conducted by the team, the visits to existing service/device providers, the technical aspects of existing devices, a report on the economics of the Japanese telehealth system, and a discussion of the marketing aspects involved in the project. Section 4 will formally define the project, which includes the project statement, the project's division into modules, and the project deliverables. Section 5 will provide a list of the ideas for possible form factors of the project that are in development. Sections 6 and 7 will describe the technical aspects of the existing prototype for both electronic hardware and software. Sections 8 will describe the final deliverables of the project as of May 2012. Section 9 discusses plans for project continuation.

2. WORK BREAKDOWN AND INTERNATIONAL COLLABORATION

Global Clinic is an extension of both HMC's Clinic Program and KU's Engineering Clinic Program (ECP) that provides a chance for students to participate in international collaboration as well as a cross-cultural exchange. During the weeklong visit by the HMC students to Japan in August 2011, the Global Clinic team focused on developing the work breakdown structure of this year's Global Clinic as well as laying the foundation for research. These preparations allowed each country's group to operate separately while still contributing to common goals. Through this summer session, the Global Clinic team defined the scope of the project and began dividing tasks based on the particular background and training of each side.

Kogakuin University was well suited to conduct cultural research, so the KU group focused on market research and form factor design that will fit with elderly lifestyles. The KU group conducted several surveys in order to determine some aspects of elderly lifestyles, such as daily activities, common household items usage, and frequency of contact with other people. The KU group's research has contributed to a thorough understanding of the Japanese elderly individuals. This understanding ensures that any produced system designs conformed to these individuals' lifestyles. The KU group was also responsible for the development for two of the three different form factors that house the sensor platform developed by the HMC group. The KU group will be continuing the project until January 2013.

The Harvey Mudd College group took responsibility for development of sensor implementations and software coding. The HMC group developed and constructed the sensor platform, the transmission module, and the software for the user interface. These tasks involved determining the types of sensors and processors to use and ensuring that the circuitry works as expected. To connect the various sensors to the microprocessor, the HMC group has had to create several printed circuit board (PCB) designs constrained by size limitations dictated by the form factors. In addition, this group has developed software code that acquired, processed, uploaded, and displayed the sensor data in a user-friendly fashion. Finally, the HMC group developed one of the three form factor options into a fully functional prototype.

In Spring 2012, the KU group made a three week visit to the HMC campus. The purpose of this visit was to solidify future goals for the project as well as to more thoroughly analyze each group's ideas for the system prototypes. During these three weeks, the Global Clinic team focused on generating ideas for a variety of viable form factors to better define the physical constraints that the embedded sensors needed to meet in the final deliverable to HMC. After three weeks of joint project work, the KU group returned to Japan, and each group continued to work toward the completion of several prototypes.

3. PROJECT BACKGROUND

3.1. MOTIVATION

The demographics of Japanese society, as with much of the world, have been gradually shifting toward a higher proportion of elderly individuals (Figure 1). Additionally, the number of Japanese elderly individuals who live alone has been increasing rapidly. From 1997 to 2003, this category rose by 39.3% to over 3 million individuals. As of 2005, 15% of all elderly individuals live alone [1]. Many of the isolated elderly live in rural Japan, due to a tendency for younger Japanese to seek jobs in urban areas, leaving their older relatives behind. For these isolated elderly individuals, in addition to the physical dangers that stem from living alone, there is a risk of gradual deterioration in mental health due to factors such as feeling lonely from a lack of contact with family and friends.

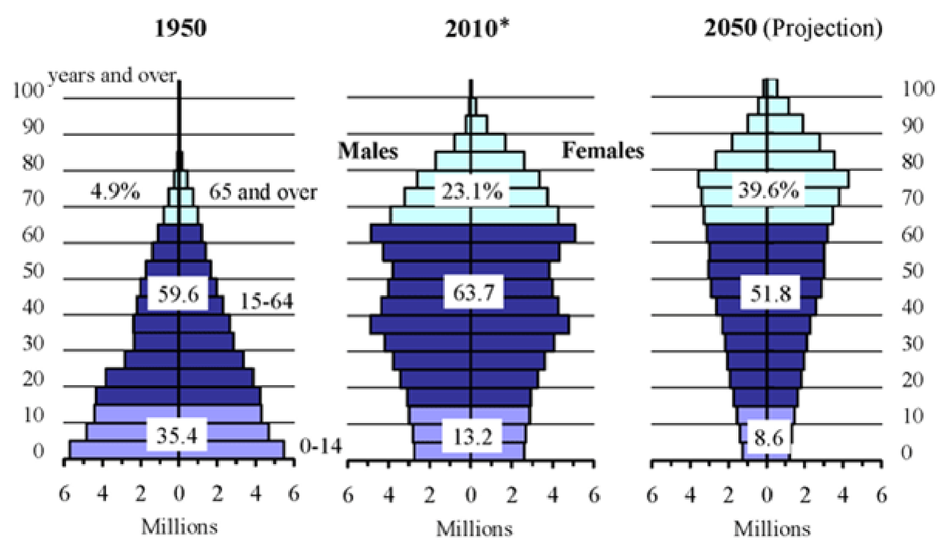


Figure 1: Past and Future Changes in the Japanese Population Pyramid [2]

3.1.1. Demographics of Japanese Elderly in Need of Care

A national 20 year longitudinal survey of the elderly in Japan was conducted to study changes in elderly mobility. This study measured the gradual change of dependency as people aged. Out of 6,000 male subjects, 19% experienced early decline, 70.1% experienced gradual decline until death, and 10.9% were independent until death (Figure 2). Out of 6,000 female subjects, 12% experienced early decline, 88% experienced gradual decline, and none remained completely independent (Figure 3). With a higher percentage of female elderly individuals living past their 70s, the majority of elderly are in gradual decline and need support and monitoring by others (Figure 4).

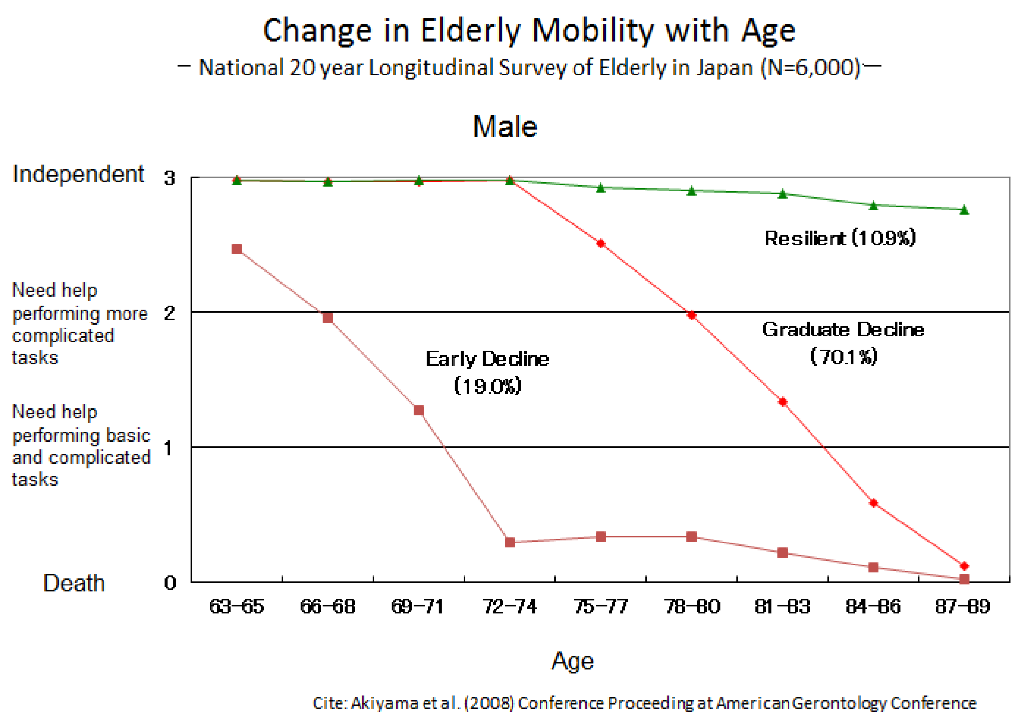


Figure 2: Change in Elderly Mobility with Age for Males [3]

In this Figure 2 and Figure 3, data point values are averages of samples. Each individual subject in the sample was assigned a whole number value from 0 to 3. So data points that fall between whole number values simply mean the sample average was not a whole number.

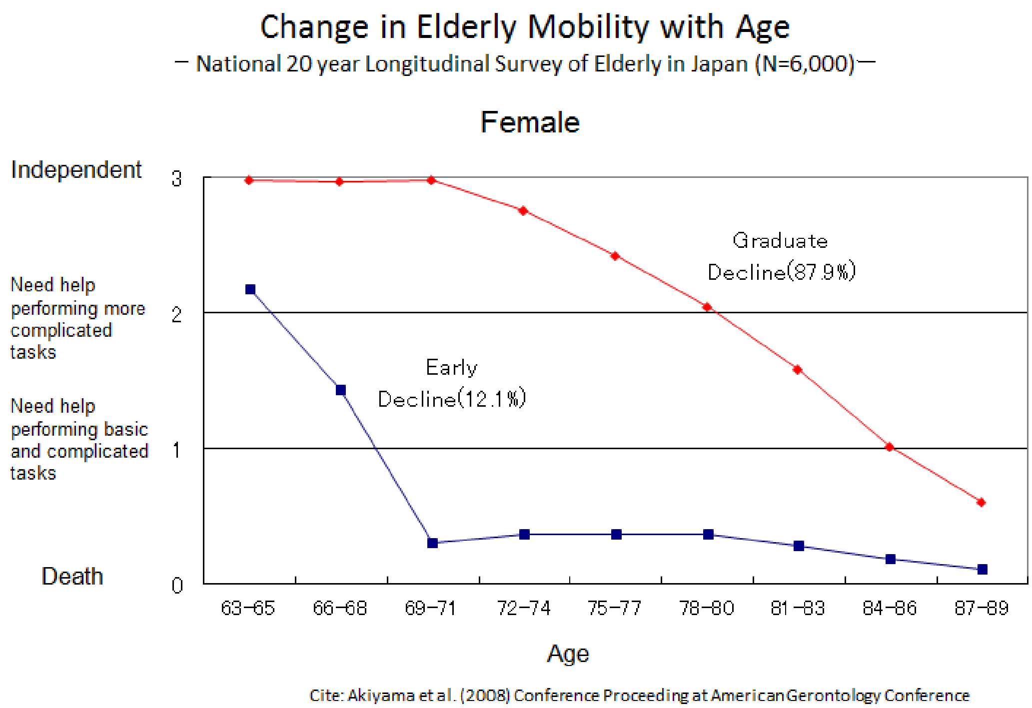


Figure 3: Change in Elderly Mobility with Age for Females [3]

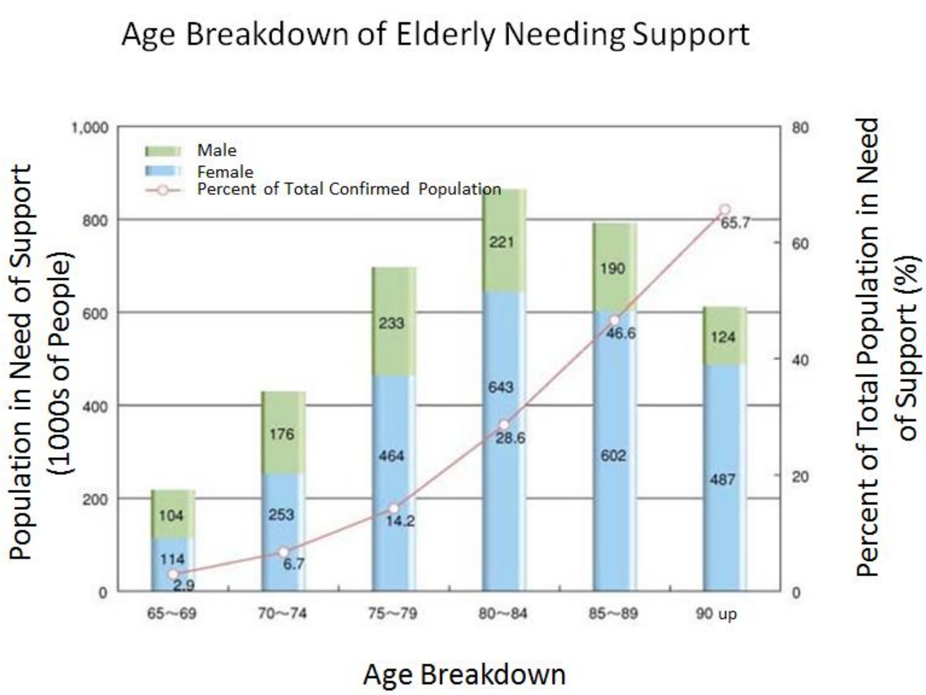


Figure 4: Age Breakdown of Elderly Needing Support [4]

Figure 4 shows that a higher percentage of female individuals live to higher age groups than male individuals. In general, as the elderly individual ages their need for support increases.

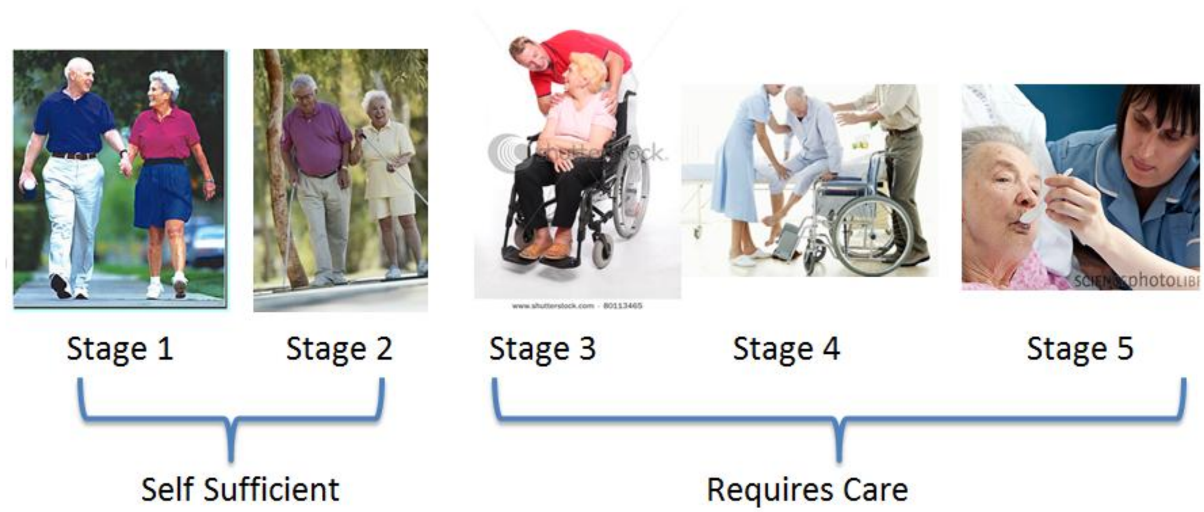


Figure 5: Stages for Required Elderly Care [4]

The Gender Information website of Japan developed a rating system that places the elderly in one of five stages of independence (Figure 5). Stage 1 is the most independent and Stage 5 is the least independent. People in Stage 1 and 2 are healthy enough to live on their own, but sometimes need help to stand up or walk. Those in Stages 3 through 5 need some level of assisted living. The percentage of elderly individuals in 3 to 5 range increases dramatically as their ages exceeds 90 (Figure 6). The intended users of the Global Clinic team's monitoring system will be from those in Stage 1 and 2 undergoing gradual decline. The system aims to help elderly individuals avoid reaching the later stages of decline.

	Man	Woman	Total
65～69	1%	1%	1%
70～74	2%	2%	2%
75～79	5%	5%	5%
80～84	8%	10%	10%
85～89	15%	22%	20%
over90	27%	45%	41%

Figure 6: The percentage of people in stages 3 to 5 in each age group's total population [4]

3.2. ADDITIONAL PROBLEMS FOR ELDERLY INDIVIDUALS

Isolated elderly individuals may feel lonely or concerned about their health. These worries can cause mental stress that leads to a physical health decline, and this decline gives the elderly individual more health issues to worry about. The new worries increase mental stress, resulting in the elderly user experiencing a cyclical decline due to isolation. Existing health monitoring systems aim to break the cycle of decline, giving the elderly user peace of mind by connecting them to distant family members.

Wandering is also a common problem for the elderly population. Even though the individual is capable of moving around, they may have Alzheimer's disease or dementia, which commonly leads to them becoming lost. Recent surveys of local police and neighborhood residents in Hachioji-Chuo, a city in Tokyo, indicated reports of 23,668 cases of elderly individuals becoming lost during the previous year. Of these, 548 cases were confirmed dead and 357 are still reported as missing [5]. One of the programs being implemented to help find lost elderly individuals is called the SOS Network. However, this program alone requested a budget of 985 million yen, which is about $12.7 million United States dollars (USD).

3.3. CURRENT SOLUTIONS

Japanese prefecture governments have funded the development of various types of medical monitoring devices and services for elderly individuals in several areas of rural Japan. During the HMC group's summer visit to Japan, the Global Clinic students attended three company site visits (Sections 3.3.1, 3.3.2, and 3.3.3). During the company site visits, company officials answered students' questions about the nature of the company's service and product for monitoring elderly individuals' health and safety. The Global Clinic team intended to build upon the shortcomings of previous devices and create a system that can be well-received by the target market.

3.3.1. Takao Calling Service Center

The Takao Center provides services that include regular telephone calls to elderly individuals living alone. The service asks the elderly individual about general health conditions and recent events in their lives. The elderly individual then has someone to confide in. When necessary, health representatives may also visit the elderly individual. This type of service was started about 22 years ago by the government and is now fairly popular in Japan. The visit to the Takao Center revealed that the Japanese people and government consider the mental condition of the elderly population to be of utmost importance in the growing problem of elderly individuals living alone.

3.3.2. ABS-1 Activity Monitor

During a site visit to IDUR's research center in Odaiba, Tokyo, the Global Clinic team met with representatives who presented the ABS-1 health monitoring device. The ABS-1 is a device designed at the request of Niigata Prefecture for the purpose of providing real time data about the current state of the elderly being monitored (Figure 7). The motivation for the device is to give family members peace of mind knowing their parent's activity level. It is able to measure ECG, heart rate, body position, and the number of steps that the user has taken. The device is then able to transmit the information through cell phone communication to an applet on a receiver cell phone.

An important thing to note about the ABS-1's development is that IDUR had selected the various sensors for incorporation into its final product simply based on the fact that it was possible to do so. IDUR's intent was to create a suite of sensors that could cover any needs that the users may have in terms of health information.

IDUR has already put part of the device through user testing and has shown that these tested parts work. Only the body position and pedometer functions were used during the user testing as the company did not advertise the ECG capabilities of the device. IDUR planned to market the device as a supplement to communication between family members rather than as a medical device due to these regulations. Recently, the Japanese government has withdrawn funding for the project, which has caused the ABS-1 development to stall.

Figure 7: ABS1 designed by IDUR [6]

3.3.3. SECOM Safety Call System

SECOM, a home security provider, created the Safety Call System with Kogakuin University. The Safety Call System is similar to a typical home security system in the United States but involves more 2-way communication between the service provider and the user. SECOM computers periodically call the users to verify that the users are well through a survey system. If the users respond that they need help, then the computers will queue the users for a company nurse to talk to (Figure 8). If the nurse determines that the user needs help, then he or she may suggest medical care.

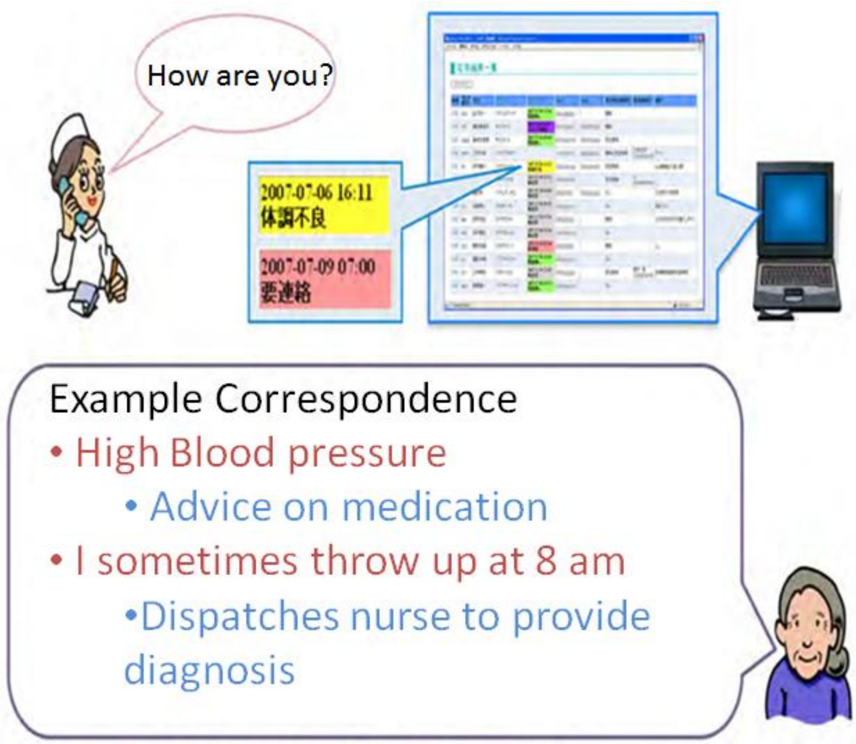


Figure 8: Illustration of the SECOM Telehealth System

The Global Clinic team learned through discussions with SECOM that the market is not large enough for systems that provide purely medical information (e.g. blood rate, body temperature, activity levels), because the cost-benefit ratio is not great enough to warrant significant investment into such new technology. Due to a decrease in the cost for sensors and microcontrollers, and a growing use of cell phone applications, the market may now be more receptive to devices that incorporate a wide variety of sensor technology. However, talking with the team from SECOM revealed that the technology that the Global Clinic team is trying to develop should focus on the idea of a "peace of mind system" as that is what SECOM attributes their system's success to.

3.3.4. HN-301

The most capable and least expensive device found by the team's research thus far is called the HN-301 manufactured by Parama-Tech [7]. The HN-301 measures blood pressure, body fat ratio, ECG, and quantity of body fat water and transmits these data wirelessly to a modem that transmits the data through the telephone line (Figure 9). The HN-301 weighs about 1 kg and costs about $2000 USD. There are currently few users because it is not being distributed to new users due to government budget cuts.

Figure 9: HN-301 Device by Parama-Tech [7]

3.3.5. VERITY

Verity is a remote medical monitoring device currently being developed by the startup iMonSys [8]. Verity is a chip built into a wristwatch that records heartbeat, body temperature, and body position data. It uses a gyroscope and an accelerometer to detect falls. It communicates with a cell phone wirelessly to call the children of the user and medical authorities. The most compelling feature of Verity is its advanced voice recognition system that allows for easy and natural interaction with an elderly user. A cartoon demonstration of the device on the Verity website shows the user slipping on the ground, at which point Verity asks the user if he is okay and alerts his daughter [8]. The user responds that he is okay, and Verity forwards this information to his daughter. However, it is implied that Verity would alert the police if the user failed to respond. This product is currently in the stage of acquiring funding.

3.3.6. Pros and Cons of Previous Solutions

Table 1 compares the positive and negative aspects of each solution discussed in Section 3.

Table 1: Positive and Negative Aspects of Previous Solutions

Existing Solution	Pros	Cons
Takao Calling Service Center	• Regular phone calls to check on elderly individual's health • Personal, interactive service • Employees actively communicate with elderly individuals	• Relies on elderly individual's self-assessment on health • Does not provide actual health vitals information
ABS Activity Monitor	• Measures ECG, heart rate, body position, and number of steps • Continuously transmits information to cell phones • Elderly individuals can carry device around with them	• Bulky • Requires elderly individuals to get used to wearing an additional accessory • Expensive to buy device and maintain communications service package
SECOM Safety Call System	• Periodic calls to check on elderly individual • Automatically queues elderly individuals for company employees to talk to based on computerized surveys	• Relies on elderly individual's self-assessment on health • Does not provide actual health vitals information
HN-301	• Measures blood pressure, body fat ratio, ECG, and quantity of body fat • Transmits data wireless through a phone line	• Weighs 1 kg and is not portable enough to carry everywhere • Expensive • Purely a health monitoring device • Requires elderly individuals to remember to use
Verity	• Measures heartbeat, body temperature, and body orientation • Automatically communicates with family members and medical authorities • Advanced voice recognition • Portable (wristwatch size)	• Still in prototype stage • Requires a cell phone to communicate wirelessly

3.3.7. Problems with the Existing Solutions

The existing solutions are all capable of providing the monitoring of elderly individuals. However, the current solutions such as the ABS1, the HN-301, and the Safety Call System necessitate a change in the lifestyle of the elderly user. The ABS1 requires users to remember to constantly wear a measurement device that otherwise has no other functions. For the HN-301, the user must pick up a device once a day solely for monitoring purposes. With the Safety Call System, the user must self-evaluate their condition and determine whether it is appropriate to contact health officials. These solutions expect the user to adapt to the device or service. In contrast, the Verity health monitoring watch demonstrates the idea of integrating sensors into an everyday item.

In addition, the least expensive of the preexisting devices costs several thousand United States dollars (USD). After the economic collapse of 2008, local governments have cut funding for the devices.

3.4. ECONOMIC EVALUATION OF CURRENT JAPANESE TELEHEALTH SYSTEMS

Monitoring devices have been manufactured, distributed and linked to nurses and doctors in hospitals to form independent telehealth systems in several different prefectures. As of 2003, a total 10,000 devices had been distributed. However, it is uncertain whether the devices mentioned in previous sections were part of the distributed devices. The Global Clinic team researched into particularly why it was difficult for several prefectures to continue funding for remote health monitoring devices.

One of the economic evaluation of the telehealth system was conducted by Masatsugu Tsuji, Wataru Suzuki, and Fumio Taoka of Osaka University and Kyoto University on systems implemented in Kamaishi City, Iwate Prefecture; Nishiaizu Town and Katsurao Village, Fukushima Prefecture; and Sangawa Town, Kagawa Prefecture [9]. The evaluation quantifies the usefulness of the system by calculating an Average Willingness to Pay (WTP) across a survey of users which asks the user how much money they would be willing to pay to continue to use the device for a month. The Average WTP is compared to the total costs associated with the telehealth system to see if the system yields more benefits than costs.

The system of Iwate Prefecture had a Benefit / Cost ratio of 1.07, but all other prefectures had a ratio below 1. This result is primarily due to the initial expense of purchasing the device. When this expense is reduced, the ratio for all prefectures is well above 1. Some of the purchases were subsidized by the national government, so the prefectures did not have to fully bear these costs. These results suggest that the high cost of the existing monitoring medical devices is the main barrier to wider adoption through Japan. The report did not investigate the ease of use for the distributed devices. As a result of this survey, the Global Clinic team realized the need for developed devices to be affordable enough for either direct purchase by the general public or subsidies from the national government.

3.5. HABITS OF ELDERLY INDIVIDUALS IN JAPAN

In order to better understand how to design monitoring devices that will fit into the Japanese elderly individual's life style, the Kogakuin students conducted surveys to determine the habits of elderly individuals.

3.5.1. First Kogakuin Survey

During October 2011, the KU group distributed a survey to 20 elderly individuals. The term "elderly individual" is defined as those that are age 65 and over. The results of survey were as follows:

1) How do you react to technology?

   Most respondents expressed a negative reaction toward more modern technologies such as cell phones and computers.

2) What do you use on a regular basis or what do you normally do when you are alone?

   Watching television (all respondents indicated this), listening to the radio (1), taking a walk (3), sleeping (1), singing karaoke (1), receiving a massage (1), and shopping (1)

3) What objects do you use every day?

   Household appliances (all respondents indicated this), cell phones (4), computers (2), pedometers (2)

4) What sensors do you believe are most important for monitoring elderly individuals?

   Heart rate (7), blood pressure (7), temperature (unknown in this survey if this is ambient or body temperature) (6), GPS (4)

From this survey, the Global Clinic team confirmed the hypothesis that elderly individuals are not comfortable around new forms of technology. The survey also shows that almost all elderly individuals in Japan are likely to use familiar technologies such as television and household appliances. By designing monitoring devices that integrate well with common elderly behavior, the Global Clinic team may be able to create a system that is compatible with elderly individuals' existing lifestyles.

3.5.2. Second Kogakuin Survey

Following the initial survey, the KU students performed another survey during November 2011 by asking 17 middle aged adults, 45 – 65 years of age, about general information regarding the situation of their parents. These individuals then forwarded the surveys to their own parents to complete. Thus, the results of the survey contain information from both family members and their elderly parents.

Questions asked of young family members regarding the living situation of their elderly parents included:

1) Do you live with your parents?

   88% of family members indicated that they do not live with their parents.

2) How far away do your parents live?

   35% indicated they lived within 30 minutes by car, 0% lived within 1 hour, 41% within 3 hours, and 35% over 3 hours.

3) What type of sensor data would be useful to know from your parents?

The question was conducted in two parts. First, participants were asked to give a score of three for the most important data, two for the second most important data, and one for the third most important data they would like to know about their parents. The least important sensors would thus not receive any points. The second part of the survey asked parents what information they would like to send to family members (Figure 10). The three sensors with the highest combined scores were for body temperature (56), blood pressure (51), and ambient temperature (33).

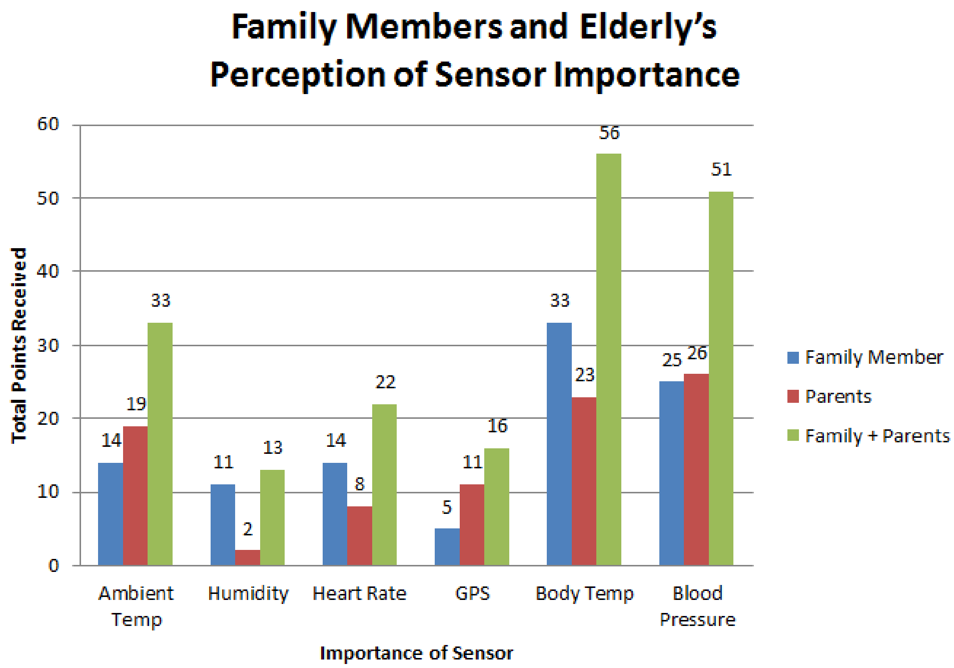


Figure 10: Family Members and Elderly Individual's Perception of Sensor Importance

Using the results of this survey, the Global Clinic team assigned a time and schedule for sensor integration. A blood pressure sensor was not implemented because the team believed that such a sensor is overly intrusive, which did not fit the design criteria.

3.5.3. Conclusions on Health and Surroundings Monitoring

Through the surveys performed by the KU group, the Global Clinic team concluded that the most relevant aspects of an elderly individual to monitor were blood pressure, body temperature, and heart rate. These health vitals provide a sense of how healthy the monitored individual is and also if he or she is in immediate need of assistance. In addition, it would be useful to measure ambient temperature and humidity. These data are helpful in determining whether the elderly individual is somewhere that may pose a risk to his or her health.

Because of the danger that the elderly individual may become lost, positioning coordinates is a helpful piece of information to acquire from systems devised by the Global Clinic team. Section 6 describes the hardware that the team chose to monitor specific points of an elderly individual's health and surroundings.

4. PROJECT DEFINITION

4.1. INTENDED USER AND MARKET

The Global Clinic team intends for the device users to be elderly individuals who are healthy enough to live alone but are at risk for health issues or accidents. However, the device purchasers are intended to be concerned family members, and the marketing efforts will reflect this. Younger family members are targeted because they are more likely to adopt a new device while the elderly individuals themselves are less likely to be aware of recent innovations.

4.2. PROJECT STATEMENT

To address the problem of a growing and increasingly isolated Japanese elderly population, the 2011-2012 Global Clinic team will research and develop a health vitals monitoring system for Japanese elderly. The system will focus on being compatible with the Japanese culture, lifestyle, and infrastructure as well as being non-intrusive in the elderly individuals' lives and habits. The project's end goal is to develop a device that performs the same health vital monitoring capabilities as existing solutions for providing peace of mind, but offers better form factors that integrate seamlessly into an elderly individual's current lifestyle.

4.3. DIVISION INTO MODULES

The flow of information in the system starts at the elderly individual. This information is then stored on the internet to be accessed and interpreted by a third party who wants to monitor the elderly individual (Figure 11).

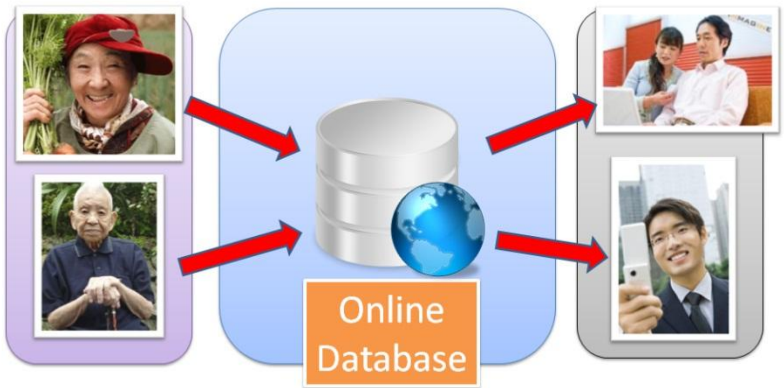


Figure 11: Information Flow of Intended System [10] [11]

The technology that accomplishes this flow of information can be broken down into components that interface with each other (Figure 12):

- Measurement device user interface – the form factor and anything that the user will interact with on the physical device
- Sensor platform – various sensors and the data acquisition module that record the health and activity levels of the elderly individual
- Power source module - the energy required to run the system
- Transmission module – the segment of the project that transmits the data between the user's device and the third party interface
- Data storage module – the online database that stores records of third party users and sensor data
- Third party interface – what the family member or a third party interacts with when they want to view the data stream being sent from the sensors

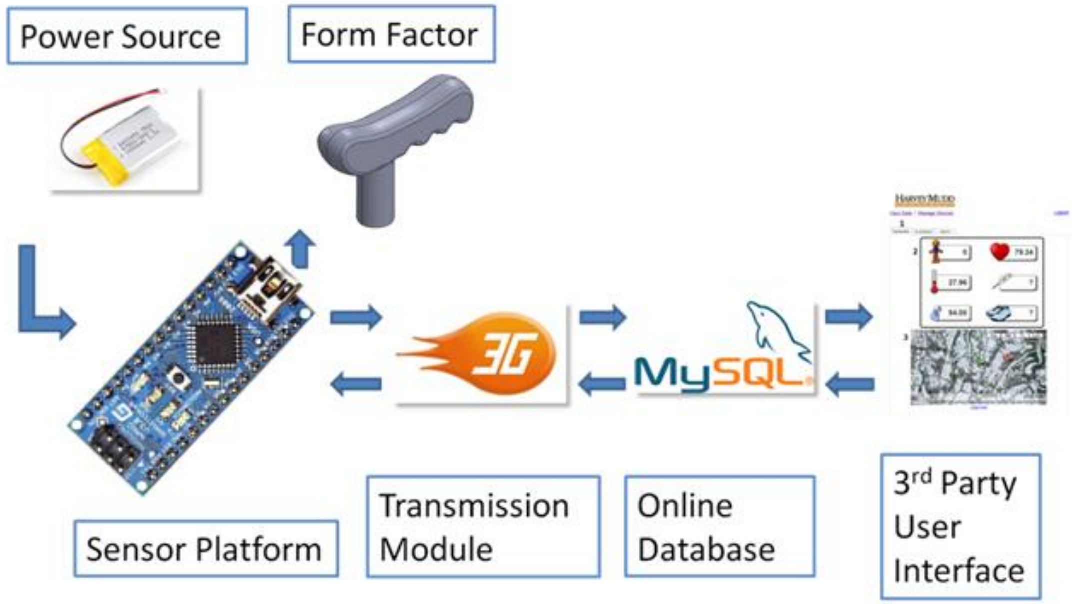


Figure 12: Technology Architecture with Data Transmission of Intended System

4.3.1. Functions

Functions are actions, processes, or tasks that the designed system has to perform. The Global Clinic team has divided the functions for their intended system into different modules.

Measurement device user interface:

- Minimize the effort the user undergoes in the process of measuring vitals
- Communicate (e.g. through graphics or audio) sensor data information to users
- Receive user inputs

Sensor platform:

- Gather information about the user (which may include body temperature, humidity, ambient temperature, heart rate, user activity, location, and possible emergencies)
- Store digitally formatted data

Power source module:

- Store energy
- Supply the necessary voltage, amperage, and wattage to the system components

Transmission module:

- Transfer data between third party interface and sensor platform

Data storage module:

- Store third party user account information
- Store data gathered by various measurement devices

Third party interface:

- Require user to log into an account to view measurement data
- Accesses database content
- Display information in a Graphical User Interface (GUI)

4.4. DELIVERABLES

At the end of the fall semester, the team delivered the following technical items:

1. Research on the elderly population and market in Japan
2. A working prototype of the designed system
3. Team Charter
4. Joint HMC-KU summer presentations

5. Three internal design review presentations
6. Team leader transition report
7. Technical Memorandum

At the end of the spring semester, the team delivered the following technical items and presentations:

1. Printed and populated functional circuit board (PCB) design
2. 10 sets of electronics for installation into form factors
3. Spring semester presentation at HMC
4. Three form factor prototypes
    a. 1 functional prototype
    b. 2 conceptual designs and initial prototypes
5. Projects Day presentation and poster at HMC
6. Final Report

5. FORM FACTORS

For a designed system to succeed in the Japanese market, it must be suitable for an elderly individual's lifestyle. Possibilities include objects that an elderly person already uses (such as a TV remote) or a device that can be used without requiring much attention (such as a bracelet). Objectives and constraints were used to select the form factors to pursue from the list of concepts, which is mentioned in Section 5.1 but listed in more detail in Section 12.

5.1. FORM FACTOR CONCEPTS

To address the various lifestyles of elderly users, the Global Clinic team decided to design a suite of form factors. A user can choose one or more of the devices that fit his or her lifestyle and habit. To provide the user with this flexibility, the Global Clinic team developed a wide variety of form factor concepts, some of which are shown in Figure 13. These concepts included adaptations of common items such as door knobs, chairs, and television sets. The Global Clinic team broke the form factor concepts into categories such as items used outside, items used at home, and items worn. Form factor concepts from each of these categories were further developed into a walking aid, stationary handheld device, and accessory clip, respectively. Further information about some of the other concepts are detailed in Section 12.

Figure 13: Sample Collection of Various Form Factor Concepts [12]

5.2. OBJECTIVES

Objectives are the characteristics of the system that are preferred in a final design. In order of importance, the objectives of the form factors being designed by the Global Clinic team are:

1. Easily integrated into existing habits
2. Frequently used
3. Familiar to the elderly user
4. Provides a sense of security
5. Emotionally connects elderly users and their families well
6. Capable of including multiple sensors
7. Durable

As a result of surveys described in Section 3.5, the Global Clinic team realized that elderly individuals would react best to systems similar to items that they are familiar with. This lent itself to the formation of the first three objectives from the objectives list. A design that fits these objectives well will be more readily accepted by elderly individuals.

The fourth and fifth objectives relate to the peace of mind concept that the Global Clinic team is trying to offer elderly individuals. This health monitoring system design intends to connect elderly users with distant family members and give them the reassurance that someone is watching over them.

The sixth objective addresses the desire to collect a wide variety of data. Depending on the type and functionality of a form factor design alternative, some sensor data may not be obtainable. The Global Clinic team is working under the assumption that an elderly user may be able to choose from a suite of form factors, each with different measurement capabilities, so this objective is not as important as ease of integration into an elderly user's lifestyle. However, a form factor capable of measuring a wider variety of data is preferable to one with a narrower range.

Because most of the objectives listed are not quantifiable, extensive user testing will be integral to the success of any form factors developed by the Global Clinic team. After the user testing phase begins, the team will improve ergonomics for the different form factors, develop concepts for new form factors based on user feedback, and integrate more functionality into the existing system.

5.3. FORM FACTOR 1: WALKING AID

A cane with embedded sensors can act as a physical support while monitoring elderly users. A cane form factor would directly target the market that already uses and relies on canes for daily activities. For these users, the cane form factor would be easily adapted into their lifestyle. Because the use of a cane is typically outdoors, the form factor would be effective at monitoring the activity level of users when they leave the house but not as much when they are indoors. The cane form factor can be extended to related form factors such as walking sticks or walkers.

Figure 14: Final Prototype of Cane Handle

The Global Clinic team initially designed the cane form factor so the various sensor and processor components would be located within the handle of the cane. Rapid prototyping was done using a 3D printer to make several revisions on the design of a walking cane handle to fit the sensor platform. After four iterations, the team determined that if all the sensor and processor components were placed in the handle, the handle would be too large and ergonomically poor. Therefore, final version uses the space in the shaft of the cane (Figure 14). This modification allows the handle to be smaller and more ergonomic.

5.4. FORM FACTOR 2: STATIONARY HANDHELD DEVICE

Surveys conducted by the Global Clinic team show that most Japanese elderly individuals often watch television at home. By incorporating sensors into a TV remote, the system can acquire health information and send it wirelessly to a data hub somewhere in the house while the elderly individual is using the remote. This method would gather heart rate and body temperature through sensors embedded in the remote that contact the skin when the remote is grasped (Figure 15). However, it would not be useful for determining body activity or location, since the remote will most likely be kept in one location.

Presently, the remote is only designed to control a television and passively collect information. An opportunity for extending its functionality is to create a console that plugs into the TV and is controlled by the remote. When the console is turned on, it allows the user to view photos and videos of their family members on the TV. This functionality will encourage the user to interact with the remote more often and thus gather more data.

Figure 15: Latest Remote Iteration with Intended Hold and Some Sensor Placement

The remote was developed by the KU group in Japan and built from the Nintendo Wii remote because of the success of the Wii remote and the fact that its thickness allows greater flexibility in embedding sensors. The Global Clinic team plans to build on the controller's ergonomic features during the summer of 2012 to develop a health sensor-based remote control (Figure 15).

5.5. FORM FACTOR 3: ACCESSORY CLIP

A high volume of information can be obtained by embedding sensors in items that the user would wear. However, a user may prefer a specific style and not want to replace it with a different version despite added monitoring functionality. For example, a user may prefer a specific brand of shoes and forcing them to replace those shoes would constitute a change in lifestyle.

The main motivation of the accessory clip design is that it does not force the user to replace an item with one designed for monitoring. Instead, the user would attach the accessory clip to one of their possessions, such as a shoe, hat, or purse. The device facilitates measurement by being located on something that the user frequently wears or carries and thus does not require additional effort by the elderly individual to use. The elderly user can be constantly monitored when he or she wears the object.

The accessory clip is meant to be adapted to the object it will fit on to, as opposed to a one-size-fits-all type of clip. For the initial form factor concept, the Global Clinic team focused on creating a clip adapted for the shoe (Figure 16). Currently this design is in the concept phase.

Figure 16: Mockup of accessory clip form factor

Because the shoe clip will not contact the user's skin, sensors that require skin contact, such as the heart rate or body temperature sensors, will not be implemented. However, other adaptations of the accessory clip, such as a hat clip, may be able to contact the skin. The limitation of not implementing sensors that require skin contact is specific to the shoe clip.

6. ELECTRONIC HARDWARE IMPLEMENTATION

The electronic hardware for the system refers to the sensors, Bluetooth modules, PCBs, and microchips necessary for the collection and wireless transmission of data. An important goal of the project was to develop a hardware platform that could be easily integrated into a variety of form factors.

Key components of the hardware are the sensors, which allow the measurement of the data. In Section 3.5.3, the Global Clinic team concluded on what health vitals and environmental conditions to monitor. These included heart rate, blood pressure, body orientation, body temperature, ambient temperature, relative humidity, and location. However, blood pressure was determined inappropriate to be included into the sensor platform. Measuring blood pressure requires the use of a cuff, which is overly-intrusive and would disrupt an elderly individual’s lifestyle. After researching available equipment and sensors for measuring each of the desired vitals and conditions mentioned above, the Global Clinic team selected the sensors listed in Table 2 for the sensor platform.

Table 2: Quantity Measured By Each Sensor

Sensor	**Part Name**	**Quantity Measured**
Inertial Measurement Unit (IMU)	ITS3200/ADXL345 IMU Digital Combo Board	Body activity and orientation
Pulse Photosensor	Photo Reflector IR LED and IR Phototransistor	Heart rate
IR Temperature Sensor	MLX90614 IF Thermometer	Body temperature
GPS	EM-406a GPS	Physical location
Humidity Sensor	HIH4030	External relative humidity
Temperature Sensor	LM35	External temperature

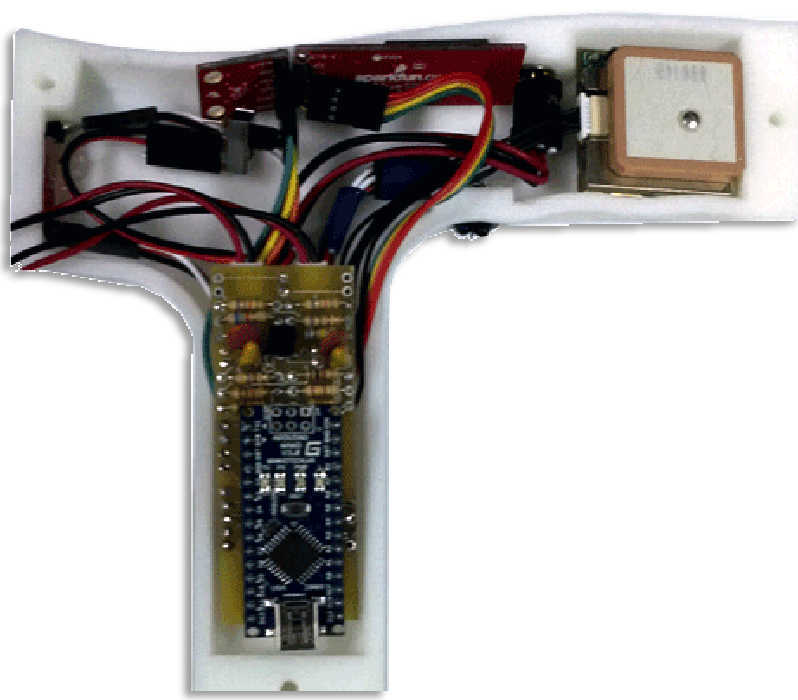


Figure 17: Circuit Board Setup of Current Sensors and Hardware

As seen in Figure 17, various sensors are connected to a central microprocessor using breakout wires, which the HMC group has selected to be the Arduino platform, via an additional breakout PCB that functioned as a signal router between the sensors and the microprocessor. A PCB was designed to fit as a shield on top of the Arduino Nano in order for the sensors to be connected to the required pins on the Nano.

All hardware components used were open source, meaning that the details and schematics of their designs were open to the public to use. The open source platform was beneficial for the team due to the availability of online resources such as tutorials, sample source code, sample PCB design files, etc. The prototype PCB design was also a compilation of the circuit design obtained from the Sparkfun distribution site, which was open source as well.

6.1. INERTIAL MEASUREMENT UNIT (IMU)

The IMU sensor (Figure 18) is a dual digital accelerometer and gyroscope combination sensor. It uses the ITG3200 for gyroscope purposes and the ADXL345 for accelerometer purposes. This sensor was chosen for its ease of implementation as well as the existence of prior code for its operations.

Figure 18: IMU Digital Combo Board [13]

The 3-axis ITG3200 gyroscope utilizes microelectromechanical systems (MEMS) technology through the use of three independent vibratory MEMS gyroscopes, which each detect a different axis of rotation. Physical movements will disrupt the normal oscillation of these independent gyroscopes. The changes are detected, digitized, and outputted to the microprocessor [14].

The 3-axis gyroscope is able to detect rate of rotation in the x, y, and z directions. This sensor can be coupled with the accelerometer sensor to determine the device user's physical orientation and movements. This sensor combination is useful mostly for determining whether the user is in motion, particularly motions involving rotation, such as falling to the side.

The 3-axis ADXL345 accelerometer detects the acceleration of gravity as well as any dynamic acceleration. The mechanical sensor consists of independent plates, differential capacitors, and a moving mass. Sets of plates are either fixed or attached to a moving mass by a differential capacitor. Acceleration causes the system to unbalance, resulting in the differential capacitor to output a voltage signal proportional to the acceleration. The phase and magnitude of this signal can be processed to determine the direction and magnitude of acceleration. The 3-axis accelerometer uses these methods to detect acceleration in the x, y, and z axes [15].

The accelerometer can be coupled with the gyroscope in order to more accurately determine the user's current position and motion. This combination of sensors can be used for determining if the user is currently in an upright position or if he or she is lying down. The sensor combination can also be used to detect any sudden changes such as falls or impacts.

Walking motion of the user can be modeled as a low frequency (~1 to 2 Hz) accelerations in the z direction (up and down) which occur whenever the user's foot impacts the ground. The ADXL345 can be configured to detect such impacts and send a signal to the microcontroller when they occur. If the acceleration signal exceeds a given threshold but then drops below that threshold within a given window of time, then the accelerometer registers that as an impulse event. By configuring this threshold and the time window, the accelerometer can register most walking motions by the user as a series of impulses.

6.2. PULSE PHOTOSENSOR

The pulse photosensor (circuitry shown in Figure 19) uses light reflection intensities to measure the heart rate of the user. The circuitry uses an IR LED and an IR photodiode sensor (Figure 20). The LED emits light onto a section of the user's skin, normally at the fingertips, while the photodiode sensor detects the intensity of the light that is reflected off the skin. With each heartbeat, a higher volume of blood pushes through the capillaries. The higher volume of blood causes a difference in the light intensity being reflected off the surface of the skin. The software detects the resulting peaks in the light intensity and calculates an average heart rate. This circuit employs a basic high pass and gain filter to clean the signal.

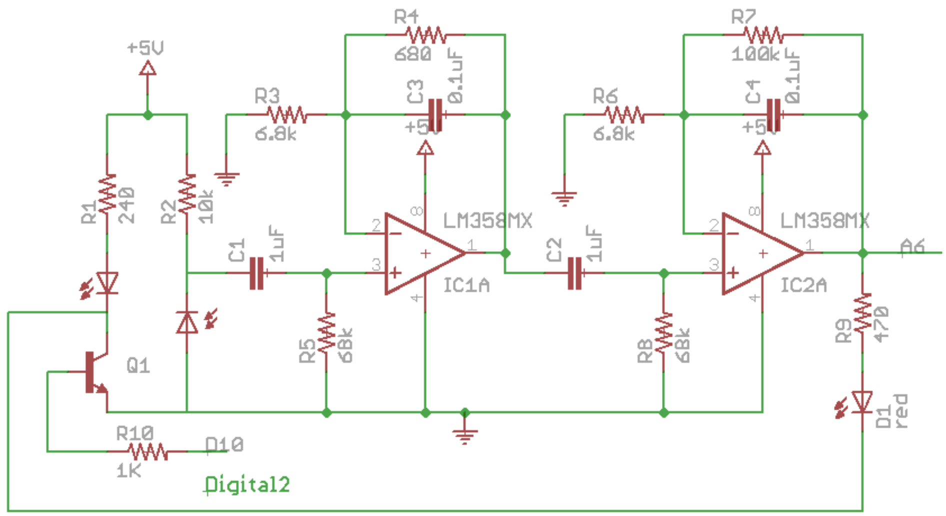


Figure 19: Pulse Photosensor Circuitry

For the final prototype, instead of using standalone IR LED's and IR photodiodes, the team designed the pulse sensor using the Vishay Photoreflector package.

Figure 20: Vishay Photo Reflector IR LED and IR Phototransistor [16]

6.3. AMBIENT TEMPERATURE SENSOR

Ambient temperature was an important measurement for the Global Clinic team's system to obtain because of the high temperatures that are common in Japan during the summer. Elderly individuals are more susceptible to the high temperatures. The system monitors the surrounding temperature and sends an alert to both the user and the family members in case the temperature rises to a dangerous level.

The LM35 sensor (Figure 21) is an ideal way of measuring ambient temperature since it is designed to output voltage linearly proportional to the measured Celsius temperature. As a result, very few calculations or conversions are necessary to convert the voltage levels into degrees Celsius. The LM35 is able to also produce a higher voltage output than thermocouples and thus does not require any additional circuitry for voltage amplification [17]. It also does not require any external calibration in order to maintain a consistent accuracy.

Figure 21: LM35 Temperature Sensor [17]

An LM35 runs on less than 60 μA at between 4 to 30 volts. It can detect -55°C to +150°C temperatures, though additional circuitry is necessary for the sensor to detect negative temperatures. It also experiences a low level of self-heating (about 0.08°C), which is a benefit for temperature sensors since their temperature readings would be offset otherwise. The sensor also has a precision of about 0.5°C, which is relatively good in comparison to more expensive alternatives.

Because the LM35 measures external temperature, it must be exposed to the outside environment, usually through a hole on the surface of the form factor. This is a design constraint since it limits where the temperature sensor can be placed. In the current sensor platform, the LM35 does not have the accompanying circuitry that enables it to detect temperatures in the negative range. Thus, the current iteration may not be useful during the winter season in Japan, when temperatures occasionally fall to the negative range. However, simple changes may be done to enable this addition in the future. The Global Clinic team noted that the temperature sensor was sometimes unstable and produced measurements that had to be run through a moving average filter.

6.4. HUMIDITY SENSOR

Humidity is often a concern for elderly individuals in Japan due to the high humidity levels that Japan experiences over the summer. High humidity levels increase the risk of health problems and accidents occurring among the elderly population in Japan.

Figure 22: HIH4030 Humidity Sensor [13]

The specific humidity sensor chosen was the HIH4030 digital humidity sensor (Figure 22). This sensor has a thermoset polymer capacitive sensing element that is able to detect how much moisture has been absorbed into the element, which changes its capacitance [18]. It outputs a voltage level that is linearly proportional to the measured percent relative humidity.

The HIH4030 operates at a 4 to 5.8 volt DC power supply and draws about 200 μA of current. According to datasheets, it is a low power design with enhanced accuracy and fast response time. The sensor itself is relatively small compared to the other components of the Global Clinic's system and was not a difficult sensor to incorporate into the design of the form factor.

Because the humidity sensor measures external humidity, the detection slot on the sensor must be in contact with the surrounding environment through a hole in the form factor walls. This limits the places that the humidity sensor can be placed. The percent relative humidity reading was also dependent on temperature, which increased the calculation complexity. The Global Clinic team noted that the temperature did not have much of an effect on the relative humidity readings close to room temperature.

6.5. SKIN TEMPERATURE SENSOR

Temperature on the surface of the user's skin is important because significant and sustained increases in skin temperature, even on the appendages, often correspond to medical problems such as a fever.

Figure 23: MLX90614 Infrared Thermometer [19]

The MLX90614 Infrared Thermometer was chosen because it does not require skin contact to obtain an accurate reading. It is mounted in a recessed hole so that if the user's skin is over the hole, the sensor is a few millimeters away. The digital sensor communicates with the board through the I^2C protocol and outputs an accurate value in the form of an 8-bit word [19]. It is able to operate at temperature between -20 and 120 °C with a resolution of 0.14 °C. This sensor is factory calibrated and has low enough noise that its output does not need additional filtering. It can also be used as an ambient temperature sensor but was not used twice because it consumes more power than the existing analog ambient temperature sensor. The version currently in use runs on 5 volts and 1 to 2 mAh.

6.6. GLOBAL POSITIONING SYSTEM RECEIVER

GPS is currently the prevalent way of finding location coordinates. It utilizes a satellite system in space to triangulate the approximate location of the GPS receiver, as long as there is a line-of-sight to four or more satellites. The prototype system designed by the Global

Clinic teams uses the EM-406a GPS (Figure 24) in order to determine where the user currently is and possibly provide a history of where the user has been.

Figure 24: EM-406a GPS Module [13]

If the elderly individual becomes lost and the GPS chip is still able to transmit location coordinates to a central database, most likely through cell phone communication towers, then it is possible for family members or police to be able to easily locate the lost individual. The primary issue with the GPS is that it is only able to detect its location when it is outside. When an elderly individual goes indoors, the system will transmit the last known outdoor position to the database. The GPS can also play a role in determining if the elderly individual is where he or she is expected to be. The Global Clinic team hopes that eventually, if the GPS detects that the individual had been at a location where he or she should not have been, a notification could be sent out by the processor to family members. This can be used primarily to see if wandering has taken place.

The GPS chip has the potential to be a major energy consumer in the designed system. In order to resolve this issue, the GPS chip may be only turned on at certain fixed intervals to minimize energy costs.

6.7. BLUETOOTH MODEM

A Bluetooth connection allows for data to be transmitted wirelessly over a short range between a transmitter and a receiver. The specific Bluetooth module that the Global Clinic chose was the BlueSMiRF Silver (Figure 25) due to a balance of cost and transmission range. A Bluetooth module and its corresponding receiver dongle create a serial connection wirelessly with a transmission frequency of 2.4 GHz. The BlueSMiRF operates on 115200 Baud with a transmission range limited to about 10-20 feet [20]. This distance can be used for separating the sensor platform from the transmission unit. This helps reduce the size of the device the user will carry, allowing the device to be much more portable.

Figure 25: Bluetooth Modem – BlueSMiRF Silver [13]

Similar to the GPS chip, the Bluetooth modem can be a major energy consumer and could drain the battery quickly. To resolve this issue, the Bluetooth modem does not have to be transmitting information continuously. Instead, the modem can periodically send bursts of data to the central database and then shut off for the other periods of time.

6.8. 3G MODEM (SAMSUNG DART)

A 3G connection allows data to be uploaded to the internet as long as the device is in range of a 3G-equipped cell phone tower. Because 3G service is available in nearly all of Japan, the Global Clinic team decided to use it as the primary medium for wireless transmission. For a stationary device, 3G is expected to transmit at a rate of 125 kilobytes per second.

In contrast to Bluetooth modems, 3G modems and operating instructions are not widely available. Therefore, the current system uses the 3G modem present in the Samsung Dart Android smartphone (Figure 26). The Global Clinic team is only using the smartphone as a temporary means of accessing the 3G network and does not intend for a cell phone to be required for future prototypes. Eventually, the team hopes that 3G modules will become widely available for the design usages so that the newer prototypes will be able to replace the cell phone and Bluetooth modules with the 3G modules.

For the current prototypes, the system interfaces with the 3G modem by sending the data first to the Bluetooth module via the processor. The Bluetooth module can then transmit the data wirelessly to a smartphone that has 3G and Bluetooth capabilities. The smartphone runs a Java program that processes the incoming data, and then uses its internal 3G modem to upload the processed data to the database. The addition of the smartphone to the system is clearly inconvenient, because the user must carry it with them. As mentioned before, in the future, it is expected that increased availability of information will allow the direct integration of a 3G modem to the microcontroller unit, which will remove the need for the Bluetooth modem.

Figure 26: Samsung Dart [21]

6.9. MICROCONTROLLER UNIT

A microcontroller unit (MCU) is used to acquire data from each sensor. Each analog sensor is read using the MCU's built-in A/D converter. Each digital sensor uses the I^2C serial protocol. Finally, the Bluetooth and GPS both use serial communication. The MCU needed to be able to handle all the sensors implemented.

6.9.1. ATmega2560

The Arduino Mega (Figure 27) was chosen as the board to be used for the first semester prototype because the Arduino ATmega2560 chip has a high number of analog inputs and multiple serial ports. This means that a series of sensors can be attached to the analog ports for data collection while the serial ports can be dedicated to Bluetooth transmission, GPS data communication, and other future devices requiring serial communication, such as cell phone transmission modules. However, in order to incorporate the ATmega2560 microcontroller unit, a custom microprocessor board has to be designed due to the large size of the Arduino Mega board. The difficulty in designing a working microprocessor board pushed the team to look for alternative solutions from commercial off-shelf products.

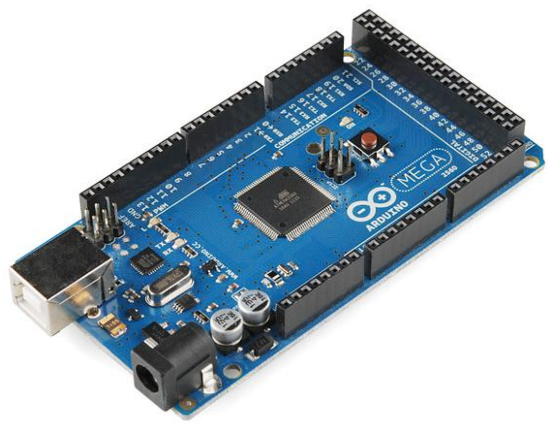


Figure 27: Arduino Mega (ATmega 2560) [13]

6.9.2. Arduino Nano (ATmega328)

The team decided to use the ATmega2560 chip because it had four hardware serial ports. In comparison, all other ATmega chips have one hardware serial port, but the sensor platform design requires two, for the Bluetooth and the GPS module. However, the team later realized that the Arduino libraries include the capability to use any two digital pins to simulate a serial port through software. Therefore, the other chips effectively had many more serial ports than previously thought. Because of this flexibility, the Global Clinic was able to look into other commercially available models of the Arduino microprocessors. After researching various Arduino options, the Global Clinic team chose the Arduino Nano as the sensor platform's processor unit (Figure 28).

Figure 28: Arduino Nano Microprocessor [22]

6.10.POWER SUPPLY

The Global Clinic team researched power supply options by creating a matrix to calculate the estimated battery life as a function of the sensors implemented, the modes that they operated in, and the energy capacity of the battery used. This involved analyzing the operating conditions of the various sensors used, particularly the current draw for each. Using the current draw, the expected percentage of the time each sensor is active, and the energy capacity of available batteries, an approximate battery life per recharge was calculated on the matrix. These calculations revealed that the Arduino Nano, GPS, and Bluetooth modules were the largest energy drain on the system. However, the GPS and Bluetooth modules are able to be switched to a sleep or idle mode that significantly reduces their current draw. The degree to which the system utilizes the sleep or idle modes depends on how the system's software is written. At this point, power optimization codes have not been written into the software.

Based on the energy consumption analysis (Section 14), the Global Clinic team determined that many of the readily available portable energy sources, such as AAA or coin cell batteries, were far too weak to be able to handle the energy usage. In addition, the team also wanted to incorporate a rechargeable battery so that the system can be recharged in between uses. Primarily due to these two factors, the team selected a 1000 mAh lithium polymer (Li-Po) battery (Figure 29) to minimize both size and cost. Lithium polymer batteries have become popular recently and are currently used for many portable devices, such as cell phones and portable gaming devices. Li-Po is also more energy dense than previous batteries based on alkaline or nickel, and as a result, is able to pack more energy in a smaller volume. Because a minimum of 5 volts is required to run the designed system and these Li-Po battery packs typically have 3.7 volts, the Global Clinic team put two battery packs in series with each other. This allows for a total of 7.4 volts to be provided to the device and allows the system to keep running longer even as the batteries' voltage drops slightly due to energy usage.

Figure 29: Lithium Polymer Ion Battery (1000 mAh) [13]

In order for the overall system to be easy to use for an elderly individual, a recharger unit should be incorporated into the device. This has not been implemented. There are breakout pins that enable an external recharger circuit to be connected to the batteries. This allows for the batteries to be recharged without being removed from the form factor. Ideally, a built-in charger circuit will work in conjunction with specially designed external power stand that will convert power outlet voltage to the appropriate 5 volts needed for the built-in charger circuit.

One of the issues that the Global Clinic team has not had time to fully resolve was the problem of recharging the two battery packs while they are connected to the system. Because the two battery packs are attached in series, they each have a different ground voltage level. This issue makes recharging difficult since the higher ground voltage level conflicts with most existing recharger circuits and causes a current overload. Recharging the two battery packs at the same time in series is also dangerous since it increases the risk of uneven charging and that one battery may be overcharged, leading to possible fire hazards. The temporary solution devised by the Global Clinic team is to cut the connection between the two batteries temporarily during the recharging period using a switch. The switch would essentially turn the entire system off during the recharging period.

The Global Clinic team has verified the accuracy of the preliminary power analysis and the efficiency of the recharger circuit intended for the prototype system. The initial analysis suggests that two 1000 mAh Li-Po battery packs in series allows the full system to be run for a little under 11 hours. This value is subject to variations primarily in the amount of time that the GPS, Bluetooth and Arduino Nano are in an active state. Because power optimization code has yet to be written, a more accurate calculation is difficult to perform.

6.11. PRINTED CIRCUIT BOARD (PCB)

To facilitate future miniaturization of the hardware and form factor shape adaptation, the HMC group designed and ordered professionally printed circuit boards. The initial intention for these PCB's was to have them serve as the entire sensor platform and have the sensors built into the board. However, this proved to be too difficult for the Global Clinic team's abilities. As a result, the team decided to print PCB's as an interface between the Arduino Nano and the various sensors, linking the input/output pins of the sensors to the correct ports on the Arduino Nano.

The bill of materials (BOM) for the final PCB design is in Section 12, the schematics are in Section 14, and PCB layouts are in Section 16. All designs were made using Eagle CAD because the project used mainly open source electronic hardware from Sparkfun, which provides schematics to most of the parts they sell. Arduino also provides their schematic source file in Eagle CAD format.

The design of the PCB's included two phases:

1. 2 layer Microprocessor design using ATMega2560 MCU chip
2. Breakout board for the Arduino Nano V.3.0

Although the initial iteration of the PCB involved designing all of the sensors and processor onto one board, the Global Clinic team was unable to make the board work properly. After several more iterations of designing a microprocessor board, which resulted in a product that is less reliable than off the shelf products, the team opted for the use of an off-the-shelf product. A breakout board was designed for the Arduino Nano to make plugging sensors easier (Figure 30). The board also includes the circuitry necessary for the pulse sensor. The final breakout board was designed with all through-hole components with the exception of the op-amp surface mount to minimize the need for an expert in soldering. It is capable of retrieving data from the following sensors: one IMU, one humidity sensor, one ambient temperature sensor, one IR temperature sensor, one IR pulse sensor, and one GPS. The system is powered at 7.4 volts by two Li-Po batteries. The final dimensions of the board after installation are 1.00” W x 2.65” H x 0.4” T (Figure 30).

Figure 30: Printed Circuit Board Post-Installation with Arduino Nano Mounted

7. SOFTWARE IMPLEMENTATION

The software consists of three modules. The first is the microprocessor code written in C++ that configures the Arduino to collect analog sensor information, processes the analog signals into physically meaningful data, and transmits the data to the Android phone via Bluetooth (Section 17). The second module is the Sensor Uploader (Section 18), a Java program on an Android phone that receives processed sensor data from the Arduino via a Bluetooth connection, parses the data, and then uploads it to a MySQL database. The third is the Data Viewer (Section 0), a website that reads the information from the MySQL database and displays the information in a user friendly format.

7.1. MICROPROCESSOR CODE

The microprocessor handles sensor output in multiple stages. When the microprocessor first starts up, it initializes and configures various sensors and parameters. Next, the microprocessor gathers data from each of the sensors attached to it and computes physically meaningful values from the analog sensor signals. The microprocessor then compares the values received from sensors to the most recent values stored in the device to determine whether a significant change has occurred. If a significant change has occurred, the microprocessor transmits these values via Bluetooth to the Java program that uploads them to the internet.

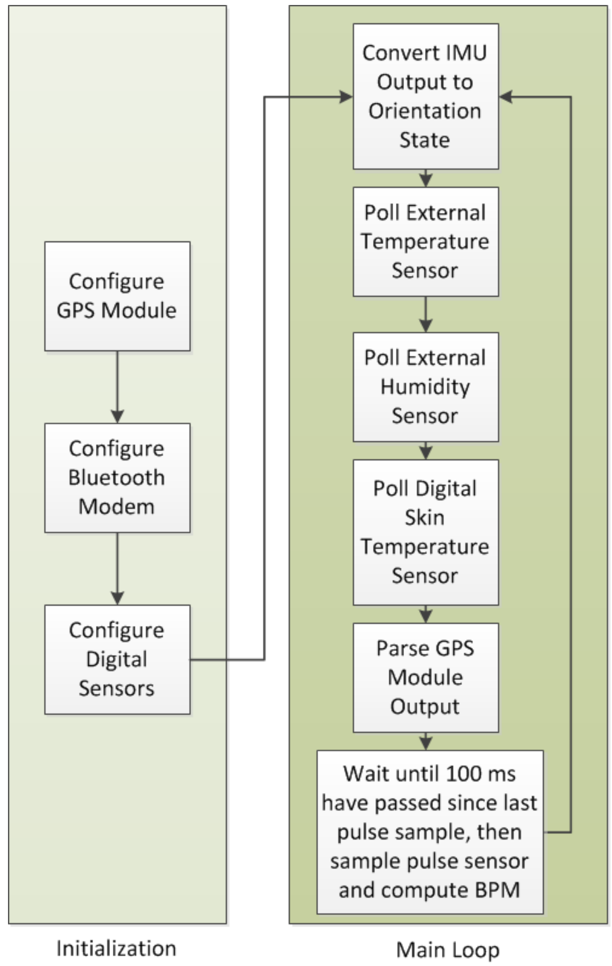


Figure 31: Sequence of Events in Initialization and Main Loop of the Microprocessor Code

The microprocessor code can be divided into the initialization code and the main loop. The initialization code is run once when the microprocessor is turned on or reset. The main loop code is run repeatedly after the initialization code has executed.

The initialization code establishes connections to the IMU, the IR skin temperature sensor, and the GPS module (Figure 31). It also configures its serial pins to transmit at the same baud rate as the Bluetooth modem.

The main loop code for the microprocessor polls each of the sensors, transforms the raw sensor values into physically meaningful values (Figure 31), and determines whether to transmit the new data by comparing them to previous values. The order in which the sensors are polled does not matter.

Angles for yaw, pitch and roll are computed from raw acceleration values (Figure 32). These angles are categorized into one of 3 possible orientation states: standing up, lying down or upside down. Additionally, if the accelerometer has detected a tap event, then the tap is added to a tally of all the taps that have occurred within a certain number of cycles. If enough taps have occurred, then orientation state is switched to a 4th state: walking. If the orientation state is different than the last transmitted state, then the new state is transmitted.

X	Y	Z
0.5g	0.5g	1.0g
0.5g	1.0g	0.5g
0.5g	0.5g	0.0g
0.5g	0.0g	0.5g

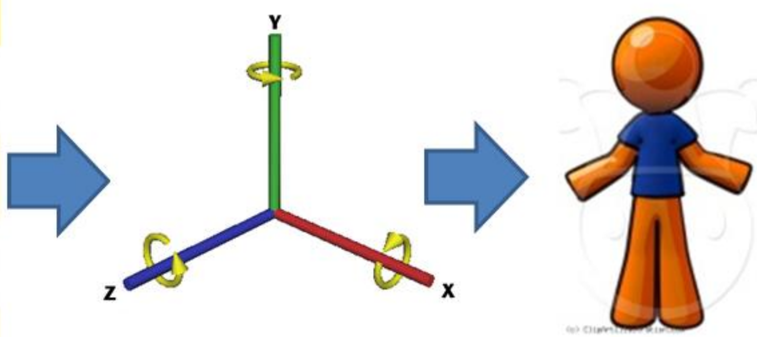


Figure 32: Conversion of Raw Accelerometer Data to Angles to Orientation States

A linear conversion formula from the external temperature sensor's data sheet is employed to convert discretized voltage values ranging from 0 to 1023 into floating point values for degrees Celsius. This value is entered in a moving average filter. If the moving average changes from the last value that was transmitted by a sufficient number of degrees, then the new value is transmitted.

A linear conversion formula from the external humidity sensor's data sheet is employed to convert the discretized voltage value into a floating point value for the humidity percentage. This value is entered in a moving average filter. If the moving average changes from the last value that was transmitted by a sufficient number of percent, then the new value is transmitted.

When the digital skin temperature sensor is polled, the two bytes received from the sensor are converted into a floating point value for degrees Celsius. This value is entered in a moving average filter. If the moving average changes from the last value that was transmitted by a sufficient number of degrees, then the new value is transmitted.

The GPS module is polled and any bytes received from the module are placed into a buffer. The buffer is then parsed and if the buffer contains a full line from the GPS module, then the latitude and longitude values are extracted from the line. If either the latitude or longitude differs significantly from the previous value that was transmitted, then the new value is transmitted.

The code for polling the pulse sensor pauses until a fixed amount of time has passed since the previous sample before collecting a new sample. Currently, this amount of time is set at 100 milliseconds, which is the highest value that gives reliable values. Sampling occurs at regular time intervals because the computation of the heart rate from the raw pulse sensor values requires knowing the amount of time between the data points. After a new value from the pulse sensor is obtained, a threshold is applied that converts the raw signal into a digital value of 1 or 0. This digital value is added to a buffer that stores a set number of the most recently collected values. The number of changes from 0 to 1 in the buffer corresponds to the number of heartbeats. The number of heartbeats is divided by the time that has elapsed between the earliest and latest data stored to get a value of beats per minute (Figure 33). The beats per minute values are entered into a moving average filter. If the moving average changes from the last value that was transmitted by a sufficient number of beats per minute, then the new value is transmitted.

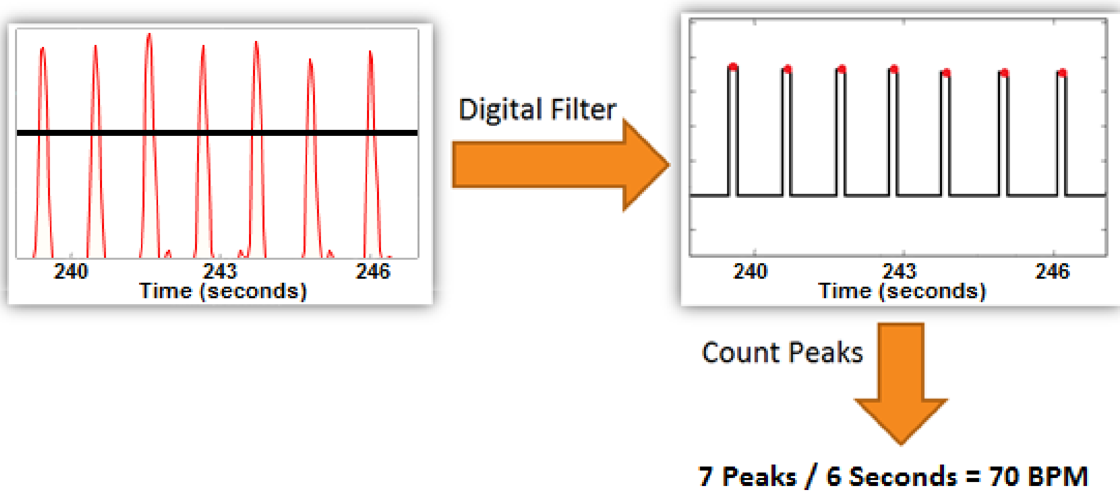


Figure 33: Processing of pulse rate data

The microprocessor transmits values as ASCII strings in the following format:

[*SENSOR_NUMBER*:*SENSOR_VALUE*@*SENSOR_COLLECTION_TIME*]

SENSOR_NUMBER is the number of the sensor for which a value is being transmitted. *SENSOR_VALUE* is the value being transmitted. *SENSOR_COLLECTION_TIME* is the number of milliseconds that have elapsed since the microprocessor was turned on at the time of transmission.

7.2. SENSOR UPLOADER

The Sensor Uploader is a Java program that runs on the Android phone, receives physically meaningful sensor values from the microprocessor via Bluetooth, and then uploads the values to the database (Figure 34).

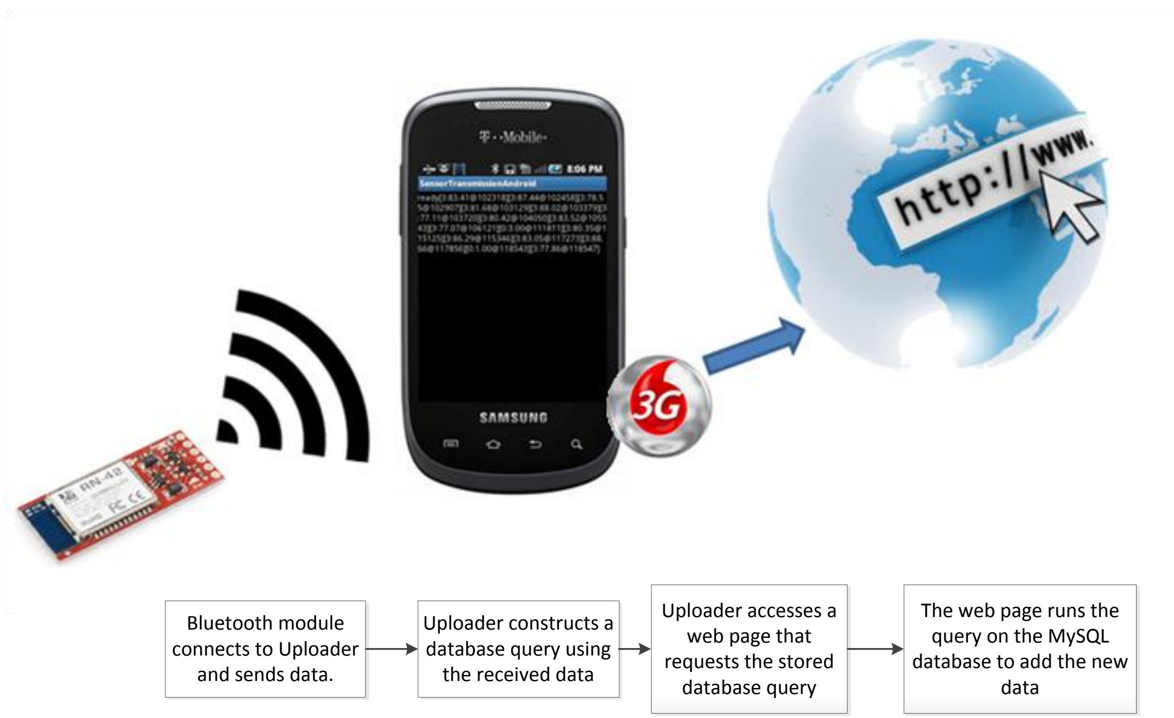


Figure 34: Flow of Data from Bluetooth to Smartphone to Internet

The code for the Sensor Uploader can be broken into two sections: the receiver and the uploader. The receiver accepts the packets transmitted by the microprocessor. These packets describe which sensor has which value and are strings in the format specified at the end of Section 7.1. Once a complete packet is received, it is parsed and its values are passed on to the uploader. The receiver makes use of a third party library called Amarino, which allows Android phones to communicate with Arduino via Bluetooth.

Once the uploader gets new values from the receiver, it sends them to the database. To communicate with the database, the uploader sends a POST request to a PHP script hosted on a webhosting service that also hosts the database. The value of the POST request is a string containing the MySQL query that the uploader wishes to execute on the database. Upon receiving the request, the PHP script executes the query on the database.

7.3. THIRD PARTY USER INTERFACE

The primary objective of the third party user interface is to display gathered data to third party users in a clear and informative manner. The information gathered by a device is displayed primarily on the web pages MainData.php (Figure 37) and DetailData.php (Figure 38).

The third party user interface must also manage third party user access to the data of the elderly user. The information gathered by the sensor devices is highly personal and should not be accessible to the general public. Only those that the elderly user grants permission to should be able to access the elderly user's data. However, the elderly users may be unfamiliar with technology and may not want to use the online interface. The Global Clinic team decided that the best way to give the elderly user control over who could view their information was to have a special access code on the device itself. The purchaser or owner of the device would then be able to enter this code as a validation of data access permission.

The third party user must input this device code on the web page ManageDevices.php (Figure 36) in order to be able to view the data of the elderly individual being monitored by that device.

7.3.1. Database Structure

The online database stores the information on registered third party users, user access permission to data sets, and the data measured by each device. The information stored in the database can be broken down into three main categories.

The first category is the data gathered by the monitoring unit. This section of the database will store the value that a particular sensor measured, which sensor obtained this value, when the sensor obtained this value, and which monitoring device transmitted the data. Therefore, in the SENSOR table of the database each entry represents a packet of data sent by the medical device (Table 3)

Table 3: Fields of SENSOR Table

Field Name	***Field Description***
Id	A unique identifier for each datum in the table. It is an automatically, incrementally assigned integer.
Deviceid	The item code associated with the data of a particular device.
Sensorid	A number that describes which of the sensors in the device the datum is associated with.
Val	The value that the sensor read that was transmitted to the database.
made_on	The date and time stamp that the data was recorded (not uploaded).

The second category is accounts for third party users. When a third party user signs up for the web site and inputs a login and password, this information is stored in the accounts section of the database. In the table called USER, each entry gives information about a particular user, such as email address, username, or password (Table 4).

Table 4: Fields of USER Table

Field Name	***Field Description***
Userid	A unique identifier for each user in the table. It also acts as a user name
Password	A password that helps to ensure that only the person who should be accessing the account is able to access it. This is currently un-encoded, which is a security hazard; however, in the future the passwords will be stored and retrieved with encoding.
made_on	A time and date stamp that records when the account was created.
Fname	The first name of the user.
Lname	The last name of the user.
Email	The email address of the user. In the future this will be used for verifying that the person signing up is in fact a person, and may also be used to send the user notifications of medical events (such as a fall) relating to the elderly user they are monitoring.

The third category is permissions to view the data of a particular monitoring device. This category of stored information determines whether or not a particular user is allowed to view the data associated with a particular device. In VIEWABLE, each entry assigns permission for one user to view one set of data (Table 5).

Table 5: Fields of VIEWABLE Table

Field Name	***Field Description***
viewer	The username of the person who is given permission to view.
viewed	This is a random string that distinguishes what device a datum comes from, and the user must input the item code of a particular device to view the data it transmits.
made_on	A time and date stamp that records when the permission was assigned.
viewed_name	The name that the viewer assigns to the data of a particular device so that the identity of the device's owner is clear and easy to understand.

7.3.2. Accounts

When an elderly individual begins using the monitoring device that the Global Clinic team has developed, third party users will want to view the data being transmitted. If the third party users do not have an account on the web site, they must first create an account. If the third party user already has an account, they only need to log in. The login and account creation processes are done through the web page accounts.php which accesses the USER table for login purposes and adds to the USER table for sign up purposes, as shown in Figure 35. The red conditional statement has yet to be implemented, and user creation does not currently have a validation process for sign up form inputs.

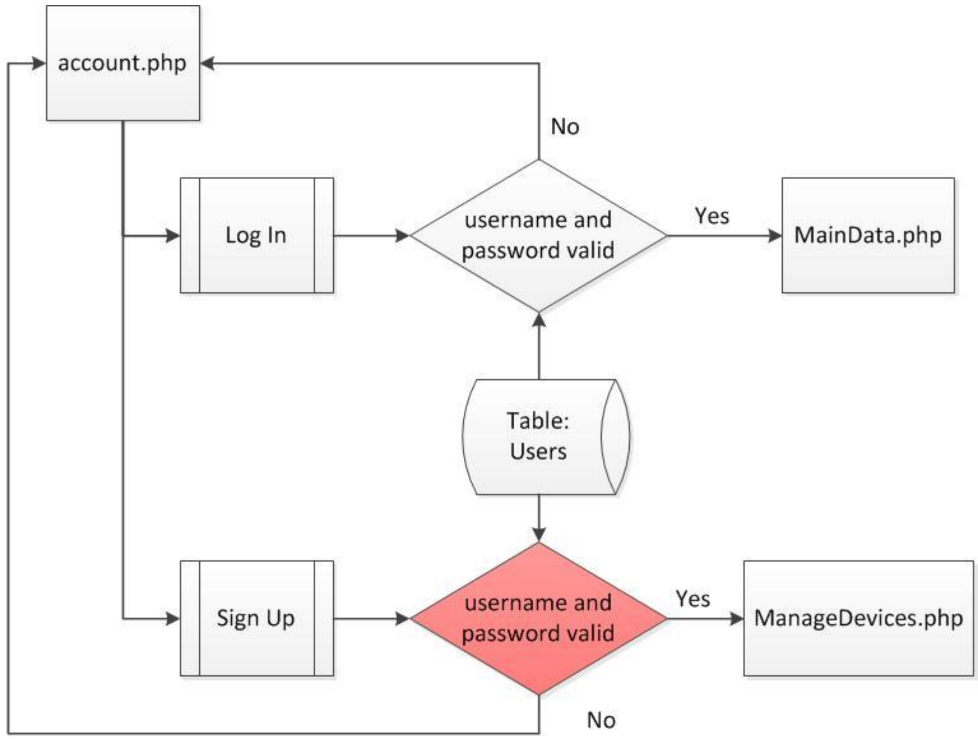


Figure 35: Accounts Management Flow Chart for Third Party User Interface

7.3.3. Managing Devices

Once the third party user creates an account, they are prompted to input the secure code of the monitoring device that they wish to view using the page ManageDevices.php (Figure 36). Once the third party users input this device code, they will be able to see the data that are taken on the elderly individual being monitored. Subsequently, third party users will only need to log in to view the data the monitoring device, because the device viewing permission was stored in the account that the third party user created.

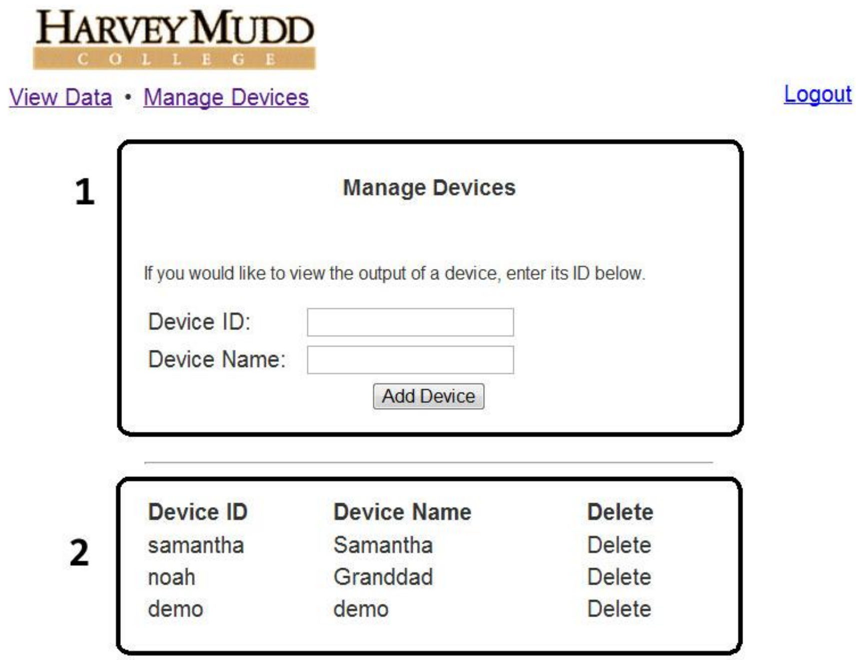


Figure 36: Device Management and Display for the Third Party User Interface

ManageDevices.php grants and removes permissions to view sets of data by adding and removing rows from the VIEWABLE table. Through this page, the third party user can add a device to view that device's data set. When a user adds a device, they will be able to see all the information that has been and is currently associated with the device. Thus, as mentioned previously, the device code is required to add device viewing permissions. In Section 1 of Figure 36, the third party user must input both a device ID and a name referring to that device's owner. The currently assigned viewing permissions are displayed below the form in section 2. Each permission can be easily removed by clicking "Delete."

7.3.4. Main Data Display

Once a third party user has added a device, he or she will be able to view the data a particular device has gathered. The data are displayed on the web site in two ways. The first way that third party users can view the data is through the MainData.php web page (Figure 37).

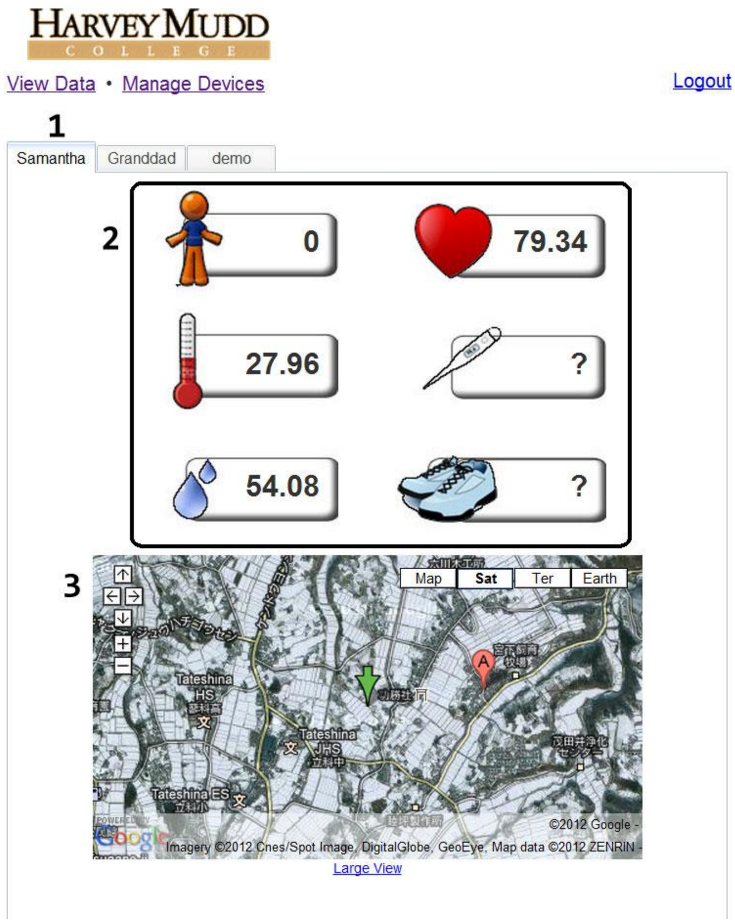


Figure 37: Main data display for the third party user interface

The section of the page labeled 1 is a set of tabs that the third party user can click to view different device outputs. Each tab is associated with a row of the table VIEWABLE where the viewer is the user that is currently logged in. The name of each tab is the name assigned to the view code in the VIEWABLE table.

The section of the page labeled 2 is the most recent set of data stored in the database associated with a particular device. If there is no information for the device under that particular sensor, a "?" will appear instead of the most recent value. Each icon is a link to a more detailed view of the data.

The section of the page labeled 3 is an iframe showing Google Maps. An iframe takes in a URL and displays the web page associated with the URL in a designated section of the page. The Google Maps uses longitude and latitude described in the URL of the iframe reference, so by manipulating the URL the iframe references, the map can be manipulated to show the location of the elderly user. The green arrow represents the longitude and latitude-based location of the device, while the red bubble represents the location of the nearest street address. When the red bubble is clicked, a display pops up showing the address of that location.

7.3.5. Detailed Data Display

The data gathered by a sensor can also be viewed over a period of time. The detailed data display is accessed when the third party user clicks one of the entries containing the most recent sensor data transmitted for a particular user. The default method of displaying the data is through a scatter plot. However, for particular sensors such as the orientation sensor, the information is instead displayed as shown in Figure 38.

Figure 38: Detailed Data Display for the Orientation Sensor

Currently in all of the data display graphs, the data gathered by the device are displayed sequentially ordered from most recent to least recent. However, due to limitations of the code library currently used, specific date and time information is not used in plotting any of the sensor information. Though there were many periods of time when the device was off, the above graph does not display these off times and only portrays the most recent data with equal weight. In the future a different plotting library will be implemented for creating time plots.

8. FINAL HMC PROJECT STATUS

Because HMC completed its academic year in May 2012, the HMC group had to conclude their contributions to the project before then. However, the KU group will continue work until Spring 2013. As of April 2012, the team has completed the goals set in the mid-year report. By the end of the 2011-2012 academic year, the team will deliver the following technical items:

1. One sample demonstration-ready prototype
2. Two conceptual designs of other prototypes
3. 10 sets of electronics for installation into form factors
4. Basic third party user interface
5. Real-time raw data display software (Section 20)

The sample demonstration-ready prototype is a fully functional form factor designed to look like a commercial product with seamless sensor embedding (Section 5). An example of a prototype in this state is shown using a cross-section in Figure 39. The device is in a state where the output data is correctly calibrated, but with further tuning will produce better data consistency. The two conceptual designs are the accessory clip and the remote control, which are described in Sections 5.4 and 5.5. 10 sets of electronics were also prepared to install into future prototypes. The third party user interface is a basic web-application that a user can use to access the output data from the prototype device (Section 7.3). The software for displaying real-time raw data is for make debugging and tuning the prototype easier by having access to live data stream from each sensor.

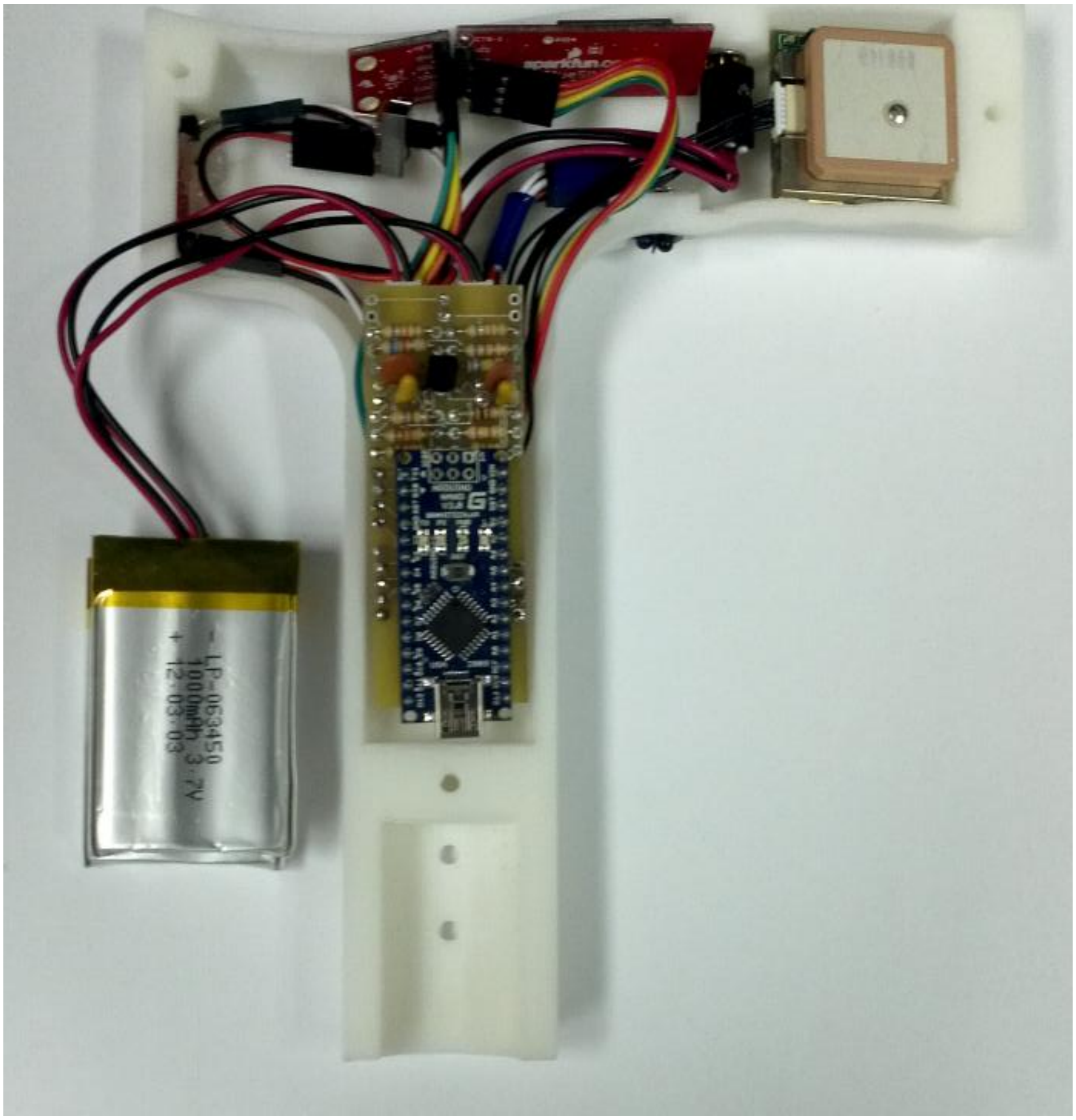


Figure 39: Functional Prototype Disassembled

The demonstration-ready prototype is an implementation of the cane form factor (Figure 39). The cane form factor was chosen because it was deemed feasible to construct in a limited amount of time and significantly different from existing products. To form the shaft of the prototype cane, a commercial cane with an adjustable length was purchased and the handle was removed. To form the handle of the prototype cane, a handle using dimensions similar to that of the commercial cane's handle was designed in Solidworks. Holes to expose the skin temperature, ambient temperature, humidity, and pulse sensors to the outside were added to the handle design. Hollow spaces inside the handle allow the sensors and batteries to fit inside and allow the board to be securely mounted so that it is oriented parallel to the shaft.

The handle was 3D-printed using ABS plastic. After the handle was printed, the sensors that need to be exposed to the outside were placed into their holes and secured with hot glue. Sensors were wired directly to the board. The assembled board and sensors were mounted in the handle.

The board is currently under testing and further calibration. A demonstration video showing basic operations was filmed before the prototype was delivered to the Japanese team for further testing.

9. FUTURE PLANS

9.1. SUMMER 2012

The Global Clinic team will continue developing the two other form factor concepts (the accessory clip and the remote control). Computer aided designs will be used to develop the prototypes to the level of commercial quality and 3D printed for sensor installation. Through these initial prototypes, the team may bring the prototypes to places such as a elderly care center and hospital to ask for feedback on the ergonomics and sensor placement. During this session while the user testing will be small-scale, user testing will direct further development of form factor ergonomics, choice in electronics and sensors, and data display methods for the final product for large scale user testing. Additional PCB designs will also be developed in order to better fit each form factors' interior space.

In terms of electronic hardware, several of the sensors will need to be further calibrated using the sensor development interface software developed this semester. User testing will provide feedback about conditions that makes sensor give unreliable data. Lowering the frequency of such conditions may require altering the location of the sensors in the form factor. Certain effects may also be accounted for using algorithms. For example, if a fall is detected after a pulse rate is no longer being detected for a period of time, the software will indicate

Database input sanitization is vital but currently not implemented. This will occur in the form of simple JavaScript that will check database inputs for potentially damaging characters. This is a necessary and important part of database security.

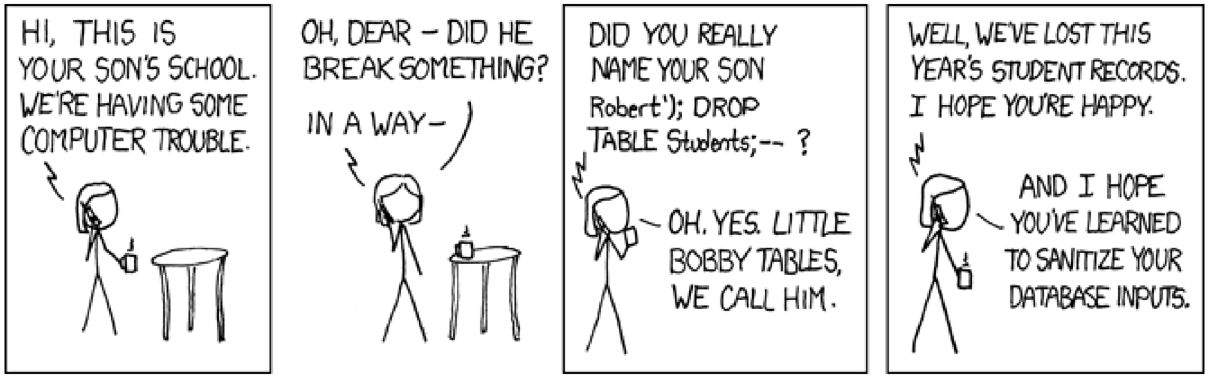


Figure 40: Importance of Database Input Sanitization [23]

9.2. BEYOND SUMMER 2012

9.2.1. Form Factors and Sensor Platform

By the end of Summer 2012, the Global Clinic team will have finished development of the accessory clip and the remote control prototypes and have implemented the sensor platforms in each type. The team can then use the full prototype for the walking cane, accessory clip, and the remote control in the user testing. With the feedback obtained from various users about the prototypes, the Global Clinic team can learn about how the devices will work under uncontrolled environments. In addition, the team can also decide on changes to the sensor platform in accordance to the user feedback. This procedure will continue for several iterations in order to improve the functionality and the ergonomics of each prototype. During this time, it might be possible the users will provide suggestions on other types of form factors to make. These suggestions may be developed into additional form factor concepts and prototypes. By January 2013, the Global Clinic team should have at least three prototypes as final deliverables.

9.2.2. Graphical User Interface

The form factor that is chosen for the final design may include a GUI designed to be used by the elderly individual. This GUI would be an additional software module, which would have different requirements than those for the data viewer. Survey results indicated that most elderly individuals are uncomfortable interacting with new technology like personal computers and smartphones and the design of the GUI must take this into account. To reduce the chance that elderly individuals will be overwhelmed by the GUI's complexity, the GUI will never display more than 3 possible choices at any given time and will display these choices in large buttons. For information that must be displayed in textual form, the characters for the text will be very large. Voice interaction will be considered as an alternative or supplement to graphical interaction. Voice synthesis will be significantly easier to develop than voice recognition, however, because the algorithms involved in voice recognition are complex and the power consumed for the microprocessor to run the algorithms may be considerable.

The third party user interface may also need further refinement through user testing. Another long-term goal is to develop a table of information on what the values of the sensor readouts should be for a healthy and active elderly individual. Though the sensor readouts are currently displayed on the web site, the typical third party user will not be able to interpret certain readouts as "dangerous" unless the web site displays recommended and dangerous sensor readout values. To that end, some research will be necessary to find out exactly what the recommended and dangerous levels would be. Once this is developed, the Global Clinic team may also consider sending alerts via SMS or email to third party users if sensor readout values reach levels perceived to be dangerous.

10. CONCLUSIONS

The Global Clinic team collaborated with students from Kogakuin University to create electronic hardware to measure key health data of an elderly user, a suite of culturally acceptable form factors to house the electronics, a software module to transmit the data from the medical device to an online database, and a web interface for third party users to view data taken. Thus far, the team has created one functional prototype with a cane form factor. This prototype is capable of sending values that the sensors measure to an online database on the internet, where it is displayed by the third party user interface. However, further development of form factor hardware will be necessary. Many of the form factors need to be properly assembled with sensors and tested for durability and suitability for data taking. Additionally, sensors may need to be recalibrated and data processing software may need to be updated to compensate for positioning of the sensors within the form factors. In the future, the Global Clinic team plans to do user testing to further refine the product for the end user.

11. REFERENCES

[1] W. Sun, M. Watanabe, et al, "Factors associated with good self-rated health of non-disabled elderly living alone in Japan: a cross-sectional study", BMC Public Health, vol. 7, pp. 297, Oct. 2007.

[2] "Statistics Bureau Handbook – Chapter 2 Population", 2011 [Online]. Available: http://www.stat.go.jp/english/data/handbook/c02cont.htm. [Accessed: November 30, 2011].

[3] Akiyama et al (2008). Conference Proceeding at American Gerontology Conference: Health Deterioration of Male and Female Elderly [*PowerPoint slides]*. Retrieved from personal contact with Yoichi Tao, Senior Director of SECOM.

[4] Gender Information Site, "At what age do Japanese elderly individuals become dependent or are given assisted living?", October 2011 [Online]. Available: http://www.gender.go.jp/whitepaper/h17/danjyo_gaiyou/danjyo/html/zuhyo/G_27.html. [Accessed: October 10, 2011].

[5] Hachioji-Cho Police Department, "Survey of Wandering Elderly in Hachioji-Cho," Personal Contact by Masahiro Kato, September 26, 2011.

[6] "Mobile BioSensor ABS1 (In Japanese)," 2011 [Online]. Available: http://www.idur.co.jp/sub/sensor-catalog.pdf. [Accessed: November 27, 2011].

[7] "System for Support of Home Treatment (in Japanese)," 2008 [Online]. Available: http://www.parama-tech.com/support/support_04.html. [Accessed: November 27, 2011].

[8] “Welcome to Verity”, 2011 [Online]. Available: http://www.imonsys.com/. [Accessed: November 30, 2011].

[9] M. Tsuji, W. Suzuki, and F. Taoka, “An Economic Evaluation of the Japanese Telehealth System by CVM: Comparison of Four Regions” in *Intl. Workshop on Enterprise Networking and Computing in Healthcare Industry*, Santa Monica, USA, June 2003, pp. 24-28.

[10] “Stock Image of Urban Japanese Person”, April 20 [Online]. Available: http://www.colourbox.com/image/image-860333. [Accessed: April 20, 2012].

[11] “Stock Photography: Lonely Elderly”, April 20 [Online]. Available: http://www.dreamstime.com/stock-photography-lonely-elderly-image21377252. [Accessed: April 20, 2012].

[12] “Bing Images”, April 20 [Online]. Available: http://www.bing.com/images. [Accessed: April 20, 2012].

[13] “Sparkfun”, April 19 [Online]. Available: http://www.sparkfun.com/. [Accessed: April 19, 2012].

[14] “ITG-3200”, April 28 [Online]. Available: http://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf. [Accessed: April 28, 2012].

[15] “ADXL345”, April 28 [Online]. Available: http://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf. [Accessed: April 28, 2012].

[16] "Reflective Optical Sensor with Transistor Output", April 19 [Online]. Available: http://www.vishay.com/docs/83760/tcrt5000.pdf. [Accessed: April 19, 2012].

[17] "LM35CAZ/NOPB", April 19 [Online]. Available: http://www.mouser.com/ProductDetail/National-Semiconductor-TI/LM35CAZ-NOPB/?qs=sGAEpiMZZMucenltShoSnkuGO8CMxhbWcN6Wkm5ZDpg%3d. [Accessed: April 19, 2012].

[18] "HIH-4030/31 Series", April 28 [Online]. Available: http://www.sparkfun.com/datasheets/Sensors/Weather/SEN-09569-HIH-4030-datasheet.pdf. [Accessed: April 28, 2012].

[19] "MLX90614 Infrared Thermometer", April 20 [Online]. Available: http://www.abra-electronics.com/products/MLX90614ESF%252dBAA-Infrared-Thermometer.html. [Accessed: April 20, 2012].

[20] "RN-42/RN-42-N Data Sheet", April 29 [Online]. Available: http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/Bluetooth/Bluetooth-RN-42-DS.pdf. [Accessed: April 29, 2012].

[21] "Samsung Dart Android Smartphone", April 19 [Online]. Available: http://www.samsung.com/us/mobile/cell-phones/SGH-T499MABTMB. [Accessed: April 19, 2012].

[22] "Arduino Nano", April 17 [Online]. Available: http://www.jameco.com/Jameco/Products/MakeImag/2121148.jpg. [Accessed: April 17, 2012].

[23] "Exploits of a Mom", April 17 [Online]. Available: http://xkcd.com/327/. [Accessed: April 17, 2012].

12. APPENDIX A: FORM FACTOR OPTIONS

"iPhone" - This form factor is an iPhone-like-device with an LCD touch screen. The user interface facilitates measurement through a simplified graphical user interface that elderly users can be made comfortable with. Sensors are embedded inside the iPhone. Heartbeat sensor is located on the surface of the iPhone. When the user takes the device with them, the device will sit in the user's pocket and measures user activity levels when they are about and allows for GPS positioning and body orientation measurement.

Sensor Keychain - Sensors are integrated into a keychain attachment that can transmit sensor information to a stationary hub or cell phone towers. For the user to leave the house, he or she must retrieve the keychain from the dock for the door to open. When re-entering the house, if the user forgets to replace the keychain to the dock, a voice reminder will be activated. The keychain sits in the user's pocket and measures user activity level when they are about and allows for GPS positioning and body orientation measurement. The primary downside to this design is that it is unable to easily measure heart rate or body temperature. In addition, the KU group has reported that rural Japanese individuals do not often lock their doors.

Bracelet/Watch - Sensors are embedded into a device that will wrap around the user's wrist and resemble a bracelet or a watch. This device can be used to detect heart rate, body temperature, GPS positioning, and possibly some activity levels. However, activity levels may be difficult to measure since the sensors are subject to arm motions.

Television Data Overlay System - The results from the survey conducted by the Japanese students suggested an overwhelming majority of Japanese elderly individuals watch television. Those surveyed also indicated interest in knowing data such as their own health vitals, the surrounding temperature, etc. Combining these needs, a device that can overlay the sensor information on a television screen could encourage usage of the measurement unit, which might be in the form of a television remote or an arm chair in front of the television.

"SupaFurby" - The form factor for this device is a robotic plush animal that can speak and move. This user interface facilitates measurement by making interaction with the device entertaining. However, it is limited in the information it can gather because the user does not touch the device often and does not carry the device with them.

Arm Rest Cover – Because an elderly individual will undoubtedly sit in some type of chair, an arm rest cover can be used to acquire information such as blood pressure or heart rate. Sensors may be embedded along the length of the cover in order to accomplish that task. An extension of the arm rest cover is seat cover that can be used to acquire the user's weight. The arm rest cover is convenient since it does not require any modifications to a user's lifestyle and can easily fit into a user's home. However, a downside is that certain sensors, such as GPS and accelerometer, will not be useful.

IR Surveillance Cameras – By setting up various infrared surveillance cameras at critical locations in a user's home, such as in front of the television set or in a bedroom, the system is able to keep track of the user's activity and to ensure that there is not an emergency in the house. Although this would not be able to provide much health information, it will be able to detect the relative activity levels within a house. A complex enough algorithm may be able to determine irregularities in activities around the house or emergency situations such as falls. However, this system would only be effective inside the house and will not be of much use outside of the house.

Eyeglasses – If the sensor platform were designed into the frame of the glasses, then some health information (e.g. body temperature, heart rate, and activity levels) can be measured simply through the contact of the frame with the skin. Because the elderly individual may be wearing their eyeglasses for extended periods of time, obtaining enough quality data would be easier than with devices that only have short periods of contact. However, designing the eyeglasses into a shape and style that would be attractive to a large proportion of the elderly population could be difficult. It may be easier to create a cover to go over part of the eyeglasses' frame.

Door Knob – Elderly individuals may have to use door knobs or some type of door handle in order to get into and out of rooms. If sensors were incorporated into these door knobs, then every time the elderly individual passes through the door, there is a chance for the door knob to obtain enough health information for processing. Because the door knobs are relatively stationary within the house, transmitting data wirelessly to a central processing hub would not be difficult. However, once the elderly individual leaves the house, then the door knobs would not be able to continue monitoring that individual. Additional checks have to be put in so that the door knob is able to identify the person going through the door. Another point to keep in mind is that some Japanese households use sliding doors rather than swinging doors. These sliding doors often do not have any type of knobs, which means that the door knob idea would have to be modified to fit in with a sliding door model.

Stationary Objects – Sensors could be built into various stationary items, such as pens, so that they can measure health vitals every time the elderly individual uses them. Because stationary is common in most households, it should be easy to obtain health information about the elderly individuals. However, this is dependent on how often elderly individuals use these objects and the duration which they use them. In addition, incorporating sensors into stationary may make the objects more expensive than other types of stationary, which would decrease their competitive market.

Toilet Seat – The elderly individuals use the toilet every day for some period of time. Sensors built into the seat can use that opportunity to collect health information from the elderly individuals. The extended, stable contact between elderly individual and the toilet seat will provide ample opportunity for the sensors to collect dependable information on weight, body temperature, and possibly heart rate. Because the toilet seat is stationary, it will be unable to collect further information about the users once they leave. In addition, the toilet seat must also determine the person using the toilet seat in order for health vital recordings are saved for the correct user.

13. APPENDIX B: MATERIAL LISTS

13.1. BILL OF MATERIALS FOR PCB

Part	**Value**	**Device**	**Package**
Batt1	Batt	M02-JST-2MM-SMT	JST-2-SMD
Batt2	Batt	M02-JST-2MM-SMT	JST-2-SMD
C1	1uF	CAPPTH1	CAP-PTH-5MM
C2	1uF	CAPPTH1	CAP-PTH-5MM
C3	0.1uF	CAPPTH1	CAP-PTH-5MM
C4	0.1uF	CAPPTH1	CAP-PTH-5MM
C5	0.1uF	CAPPTH1	CAP-PTH-5MM
D1	red	LED0603	LED-0603
IC1	LM358MX	LM358MX	SO08
IC2	LM358MX	LM358MX	SO08
J1	HEAD15-NOSS	HEAD15-NOSS	HEAD15-NOSS
J2	HEAD15-NOSS-1	HEAD15-NOSS-1	HEAD15-NOSS-1
JP1	BLUETOOTH	M04PTH	1X04
JP2	ACCEL	M04PTH	1X04
JP3	HUMIDITY	M03PTH	1X03

JP4	PULSE	M04PTH	1X04
JP5	IRTEMP	M04PTH	1X04
JP6	AMB_TEMP	M04PTH	1X03
JP7	GPS	M05PTH	1X05
JP8	SWITCH	M02PTH	1X02
JP9	SWITCH1	M02PTH	1X02
JP10	SWITCH2	M02PTH	1X02
JP11	BATT1	M02PTH	1X02
JP12	BATT2	M02PTH	1X02
Q1		TRANSISTOR_NPNTO-92-AMMO	TO-92-AMMO
R1	240	RESISTORPTH-1/4W	AXIAL-0.4
R2	10k	RESISTORPTH-1/4W	AXIAL-0.4
R3	6.8k	RESISTORPTH-1/4W	AXIAL-0.4
R4	680	RESISTORPTH-1/4W	AXIAL-0.4
R5	68k	RESISTORPTH-1/4W	AXIAL-0.4
R6	6.8k	RESISTORPTH-1/4W	AXIAL-0.4
R7	100k	RESISTORPTH-1/4W	AXIAL-0.4
R8	68k	RESISTORPTH-1/4W	AXIAL-0.4

R9	470	RESISTORPTH-1/4W	AXIAL-0.4
R10	1k	RESISTORPTH-1/4W	AXIAL-0.4
R11	10k	RESISTORPTH-1/4W	AXIAL-0.4
R12	10k	RESISTORPTH-1/4W	AXIAL-0.4

13.2. TECHNICAL SPECIFICATIONS FOR PARTS

Part Name	**Part Number**	**Distributor**	**Dimensions (inches)**	**Required Voltage (V)**	**Required Current (mA)**
IMU Digital Combo Board – ADXL345/ITG3200	SEN-10121	Sparkfun	0.65 x 0.63	2.1 – 3.6	6.645
Humidity Sensor – HIH4030	SEN-05969	Sparkfun	0.75 x 0.30	4.0 – 5.8	0.200
GlobalSat EM-406a	182135	Expansys	1.2 x 1.2 x 0.4	4.5 – 6.5	44 (Active)
Bluetooth Modem – BlueSMiRF Silver	WRL-10269	Sparkfun	1.77 x 0.65 x 0.15	3.0 – 3.6	40 – 45 (Active)
IC Thermometer IR	MLX90614ESF-DAA	Digikey	0.36 x 0.36 x 0.16	2.6 – 3.6	1 – 2
LM35DZ-MD	LM35DZ/NOPB	Digikey	0.20 x 0.20 x 0.17	4.0 – 30	0.060
Reflective Optical Sensor 950 nm	TCRT5000L	Tayda Electronics	0.40 x 0.23 x 0.28		1
Arduino Nano v3.0	B003YVL34O	Amazon	1.70 x 0.73 x 0.8	7.0 – 12.0	40
Polymer Lithium Ion Battery – 1000mAh	PRT-00339	Sparkfun	2.09 x 1.30 x 0.23	3.7	
LiPo Charger Basic – Micro-USB	PRT-10217	Sparkfun	1.16 x 0.43	3.75 – 6.0	0.510 – 1.500
JST Right Angle Connector	PRT-08612	Sparkfun			
Resistor 240 Ohms Carbon Film 1/4W 5%	ERD-S2TJ241V	Digikey	0.067 diameter x 0.126 length		
Resistor 240 Ohms	CFM14JT470	Digikey	0.067		

Carbon Film 1/4W 5%	R		diameter x 0.130 length		
Resistor 680 Ohms Carbon Film 1/4W 5%	CFM14JT680R	Digikey	0.067 diameter x 0.130 length		
Resistor 1k Ohms Carbon Film 1/4W 5%	CFM14JT1K00	Digikey	0.067 diameter x 0.130 length		
Resistor 6.8k Ohms Carbon Film 1/4W 5%	CFM14JT6K80	Digikey	0.067 diameter x 0.130 length		
Resistor 10k Ohms Carbon Film 1/4W 5%	CFM14JT10k0	Digikey	0.067 diameter x 0.130 length		
Resistor 68k Ohms Carbon Film 1/4W 5%	CFM14JT68k0	Digikey	0.067 diameter x 0.130 length		
Resistor 68k Ohms Carbon Film 1/4W 5%	CFM14JT68k0	Digikey	0.067 diameter x 0.130 length		
Resistor 100k Ohms Carbon Film 1/4W 5%	CFM14JT100K	Digikey	0.067 diameter x 0.130 length		
Capacitor Ceramic 0.1uF 50V 5% Radial	FK20C0G1H104J	Digikey	0.217 x 0.138		
Capacitor Ceramic 1uF 50V 5% Radial	FK24X7R1H105K	Digikey	0.177 x 0.098		
IC OpAmp Dual 0-70 Degree LM358	LM368MX	Digikey	0.154 x 0.154		
Transistor NPN 400V 2A TO-92	STX13004G	Digikey			
LED 630nm Red Diffused 0603 SMD	HSMS-C190	Digikey	0.06 x 0.03		
Customized Arduino Nano Shield		4PCB.com	2.60 x 1.00		

13.3. COST FOR PARTS

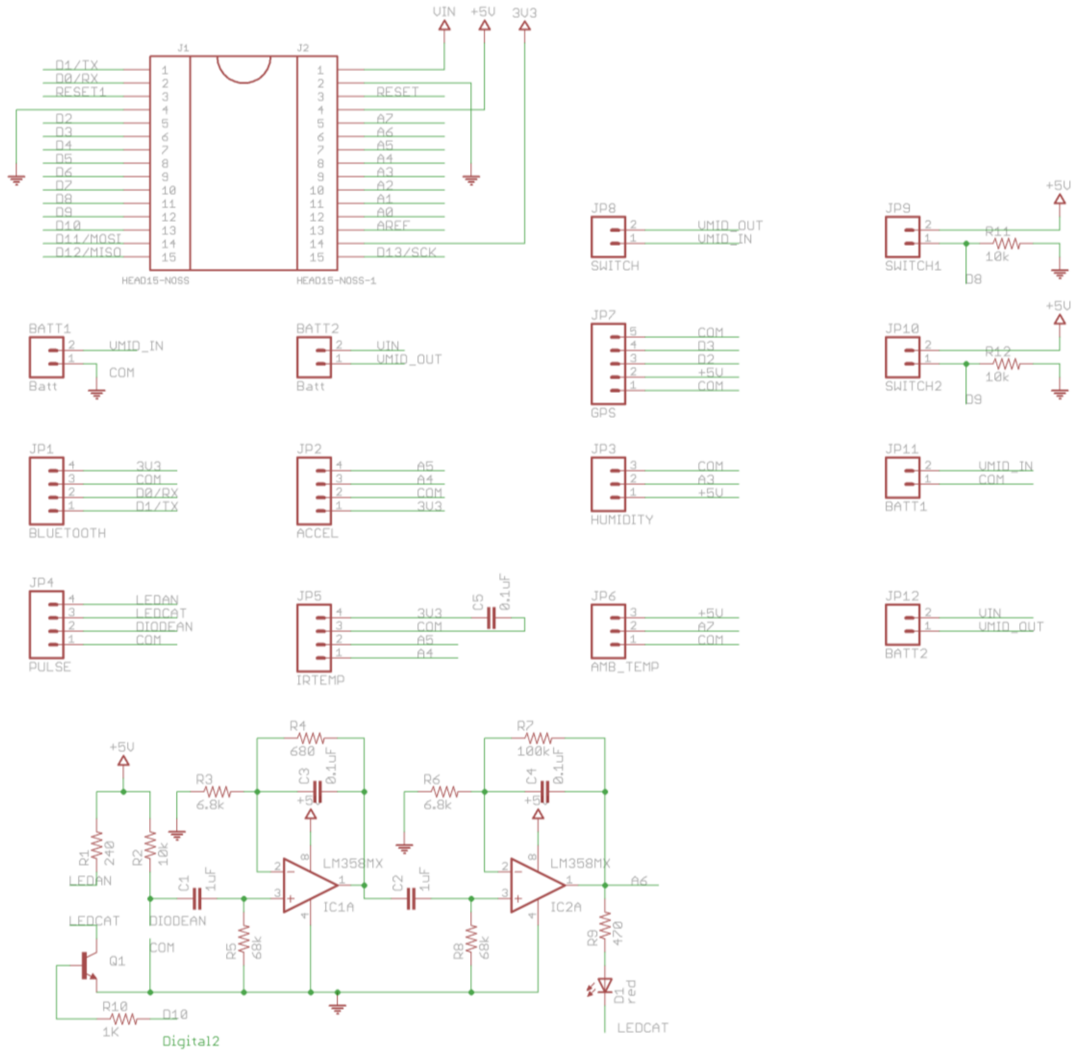
Part Name	Part Number	Distributor	Quantity	Price Per Unit	Total Cost
IMU Digital Combo Board – ADXL345/ITG3200	SEN-10121	Sparkfun	1	$64.95	$64.95
Humidity Sensor – HIH4030	SEN-05969	Sparkfun	1	$16.95	$16.95
GlobalSat EM-406a	182135	Expansys	1	$39.99	$39.99
Bluetooth Modem – BlueSMiRF Silver	WRL-10269	Sparkfun	1	$39.95	$39.95
IC Thermometer IR	MLX90614ESF-DAA	Digikey	1	$14.01	$14.01
LM35DZ-MD	LM35DZ/NOPB	Digikey	1	$1.57	$1.57
Reflective Optical Sensor 950 nm	TCRT5000L	Tayda Electronics	1	$0.49	$0.49
Arduino Nano v3.0	B003YVL34O	Amazon	1	$37.50	$37.50
Polymer Lithium Ion Battery – 1000mAh	PRT-00339	Sparkfun	2	$11.95	$23.90
LiPo Charger Basic – Micro-USB	PRT-10217	Sparkfun	1	$9.95	$.9.95
JST Right Angle Connector	PRT-08612	Sparkfun	2	$0.95	$1.90
Resistor 240 Ohms Carbon Film 1/4W 5%	ERD-S2TJ241V	Digikey	1	$0.09	$0.09
Resistor 240 Ohms Carbon Film 1/4W 5%	CFM14JT470R	Digikey	1	$0.08	$0.08
Resistor 680 Ohms Carbon Film 1/4W 5%	CFM14JT680R	Digikey	1	$0.08	$0.08
Resistor 1k Ohms Carbon Film 1/4W 5%	CFM14JT1K00	Digikey	1	$0.08	$0.08
Resistor 6.8k Ohms Carbon Film 1/4W 5%	CFM14JT6K80	Digikey	2	$0.08	$0.16
Resistor 10k Ohms Carbon Film 1/4W 5%	CFM14JT10k0	Digikey	3	$0.08	$0.24
Resistor 68k Ohms Carbon Film 1/4W 5%	CFM14JT68k0	Digikey	2	$0.08	$0.16

Resistor 100k Ohms Carbon Film 1/4W 5%	CFM14JT100K	Digikey	1	$0.08	$0.08
Capacitor Ceramic 0.1uF 50V 5% Radial	FK20C0G1H104J	Digikey	3	$0.96	$2.88
Capacitor Ceramic 1uF 50V 5% Radial	FK24X7R1H105K	Digikey	2	$0.70	$1.40
IC OpAmp Dual 0-70 Degree LM358	LM368MX	Digikey	2	$0.44	$0.88
Transistor NPN 400V 2A TO-92	STX13004G	Digikey	1	$0.13	$0.13
LED 630nm Red Diffused 0603 SMD	HSMS-C190	Digikey	1	$0.50	$0.50
Customized Arduino Nano Shield		4PCB.com	1	$33	$33

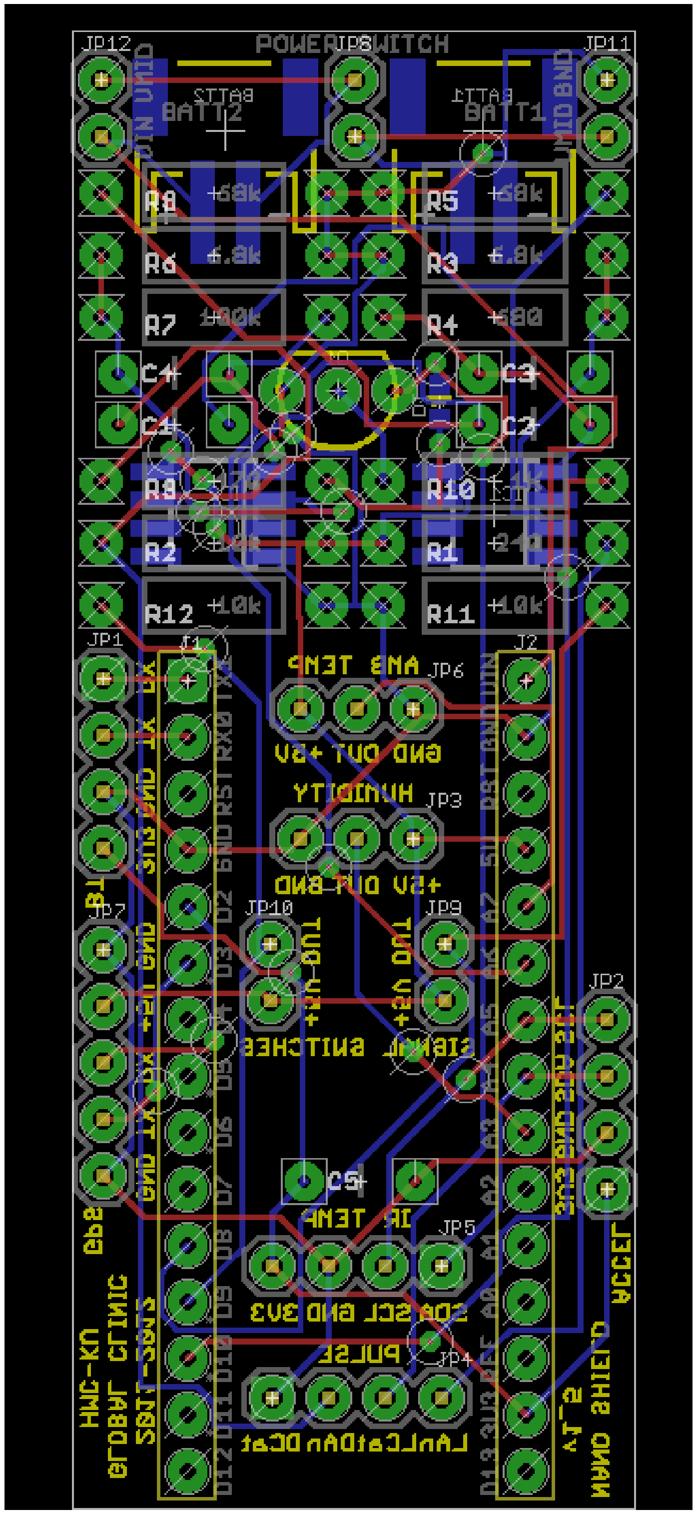
14. APPENDIX C: POWER ANALYSIS FOR SENSOR PLATFORM

Sensor/Component	Description	Typical Voltage (V)	Average Ampere Usage (µA)	Maximum Ampere Usage (µA)	Comments
ADXL345+ITG3200 IMU Combo	Accelerometer and Gyroscope	2.1-3.6	6645		
HIH4030	Humidity Sensor	4-5.8	200	500	
LM35	Environment Temperature	5	60		
MLX90614	IR Temperature Sensor	5	1000	2000	
RN41	Bluetooth Module	3.3	40000		Active
			8000		Sniff
Arduino Nano v3.0	Computing Platform	7-12	40000		Active
EM406a	GPS	4.5-6.5	44000		Active
			25000		Trickle
Reflective Optics	Pulse Photosensor		1000		
GPS Active Time (%) (Approx.)	1				
Bluetooth Transmission Time (%) (Approx.)	1				
IR Sensor Active Time (%) (Approx.)	1				
Total (mAh)	132.905				
Battery Power Capacity (mAh)	1000				
Approximate Life (hours)	7.52				

15. APPENDIX D: PRINTED CIRCUIT BOARD SCHEMATIC



16. APPENDIX E: PRINTED CIRCUIT BOARD LAYOUT



17. APPENDIX F: ARDUINO CODE

```
//CONFIGURATION SETTINGS
#define PULSE_SAMPLE_PERIOD                100.0   //Time in milliseconds between samples of the pulse sensor
#define BPM_COMPUTATION_PERIOD           10 //How many iterations of the main loop do we go through before we recompute BPM?
#define PULSE_BUFFER_SIZE                250 //How big is the buffer that stores raw pulse samples
#define PULSE_DIGITAL_FILTER_THRESHOLD 110 //What is the raw analog value above which a pulse value is counted as a peak? Voltage = 5*(raw/1023)
#define PULSE_TRANSISTOR_PIN           10

#define SKIN_TEMP_SAMPLE_PERIOD 11 //How many iterations of the main loop do we go through before we resample SKIN TEMP?


#define ANGULAR_ORIENTATION_THRESHOLD 30

#define WALKING_FREQ_THRESHOLD 0.12 //If taps / (no taps + taps) exceeds this threshold, then we are walking
#define STANDING      0
#define FLAT_ON_BACK  1
#define FLAT_ON_FACE  2
#define WALKING       3

#define CONNECT_TO_ANDROID true //Set to true if we want to talk to an android smartphone

#if CONNECT_TO_ANDROID == false
  #define MAIN_SERIAL_BAUD_RATE 9600 //Baud rate for the serial device that is talking to the smartphone. Should be 115200 for BlueSmirf and 9600 for mini-usb
#else
  #define MAIN_SERIAL_BAUD_RATE 115200 //Baud rate for the serial device that is talking to the smartphone. Should be 115200 for BlueSmirf and 9600 for mini-usb
#endif

//Sensor Numbers For DataBase Upload
#define ORIENTATION_SENSOR_NUMBER    0
#define EXT_TEMP_SENSOR_NUMBER       1
#define HUMIDITY_SENSOR_NUMBER       2
#define SKIN_TEMP_SENSOR_NUMBER      3
#define HEARTRATE_SENSOR_NUMBER      6
#define LONGITUDE_SENSOR_NUMBER      7
#define LATITUDE_SENSOR_NUMBER       8

//Sensor Pin Numbers
#define PULSE_SENSOR_PIN            A6
#define EXT_TEMPERATURE_SENSOR_PIN  A7
#define HUMIDITY_SENSOR_PIN         A3
#define GPS_RX_PIN                  3
#define GPS_TX_PIN                  2
```

```
//Values  for  how  much  a  sensor  has  to  change  from  the  previously
transmitted value
//before a new value is transmitted
#define SKIN_TEMP_CHANGE_TOLERANCE   0.1f
#define GPS_CHANGE_TOLERANCE         0.001f
#define HEARTRATE_CHANGE_TOLERANCE   2
#define EXT_TEMP_CHANGE_TOLERANCE    2
#define HUMIDITY_CHANGE_TOLERANCE    2

//For orientation, how many consistent state changes do we need before it
is accepted?
#define ORIENTATION_STATE_CHANGE_COUNTER_TOLERANCE 5

//Size of the buffer for our moving average filters
#define EXT_TEMP_MOVING_AVERAGE_WINDOW_SIZE   30
#define HUMIDITY_MOVING_AVERAGE_WINDOW_SIZE   10
#define WALKING_MOVING_AVERAGE_WINDOW_SIZE    60

/////////////////

#include <Wire.h>
#include <ADXL345.h>

//ANDROID ONLY
#include <MeetAndroid.h>
MeetAndroid meetAndroid;

//TIMER INIT
#include <TimerOne.h>
#include <MsTimer2.h>

void setup_TIMER()
{

 /*
  Timer1.initialize(PULSE_SAMPLE_PERIOD * 1000);              // initialize
timer1, and set a 1/2 second period
  Timer1.pwm(9, 512);                                          // setup pwm
on pin 9, 50% duty cycle
  Timer1.attachInterrupt(sampleSet);                            //  attaches
routine to sample pulseOx to Timer 1

  MsTimer2::set(SKIN_TEMP_SAMPLE_PERIOD,   sample_skin_temp);   //attaches
routine to sample skin temperature to Timer 2
  MsTimer2::start();                                          //starts Timer
2
 */
}

boolean sample = false;

void sampleSet()
{
  sample = true;
}
```

```
//Heart Beat Window INIT
/*
The heartBeatWindow class is a filter that stores a window of the several
of the last
readings from the pulse sensor and keeps track of the number of peaks in
that window
and computes the bpm from the number of peaks
*/
class heartBeatWindow
{
   boolean * readings; //buffer storing whether each reading was HIGH or
LOW, boolean saves us a ton of space
   int numDatas;       //number of readings current in the buffer
   int index;          //index of the buffer for the newest reading
   int iter_start;     //index of the buffer for the oldest reading
   int windowSize;        //maximum number of readings are buffer holds
before it starts discarding old readings

   public:
   unsigned long last_sample_time;

   double BPM;         //Most recently computed Beats Per Minute

   heartBeatWindow(int windowSize_):numDatas(0), windowSize(windowSize_),
index(0), iter_start(0)
   {
     readings = (boolean *) malloc(windowSize * sizeof(boolean));
//Allocates memory for the dynamic array readings
   }

   double calculateBPM()
   {
      int dataSize = numDatas;
      int numPeaks = 0;
      int iter_start_temp = iter_start;
      boolean hit_top = false; //Have we previously seen a HIGH in our
loop through the readings?

      //Loops through the array of readings, from oldest to newest
      //Each time it detects a HIGH when it had previously seen a LOW
      //It adds one to the peak counter
      //If it detects a HIGH when it had previously seen a HIGH it does
nothing

      for(int i = 0; i < dataSize; ++i)
      {
          boolean this_val = readings[iter_start_temp];

          //We will have to loop around the buffer if our oldest entry is
not at index 0
          iter_start_temp++;
          if (iter_start_temp >= windowSize)
            iter_start_temp = 0;

          if (!hit_top && this_val != false)
          {
             ++numPeaks;
```

```
            hit_top = true;
          }

          if (this_val == false)
            hit_top = false;
      }

      //Stores the time elapsed between the first and last reading in
seconds. Since there is a constant spacing between readings this is easy
to compute
      double      secondsDifference      =      ((double)(dataSize      *
PULSE_SAMPLE_PERIOD)) / 1000.0;

      //Once we have computed the number of peaks we divide it by the time
elapsed in seconds
      double peaksPerSecond = (double) numPeaks / secondsDifference;

      BPM = peaksPerSecond * 60.0;

      //Old debugging code
      /*
      Serial.print("BPM ");
      Serial.print(BPM);
      Serial.print(" = ");
      Serial.print(numPeaks);
      Serial.print(" / ");
      Serial.println(secondsDifference);
      */

      return BPM;
   }

   //Old debugging code
   /*
   void printContents()
   {
     Serial.print("LIST START ");
     int list_size = 0;
     int iter_start_temp = iter_start;

     for(int i = 0; i < numDatas; ++i)
     {
        boolean this_val = readings[iter_start_temp];

        iter_start_temp++;
        if (iter_start_temp >= windowSize)
          iter_start_temp = 0;

        Serial.print(this_val);
        Serial.print(", ");

        ++list_size;
     }
     Serial.print("LIST END SIZE = ");
     Serial.println(list_size);
   }
   */
```

```
   //When the instance is destroyed, we free the dynamically allocated
array readings
   ~heartBeatWindow()
   {
     free(readings);
   }

   //Called when a new reading is added to the buffer
   void addValue(boolean newValue)
   {
      last_sample_time = millis();

      // read from the sensor:
      readings[index] = newValue;
      // advance to the next position in the array:
      index = index + 1;

      if (numDatas < windowSize)
        numDatas++;
      else
      {
        //Once we start discarding old data, the index of the oldest data
point starts shifting up every time we add new data
        iter_start++;
        if (iter_start >= windowSize)
         iter_start = 0;
      }

      // if we're at the end of the array then we're going to start
overwriting the oldest data in the buffer
      if (index >= windowSize)
        index = 0;
   }
};

//Declare a heart beat window instance
heartBeatWindow pulseOxHeartBeatWindow(PULSE_BUFFER_SIZE);
//Lets us keep track of how many iterations of the main loop we've been
through
//So we can sample over few iterations
int pulseOxSamplerIter = 0;

//Routine called by interrupt which reads the analog pin for the pulse
sensor
//Samples it, thresholds it to a HIGH or LOW value and passes the
thresholded
//value to the heart beat window
void samplePulseOx()
{
   int sensor_value = analogRead(PULSE_SENSOR_PIN);

   if (sensor_value > PULSE_DIGITAL_FILTER_THRESHOLD)
     pulseOxHeartBeatWindow.addValue(true);
   else
     pulseOxHeartBeatWindow.addValue(false);
```

```
}

//Sets transistor for pulseOx LED to HIGH
void setup_pulseOx()
{
  pinMode(PULSE_TRANSISTOR_PIN, OUTPUT);
  digitalWrite(PULSE_TRANSISTOR_PIN, HIGH);
}


//Moving Average Filter INIT
/*
The moving average filter is a filter that keeps a window of the last few
values
put into it and keeps track of the average of the values in the window

It is used to smooth out noise for noisy sensors

Very similar in structure to heartbeatWindow actually. I should've made
these descendants of a single class.
*/
class movingAverageFilter
{
  float * readings; //dynamic array to store the values read
  int mySize;
  int numDatas;
  int index;
  double total;
  public:

   movingAverageFilter(int size_)
   {
     mySize = size_;
     total = 0;
     index = 0;
     numDatas = 0;

     readings = (float*) malloc(size_ * sizeof(float));

     for(int i = 0; i < mySize; ++i)
       readings[i] = 0;
   }

   void addValue(float val)
   {
    total = total - readings[index];
    // read from the sensor:
    readings[index] = val;
    // add the reading to the total:
    total= total + readings[index];
    // advance to the next position in the array:
    index = index + 1;

    if (numDatas < mySize)
      numDatas++;

    // if we're at the end of the array...
```

```
     if (index >= mySize)
       index = 0;
    }

    //Take that average!
    double getAverage()
    {
      return total / ((double)numDatas);
    }

    ~movingAverageFilter()
    {
      free(readings);
    }
};

//Declares moving average filters for the noisy sensors that need them
movingAverageFilter
ExternalTempMovingAverage(EXT_TEMP_MOVING_AVERAGE_WINDOW_SIZE);
movingAverageFilter
HumidityMovingAverage(HUMIDITY_MOVING_AVERAGE_WINDOW_SIZE);
//Walking filter is just using it keep track of the frequency of taps
//that have been detected in a window of time.
movingAverageFilter walkingFilter(WALKING_MOVING_AVERAGE_WINDOW_SIZE);

//GPS INIT
//GPS runs off an emulated serial port controlled by the software
#include <SoftwareSerial.h>
SoftwareSerial GPSSerial(GPS_RX_PIN, GPS_TX_PIN); //RX, TX

void setup_GPS()
{
  //The module runs at 4800 baud
  GPSSerial.begin(4800);
}

//SKIN TEMP INIT
#include <i2cmaster.h>

//I2C protocols to register the device
void setup_skin_temp()
{
  i2c_init(); //Initialise the i2c bus
  PORTC = (1 << PORTC4) | (1 << PORTC5);//enable pullups
}

//ACCELEROMETER INIT
//I am no longer using the GYROSCOPE because the IMU calculations took too
long
ADXL345 adxl;

void setup_imu()
{
  adxl.powerOn();

  //set activity/ inactivity thresholds (0-255)
  adxl.setActivityThreshold(15); //62.5mg per increment
```

```
  adxl.setInactivityThreshold(15); //62.5mg per increment
  adxl.setTimeInactivity(10);  // how many seconds of no activity is
inactive?

  //look of activity movement on this axes - 1 == on; 0 == off
  adxl.setActivityX(1);
  adxl.setActivityY(1);
  adxl.setActivityZ(0);

  //look of inactivity movement on this axes - 1 == on; 0 == off
  adxl.setInactivityX(1);
  adxl.setInactivityY(1);
  adxl.setInactivityZ(0);

  //look of tap movement on this axes - 1 == on; 0 == off
  adxl.setTapDetectionOnX(0);
  adxl.setTapDetectionOnY(1);
  adxl.setTapDetectionOnZ(0);

  //set values for what is a tap, and what is a double tap (0-255)
  adxl.setTapThreshold(22); //62.5mg per increment
  adxl.setTapDuration(6); //625μs per increment
  adxl.setDoubleTapLatency(80); //1.25ms per increment
  adxl.setDoubleTapWindow(200); //1.25ms per increment

  //set values for what is considered freefall (0-255)
  adxl.setFreeFallThreshold(7);  //(5 - 9) recommended - 62.5mg per
increment
  adxl.setFreeFallDuration(45);  //(20 - 70) recommended - 5ms per
increment

  //setting all interupts to take place on int pin 1
  //I had issues with int pin 2, was unable to reset it
  adxl.setInterruptMapping( ADXL345_INT_SINGLE_TAP_BIT,  ADXL345_INT1_PIN
);
  adxl.setInterruptMapping( ADXL345_INT_DOUBLE_TAP_BIT,  ADXL345_INT1_PIN
);
  adxl.setInterruptMapping( ADXL345_INT_FREE_FALL_BIT, ADXL345_INT1_PIN );
  adxl.setInterruptMapping( ADXL345_INT_ACTIVITY_BIT, ADXL345_INT1_PIN );
  adxl.setInterruptMapping( ADXL345_INT_INACTIVITY_BIT,  ADXL345_INT1_PIN
);

  //register interupt actions - 1 == on; 0 == off
  adxl.setInterrupt( ADXL345_INT_SINGLE_TAP_BIT, 1);
  adxl.setInterrupt( ADXL345_INT_DOUBLE_TAP_BIT, 0);
  adxl.setInterrupt( ADXL345_INT_FREE_FALL_BIT, 0);
  adxl.setInterrupt( ADXL345_INT_ACTIVITY_BIT, 0);
  adxl.setInterrupt( ADXL345_INT_INACTIVITY_BIT, 0);

}

//GENERAL SETUP
void setup()
{
  //setup all the main components
  setup_TIMER();
  setup_GPS();
```

```
  setup_pulseOx();


  Serial.begin(MAIN_SERIAL_BAUD_RATE);

  //setup the two digital sensors with a pause in between for safety's
sake
  setup_imu();
  delay(100);
  setup_skin_temp();

}

//MAIN LOOP VARIABLES
int sensor_value = 0;

//stores the raw x y z values that comes straight from the accelerometer
int ORIENTATION_ANGLES_RAW[3]; // raw yaw pitch roll
//stores the angular values that are computed from the raw values
float ORIENTATION_ANGLES[3]; // yaw pitch roll
//stores the current state of orientation as an integer
float ORIENTATION_STATE = -1;
//a filter to make sure that we only change the orientation state
//after we have detected the same change for several time steps
int ORIENTATION_STATE_COUNTER = 0;

//For all variables of the form X_value and X_state
//X_value represents the most recent value obtained from the sensor X
//X_state is changed to X_value when the X_state differs by X_value
//by a specified threshold
//when X_state changes the value of X_state is transmitted
//This prevents us from transmitting all the time do to minor fluctations
from the sensors

String latitude_value = "";
String latitude_state = "";
String longitude_value = "";
String longitude_state = "";

float pulseBPM_value = 0;
float pulseBPM_state = 0;

float humidity_power_voltage = 3.3;
float humidity_value = 0;
float humidity_state = 0;

float skin_temperature_value = 0;
float skin_temperature_state = 0;

float ext_temperature_value = 0;
float ext_temperature_state = 0;

//Buffer storing the lines read from the GPS module
String GPS_OUTPUT_BUFFER = "";
String GPS_TEMP_BUFFER = "";
char SENTENCE_HEADER[7] = "$GPRMC";
int SENTENCE_HEADER_FIND_ITER = 0;
```

```
//GPS PARSING
void parseGPS(String * latitude, String * longitude)
{
  int comma_iter = 0;
  boolean start_grab = false;
  String grab_str = "";


  boolean full_line = false;
  String latitude_temp;
  String longitude_temp;

  //GPS_OUTPUT_BUFFER                                                          =
"$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,,*10";

  int str_len = GPS_OUTPUT_BUFFER.length();
  for(int i = 0; i < str_len; ++i)
  {
     char c = GPS_OUTPUT_BUFFER[i];

     if (c == ',')
     {
       if (!start_grab)
       {
         if (comma_iter == 1 || comma_iter == 2 || comma_iter == 4)
           start_grab = true;

         comma_iter++;
         if (comma_iter > 6)
           full_line = true;
       }
       else
       {
         if (comma_iter == 2 || comma_iter == 3 || comma_iter == 5)
         {
           /*
           if (comma_iter == 2) Serial.print("status ");
           if (comma_iter == 3) Serial.print("latitude ");
           if (comma_iter == 5) Serial.print("longitude ");
           Serial.println(grab_str);
           */

           if (comma_iter == 2 && grab_str != "A")
              return;
           if (comma_iter == 3 || comma_iter == 5)
           {
             double parsed_val = atof(&grab_str[0]);

             if (parsed_val == 0.0)
               return;
             if (comma_iter == 3)
               latitude_temp = grab_str;
             if (comma_iter == 5)
               longitude_temp = grab_str;
           }
         }
```

```
          start_grab = false;
          grab_str = "";
          i--;
        }
      }
      else
        if (start_grab)
          grab_str += c;
  }

  if (full_line)
  {
    *latitude = latitude_temp;
    *longitude = longitude_temp;
  }
}

void getGPSInfo()
{
  while (GPSSerial.available() > 0)
  {
    char new_char = (char)GPSSerial.read();

    if (SENTENCE_HEADER_FIND_ITER <= 5)
    {
      if (new_char == SENTENCE_HEADER[SENTENCE_HEADER_FIND_ITER])
        SENTENCE_HEADER_FIND_ITER++;
      else
        SENTENCE_HEADER_FIND_ITER = 0;
    }
    else
    {
      if (new_char == '\n' || new_char == '$')
      {
        GPS_TEMP_BUFFER += '\n';

        SENTENCE_HEADER_FIND_ITER = 0;
        GPS_OUTPUT_BUFFER = GPS_TEMP_BUFFER;
        GPS_TEMP_BUFFER = "";

        if (new_char == '$')
          SENTENCE_HEADER_FIND_ITER++;
      }
      else
        GPS_TEMP_BUFFER += new_char;
    }
  }
}

//SKIN TEMP CALCULATION
//Uses I2C protocol which I have very limited knowledge of
float computeSkinTemperature()
{
  int dev = 0x5A << 1;
  int data_low = 0;
  int data_high = 0;
```

```
  int pec = 0;

  i2c_start_wait(dev + I2C_WRITE);
  i2c_write(0x07);

  // read
  i2c_rep_start(dev + I2C_READ);
  data_low = i2c_readAck(); //Read 1 byte and then send ack
  data_high = i2c_readAck(); //Read 1 byte and then send ack
  pec = i2c_readNak();
  i2c_stop();


  //This converts high and low bytes together and processes temperature,
MSB is a error bit and is ignored for temps
  double tempFactor = 0.02; // 0.02 degrees per LSB (measurement
resolution of the MLX90614)
  double tempData = 0x0000; // zero out the data
  int frac; // data past the decimal point

  // This masks off the error bit of the high byte, then moves it left 8
bits and adds the low byte.
  tempData = (double)(((data_high & 0x007F) << 8) + data_low);

  tempData = (tempData * tempFactor) - 0.01;

  float celcius = tempData - 273.15;
  float fahrenheit = (celcius * 1.8) + 32;

  return celcius;
}

//Called by interrupt whenever skin temperature is to be sampled
void sample_skin_temp()
{
  //Skin Temperature Sensor - Sensor 3
  skin_temperature_value = computeSkinTemperature();
  sendPacketIfChanged(&skin_temperature_state, 0, skin_temperature_value,
SKIN_TEMP_SENSOR_NUMBER, 2, 0);
}

//HUMIDITY CALCULATION
float computeHumidity(int rawValue, float temperature)
{
  float Vout = discreteToVolts(rawValue);
  return (Vout - 0.958f) / 0.0307f;
}

//EXTERNAL TEMPERATURE CALCULATION
float computeTemperature(int rawValue)
{
  return discreteToVolts(rawValue) * 100.0;
}

//ORIENTATION CALCULATION
//Converts raw accelerometer readings which are the value xRead, yRead and
zRead
```

```
//into angular values and assigns them to the variables xOrientationAngle, yOrientationAngle and zOrientationAngle
void computeAccelerometerAngles(int xRead, int yRead, int zRead, float * xOrientationAngle, float * yOrientationAngle, float * zOrientationAngle)
{
  //convert read values to degrees -90 to 90 - Needed for atan2
  int xAng = map(xRead, -256, 256, -90, 90);
  int yAng = map(yRead, -256, 256, -90, 90);
  int zAng = map(zRead, -256, 256, -90, 90);

  //We are then converting the radians to degrees
  *xOrientationAngle = xAng;
  *yOrientationAngle = yAng;
  *zOrientationAngle = zAng;
}

//Converts the orientation angles into an orientation status of standing up, flat on back and flat on face
//If the frequency of taps, given by walking average, exceeds a threshold, then the orientation is switched to walking
//regardless of the angles
float computeOrientationState(float xOrientationAngle, float yOrientationAngle, float zOrientationAngle, float walkingAVG)
{
  //Presently we only look at the yOrientationAngle to compute Orientantion
  //First, we make the sure the value is between 0 and 360 degrees
  float value = fmod(yOrientationAngle, 360.0f);

  float value_to_return;

  //Then we threshold it to see what orientation it should be
  if (abs(value) < ANGULAR_ORIENTATION_THRESHOLD)
    value_to_return = STANDING;
  else if (value >= ANGULAR_ORIENTATION_THRESHOLD)
    value_to_return = FLAT_ON_FACE;
  else
    value_to_return = FLAT_ON_BACK;

  if (walkingAVG > WALKING_FREQ_THRESHOLD)
    value_to_return = WALKING;

  return value_to_return;
}

//UTILITY FUNCTIONS
float discreteToVolts(int rawValue)
{
  return 5.0 * (rawValue / 1023.0);
}

//MAIN LOOP
void loop()
{
  meetAndroid.receive();
```

```
  //Orientation Sensor - Sensor 0
  byte interrupts = adxl.getInterruptSource();

  //Check if we got an interrupt that the accel detected a tap. This
corresponds to a footfall or cane hitting the ground when walking
  //If so, send a 1 to a moving average filter
  //If not, send a 0
  if(adxl.triggered(interrupts, ADXL345_SINGLE_TAP))
    walkingFilter.addValue(1);
  else
    walkingFilter.addValue(0);

  //Read the raw values from the accel
  adxl.readAccel(&ORIENTATION_ANGLES_RAW[0],    &ORIENTATION_ANGLES_RAW[1],
&ORIENTATION_ANGLES_RAW[2]);
  //Convert them to angular values
  computeAccelerometerAngles(ORIENTATION_ANGLES_RAW[0],
ORIENTATION_ANGLES_RAW[1],                      ORIENTATION_ANGLES_RAW[2],
&ORIENTATION_ANGLES[0], &ORIENTATION_ANGLES[1], &ORIENTATION_ANGLES[2]);
  //Convert the angular values combined with the walking filter's average
to an orientation state
  sendPacketIfChanged(&ORIENTATION_STATE,       &ORIENTATION_STATE_COUNTER,
computeOrientationState(ORIENTATION_ANGLES[0],       ORIENTATION_ANGLES[1],
ORIENTATION_ANGLES[2],                         walkingFilter.getAverage()),
ORIENTATION_SENSOR_NUMBER, 0, ORIENTATION_STATE_CHANGE_COUNTER_TOLERANCE);

  //External Temperature Sensor - Sensor 1
  sensor_value = analogRead(EXT_TEMPERATURE_SENSOR_PIN);
  ext_temperature_value = computeTemperature(sensor_value);
  ExternalTempMovingAverage.addValue(ext_temperature_value);
  sendPacketIfChanged(&ext_temperature_state,                             0,
ExternalTempMovingAverage.getAverage(),           EXT_TEMP_SENSOR_NUMBER,
EXT_TEMP_CHANGE_TOLERANCE, 0);

  //Humidity Sensor - Sensor 2
  sensor_value = analogRead(HUMIDITY_SENSOR_PIN);
  humidity_value              =           computeHumidity(sensor_value,
ExternalTempMovingAverage.getAverage());
  HumidityMovingAverage.addValue(humidity_value);
  sendPacketIfChanged(&humidity_state,                                    0,
HumidityMovingAverage.getAverage(),                HUMIDITY_SENSOR_NUMBER,
HUMIDITY_CHANGE_TOLERANCE, 0);

  //Skin Temp Sensor - Sensor 3
  pulseOxSamplerIter++;
  if (pulseOxSamplerIter % SKIN_TEMP_SAMPLE_PERIOD == 0)
  {
    skin_temperature_value = computeSkinTemperature();
    sendPacketIfChanged(&skin_temperature_state,                          0,
skin_temperature_value,                           SKIN_TEMP_SENSOR_NUMBER,
SKIN_TEMP_CHANGE_TOLERANCE, 0);
  }

  //GPS Sensor - Sensor 4 + 5
  getGPSInfo();
  if (GPS_OUTPUT_BUFFER != "")
  {
```

```
    parseGPS(&latitude_value, &longitude_value);
    GPS_OUTPUT_BUFFER = "";
  }

  sendPacketIfChangedSTRING(&latitude_state,                latitude_value,
LONGITUDE_SENSOR_NUMBER, GPS_CHANGE_TOLERANCE);
  sendPacketIfChangedSTRING(&longitude_state,              longitude_value,
LATITUDE_SENSOR_NUMBER, GPS_CHANGE_TOLERANCE);

  //PULSEOX - Sensor 6
  if (pulseOxSamplerIter % BPM_COMPUTATION_PERIOD == 0)
  {
    pulseBPM_value = pulseOxHeartBeatWindow.calculateBPM();
    sendPacketIfChanged(&pulseBPM_state,         0,         pulseBPM_value,
HEARTRATE_SENSOR_NUMBER, HEARTRATE_CHANGE_TOLERANCE, 0);
  }

  //Halt everything until enough time has elapsed since the last pulse
sample
  //This effectively limits the speed of the program, but it works
  while(millis()      -      pulseOxHeartBeatWindow.last_sample_time      <
PULSE_SAMPLE_PERIOD)
  {
  }
  samplePulseOx();
}

void sendPacketIfChangedSTRING(String * stateVariable, String newState,
int whichSensor, float tol)
{
  double parsed_val = atof(&(*stateVariable)[0]);
  double parsed_val_2 = atof(&newState[0]);
  if (abs(parsed_val-parsed_val_2) > tol)
  {
   *stateVariable = newState;
   sendPacket(newState, whichSensor);
  }
}

//A master function for sending information only when there is a change
//Most of the time this is used to only send if the current value and the
new value differ by a threshold
//But if stateChangeCounter and stateChangeCounterTolerance are
void sendPacketIfChanged(float * stateVariable, int * stateChangeCounter,
float    newState,    int    whichSensor,    float    tolerance,    int
stateChangeCounterTolerance)
{
  if (abs(*stateVariable - newState) > tolerance)
  {

    //Only if stateChangeCounter is not null
    //Do we do this complex process
    //Presently only the orientation sensor uses this
    if (stateChangeCounter != 0)
      (*stateChangeCounter)++;
```

```
    if   (stateChangeCounterTolerance   ==   0   ||   *stateChangeCounter   >
stateChangeCounterTolerance)
    {
      *stateVariable = newState;

      if (stateChangeCounter != 0)
        *stateChangeCounter = 0;

      sendPacket(newState, whichSensor);
    }
  }
  else
    if (stateChangeCounter != 0)
      *stateChangeCounter = 0;
}



#if CONNECT_TO_ANDROID == false

  void sendPacket(float value, int whichSensor)
  {
    Serial.print('[');
    Serial.print(whichSensor);
    Serial.print(':');
    Serial.print(value);
    Serial.print('@');
    Serial.print(millis());
    Serial.print(']');
  }
 void sendPacket(String value, int whichSensor)
  {
    Serial.print('[');
    Serial.print(whichSensor);
    Serial.print(':');
    Serial.print(value);
    Serial.print('@');
    Serial.print(millis());
    Serial.print(']');
  }

#else
   void sendPacket(String value, int whichSensor)
  {
    meetAndroid.send('[');
    meetAndroid.send(whichSensor);
    meetAndroid.send(':');
    meetAndroid.send(&value[0]);
    meetAndroid.send('@');
    meetAndroid.send(millis());
    meetAndroid.send(']');
  }
  void sendPacket(float value, int whichSensor)
  {
    meetAndroid.send('[');
    meetAndroid.send(whichSensor);
    meetAndroid.send(':');
```

```
    meetAndroid.send(value);
    meetAndroid.send('@');
    meetAndroid.send(millis());
    meetAndroid.send(']');
  }

#endif
```

18. APPENDIX G: SENSOR UPLOADER CODE

Code repository can be found at:
https://www.dropbox.com/sh/85yjcqbvb2fw9wi/_XVptTeRL6

```
package com.sensortransmissionandroid;

import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.util.ArrayList;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import android.util.Log;

//A simple class that acts as a bridge to the mySQL database

public class processedDataUploader {
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    PreparedStatement pst = null;

    //Login information for the mySQL connection
    String url = "jdbc:mysql://134.173.38.36:3306/testdb2";
    String user = "from_pc";
    String password = "gloclinic";

    //We maintain references to all the sensors
    ArrayList<Sensor> mySensors;

    private void sendQuery(String query_str)
    {
        ArrayList<NameValuePair> nameValuePairs = new
ArrayList<NameValuePair>();
        nameValuePairs.add(new BasicNameValuePair("query", query_str));

        //http post
        try
        {
            HttpClient httpclient = new DefaultHttpClient();
```

```
            HttpPost httppost = new HttpPost("http://global-clinic-
test.zxq.net/test.php");
            httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

            HttpResponse response = httpclient.execute(httppost);
            //HttpEntity entity = response.getEntity();
            //InputStream is = entity.getContent();
        }
        catch(Exception e)
        {
            Log.e("log_tag", "Error in http connection "+e.toString());
        }
    }

    public processedDataUploader(ArrayList<Sensor> mySensors_)
    {
        mySensors = mySensors_;
    }

    //Connects to the DB
    void initializeConnection()
    {
        /*
        try {
            con = DriverManager.getConnection(url, user, password);
        }
        catch (SQLException ex)
        {
            System.out.println("err establishing connection " +
ex.getMessage());
        }
        */
    }

    //Closes the connection to the DB
    void closeConnection()
    {
        /*
        try {
            if (rs != null) {
                rs.close();
            }
            if (st != null) {
                st.close();
            }
            if (con != null) {
                con.close();
            }

        } catch (SQLException ex) {
            System.out.println("err 2 " + ex.getMessage());
        }
        */
    }

    //Given a sensor number and a value for that sensor, adds it to the
database
```

```
    void upload(int theSensorNumber, Object processedValue, String
username)
    {
        //String query_str = query_str = "INSERT INTO "  + "SENSOR_" +
theSensorNumber + " (val) VALUES(" + processedValue.toString() + ")";
        String query_str = query_str = "INSERT INTO SENSOR (userid,
deviceid, sensorid, val) VALUES('" + username + "', '0000', "+
theSensorNumber +"," + processedValue.toString() + ")";

        sendQuery(query_str);
    }

    //Called when the tables for the database need to be created
    void setupTables()
    {

        /*
        try
        {
            //drop and recreate database
            st = con.createStatement();
            st.executeUpdate("drop database testdb2");
            st.executeUpdate("create database testdb2");
            st.executeUpdate("use testdb2");
        }
        catch (SQLException ex)
        {
            System.out.println("Error destroying database " +
ex.getMessage());
        }

        try
        {
            //makes one sensor table
            st.executeUpdate("CREATE TABLE SENSOR ( id INT NOT NULL
AUTO_INCREMENT, PRIMARY KEY(id), userid VARCHAR(50), deviceid
VARCHAR(255), sensorid INT, val FLOAT, made_on TIMESTAMP)");
            //makes one user table
            st.executeUpdate("CREATE TABLE USER ( userid VARCHAR(50) NOT
NULL, PRIMARY KEY(userid), password VARCHAR(50), deviceid VARCHAR(255),
made_on TIMESTAMP)");
            st.executeUpdate("INSERT INTO USER ( userid, password,
deviceid) VALUES ('samantha', 'gloclinic', '0000')");
            st.executeUpdate("INSERT INTO USER ( userid, password,
deviceid) VALUES ('noah', 'gloclinic', '0001')");
            //makes viewables table
            st.executeUpdate("CREATE TABLE VIEWABLE ( id INT NOT NULL
AUTO_INCREMENT, PRIMARY KEY(id), viewer VARCHAR(50), viewed VARCHAR(50),
made_on TIMESTAMP)");
            st.executeUpdate("INSERT INTO VIEWABLE ( viewer, viewed)
VALUES ('samantha', 'samantha')");
            st.executeUpdate("INSERT INTO VIEWABLE ( viewer, viewed)
VALUES ('samantha', 'noah')");
            /*
            for(Sensor sensor : mySensors)
            {
```

```
                st.executeUpdate("CREATE TABLE " + "SENSOR_" +
sensor.number + " ( id INT NOT NULL AUTO_INCREMENT, PRIMARY KEY(id), val
FLOAT, made_on TIMESTAMP)");
            }
            */
          //  System.out.println("Tables Built");
        //}
        //catch (SQLException ex)
       // {
        //  System.out.println("Error making tables " + ex.getMessage());
       // }

    }
}
```

19. APPENDIX H: DATA VIEWER CODE

19.1. HEADER.PHP

```
<?php
function print_header($option = 0, $page = "")
{
?>
     <head>
          <meta charset="utf-8">
          <link  rel="stylesheet"  href="../../../resources/style/demo.css"
media="screen">
          <link rel="stylesheet" href="style.css" media="screen">
          <link                                             rel="stylesheet"
href="http://ajax.googleapis.com/ajax/libs/dojo/1.7.1/dijit/themes/claro/c
laro.css" media="screen">
          <!-- load dojo and provide config via data attribute -->
          <script
src="http://ajax.googleapis.com/ajax/libs/dojo/1.7.1/dojo/dojo.js"
                    data-dojo-config="async: true, parseOnLoad: true">
          </script>
          <script                                      type="text/javascript"
src="../javascript/validate.js"></script>
          <script                                      type="text/javascript"
src="../javascript/util.js"></script>
          <script>
               require(["dijit/layout/BorderContainer",
"dijit/layout/TabContainer", "dijit/layout/ContentPane", "dojo/parser"]);
          </script>
     </head>
<?php
/*
     $m = new msg();
*/
     // login,singup,forgot password,logout
     @session_start();

?>
     <body class="claro">
     <center>
     <table>    <!--  Start page -->
     <tr>
          <td colspan="3">
          <a         href="../index.php"><img          src="../img/logo.gif"
border=0></a><br>
          <table style="align:left; width: 600px;">   <!-- Start Menu bar -
->
               <tr valign="top">
                    <td valign="center">
                         <table class="mainmenu">
                              <tr>
                                   <td><a    href='../mainData.php'><?php    if
($page=="mainData") echo "<font color=0076A3>View Data</font>"; else echo
"View Data"; ?></a></td>
```

```
                                <td> &bull; </td>
                                <td><a href='../manageDevices.php'><?php if
($page=="manageDevices") echo "<font color=0076A3>Manage Devices</font>";
else echo "Manage Devices"; ?></a></td>
                            </tr>
                        </table>
                    </td>
                    <td align="right">

<?php

                    // display login form
                    if($_SESSION["id"] == null)
                    {
                        print '<a href="../account.php?fcn=signup"
style="color:blue;">Join now</a>   ';
                        print '<a href="../account.php?fcn=login"
style="color:blue;">Login</a>   ';
                    }
                    // after login
                    else
                    {
                        print '<a href="../lib/login.class.php?ac=logout"
style="color:blue;">Logout</a>';
                    }

?>
                    </td>
                </tr>
            </table> <!-- End Menu table -->
            </td>
        </tr>
        <tr><td colspan="2"><br></td></tr>
        </table>
<?php
}
?>
```

19.2. LOGIN.CLASS.PHP

```
<?php

class login
{
    function __construct()   // php 5 constructor
    {
        $databasehost = "localhost";
        $databasename = "global-clinic-test_zxq_testdb2";
        $databaseusername ="655453_root";
        $databasepassword = "gloclinic";
        $con =
mysql_connect($databasehost,$databaseusername,$databasepassword) or
die(mysql_error());
        mysql_select_db($databasename) or die(mysql_error());
        mysql_select_db("global-clinic-test_zxq_testdb2");
    }
```

```
    function login()   // php 4 constructor
    {
         if(version_compare(PHP_VERSION,"5.0.0","<"))
         {
            $this->__construct();
            register_shutdown_function(array($this,"__destruct"));
        }
    }

    function __destruct()  // php 4 does not have destructor
    {
    }

    function action()
    {
         session_start();

         $ac = ($_REQUEST['ac'] == null ? '' : $_REQUEST['ac']);

         if ($ac == "new")
         {
              // create new user in database
              $user = array();
              $user['userid']  =  ($_REQUEST['userName']  ==  null  ?  ''  :
$_REQUEST['userName']);
              $user['password']  =  ($_REQUEST['password']  ==  null  ?  ''  :
$_REQUEST['password']);
              $user['fname']  =  ($_REQUEST['firstName']  ==  null  ?  ''  :
$_REQUEST['firstName']);
              $user['lname']  =  ($_REQUEST['lastName']  ==  null  ?  ''  :
$_REQUEST['lastName']);
              $user['email']  =  ($_REQUEST['email']  ==  null  ?  ''  :
$_REQUEST['email']);
              $qry  =  "INSERT  INTO  USER  (userid,  password,  fname,  lname,
email) ".
                        "VALUES
('".$user['userid']."','".$user['password']."', '".$user['fname']."','".
                        $user['lname']."','".$user['email']."')";
              $result = mysql_query($qry);
              $_SESSION['id'] = $user['userid'];
              header ('Location: /manageDevices.php');
         }
         else if ($ac == "adddevice")
         {
              $device = array();
              $device['viewer'] = $_SESSION["id"];
              $device['viewed']=  ($_REQUEST['deviceID']  ==  null  ?  ''  :
$_REQUEST['deviceID']);
              $device['viewed_name']= ($_REQUEST['deviceName'] == null ? ''
: $_REQUEST['deviceName']);
              $qry  =  "INSERT  INTO  VIEWABLE  (viewer,  viewed,  viewed_name)
VALUES          ('".$device['viewer']."',          '".$device['viewed']."',
'".$device['viewed_name']."')";
              $result = mysql_query($qry);
              header ('Location: /manageDevices.php');
         }
```

```
         // login
         else if ($ac == "login")
         {
              $uname = ($_REQUEST['id'] == null ? '' : $_REQUEST['id']);
              $pwd = ($_REQUEST['pwd'] == null ? '' : $_REQUEST['pwd']);
              // validate against database

              $qry = "select * from USER where userid='".$uname."' and
password = '".$pwd."'";
              $result=mysql_query($qry);

              if (mysql_num_rows($result) > 0)
              {
                   $user = mysql_fetch_assoc($result);
                   $_SESSION['id'] = $user['userid'];
                   header ('Location: /mainData.php');
              }
              else
                   header ('Location: /account.php?fcn=login&err=1');
         }
         //delete device
         else if ($ac == "delDevice")
         {
              $viewer = ($_REQUEST['viewer'] == null ? '' :
$_REQUEST['viewer']);
              $viewed = ($_REQUEST['viewed'] == null ? '' :
$_REQUEST['viewed']);
              // validate against database

              $qry = "DELETE FROM VIEWABLE WHERE viewer='".$viewer."' AND
viewed='".$viewed."'";
              $result=mysql_query($qry);

              header ('Location: /manageDevices.php');
         }
         // logout
         else if ($ac == "logout")
         {
              //$s = new iatSession();
              //$s->destroy();
              session_start();
              session_destroy();
              header("Location: /index.php");
         }

    }

}

?>
```

19.3. TEST.PHP

```
<?php

$databasehost = "localhost";
```

```
$databasename = "global-clinic-test_zxq_testdb2";
$databaseusername ="655453_root";
$databasepassword = "gloclinic";

$con = mysql_connect($databasehost,$databaseusername,$databasepassword) or
die(mysql_error());
mysql_select_db($databasename) or die(mysql_error());
mysql_select_db("global-clinic-test_zxq_testdb2");

$q=mysql_query($_REQUEST['query']);
?>
```

19.4. MANAGEDEVICES.PHP

```
<?php
require_once("./header.php");
$page = "manageDevices";
print_header();

$databasehost = "localhost";
$databasename = "global-clinic-test_zxq_testdb2";
$databaseusername ="655453_root";
$databasepassword = "gloclinic";
$con = mysql_connect($databasehost,$databaseusername,$databasepassword) or
die(mysql_error());
mysql_select_db($databasename) or die(mysql_error());
mysql_select_db("global-clinic-test_zxq_testdb2");

$currentuser = $_SESSION["id"];
?>
<script>
function check()
{
    if          (document.getElementById("deviceID").value==""          ||
document.getElementById("deviceName").value=="")
         return false;
    else return true;
}
</script>
<h3>Manage Devices</h3><br>

    <div class="centerPanel" style="height:800px; width:400px">
    <?php
         if ($currentuser){
              echo   '<form   action="lib/login.class.php"   method="post"
name="frm1" id="frm1">
                        <input type="hidden" name="ac" value="adddevice">
                        <div align="left">
                             <p>If you would like to view the output of a
device, enter its ID below.</p>
                             <table>
                                  <tr>
                                       <td>
                                            <label   for="deviceID">Device   ID:
 </label>
                                       </td><td>
```

```
                                        <input type="text" name="deviceID"
id="deviceID" value="" />
                                    </td>
                                </tr><tr>
                                    <td>
                                        <label for="deviceID">Device Name:
 </label>
                                    </td><td>
                                        <input                  type="text"
name="deviceName" id="deviceName" value="" />
                                    </td>
                                </tr>
                            </table>
                        </div>
                        <input      type="submit"      value="Add      Device"
onClick="return check();"/>
                    </form>
                    <br><hr><br>
                    <table width=100%>
                        <tr>
                            <th         align=left>Device         ID</th><th
align=left>Device Name</th><th align=left>Delete</th>
                        </tr>';

              $result=mysql_query("SELECT    *    FROM    VIEWABLE    WHERE
viewer='".$currentuser."'");
              while($row = mysql_fetch_array($result))
              {
                  echo '<tr>';
                  echo
'<td>'.$row['viewed'].'</td><td>'.$row['viewed_name'].'</td><td><a
onclick="deleteMe(\''.$row['viewed'].'\',
\''.$currentuser.'\')">Delete</a></td>';
                  echo '</tr>';
              }
          }
          else
                      echo 'You are not currently logged in. You must log
in to add devices.';
              ?>
     </div>

<?php
require_once("./footer.php");
print_footer();
?>
```

19.5. MAINDATA.PHP

```
<?php
require_once("./header.php");
print_header();
$databasehost = "localhost";
$databasename = "global-clinic-test_zxq_testdb2";
$databaseusername ="655453_root";
$databasepassword = "gloclinic";
```

```
$con = mysql_connect($databasehost,$databaseusername,$databasepassword) or
die(mysql_error());
mysql_select_db($databasename) or die(mysql_error());
mysql_select_db("global-clinic-test_zxq_testdb2");

$page = "mainData";
$currentuser = $_SESSION["id"];
$vieweduser = $_GET["sid"];

$sensorid = array( //the values in this array have no functionality and
are for reference only. The number of entries in the array is used.
    '0'    => "Orientation",
    '1'  => "Temperature",
    '2'  => "Humidity",
     '4'    => "Heart Rate",
    '3'  => "Body Temperature",
    '5'  => "Pedometer",
);
$imgdimension = array(
     'x' => "100px",
     'y' => "173px",
     'yoffset'=> "30px",
);
?>

<div id="appLayout" class="demoLayout" data-dojo-
type="dijit.layout.BorderContainer" data-dojo-props="design: 'headline'"
style="height:800px; width:600px">
                  <div class="centerPanel" data-dojo-
type="dijit.layout.TabContainer" data-dojo-props="region: 'center',
tabPosition: 'top'">
                       <?php
                            $result=mysql_query("SELECT * FROM VIEWABLE
WHERE viewer='".$currentuser."'");

                            while($row = mysql_fetch_array($result)){
                                $selected = 'false';
                                if ($vieweduser ==$row['viewed']) $selected
= 'true';
                                echo '<div data-dojo-
type="dijit.layout.ContentPane" id="'.$row['viewed'].'"
title="'.$row['viewed_name'].'" selected="'.$selected.'"><table><tr><td>';
                                     for ($count = 0; $count <
count($sensorid); $count ++)
                                     {
                                          $data=mysql_query("SELECT * FROM
SENSOR WHERE userid='".$row['viewed']."' AND sensorid='".$count."' ORDER
BY made_on DESC");
                                          echo "<a
href='detailData.php?sid=".$count."&uid=".$row['viewed']."'>";
                                          echo "<div style='background-
image:url(\"/img/sensor/".$count.".jpg\"); height:".$imgdimension['x'].";
width:".$imgdimension['y']."'><table align='right'><tr><td
width=40></td><td>";
                                          echo "<h1 style='margin-
top:".$imgdimension['yoffset']."'>";
```

```
                                        if($datum                         =
mysql_fetch_array($data))
                                            echo $datum['val'];
                                        else
                                            echo "?";
                                        echo
"  </h1></td></tr></table></div></a>";
                                        if ($count==2)
                                            echo                "</td><td
width=40></td><td>";
                                    }
                                echo '</td></tr></table>';
                                $latitude = '138.324137';
                                $longitude = '36.28801';
                                $url                                      =
'http://maps.google.com/?t=h&amp;ie=UTF8&amp;q='.$longitude.','.$latitude.
'&amp;spn=0,0&amp;ll='.$longitude.','.$latitude.'&amp;z=14&amp;output=embe
d';
                                // echo '
                                    // <script>

                                        //                          function
createIframe'.$row['viewed'].'(){
                                          //
document.getElementById("iframefor'.$row['viewed'].'").innerHTML="<iframe
width=\"450\"     height=\"250\"     frameborder=\"0\"     scrolling=\"no\"
marginheight=\"0\" marginwidth=\"0\" src=\"'.$url.'\"></iframe>";
                                        // };
                                        // if (window.addEventListener)
                                        //  window.addEventListener("load",
createIframe'.$row['viewed'].', false);
                                        // else if (window.attachEvent)
                                        //     window.attachEvent("onload",
createIframe'.$row['viewed'].');
                                        //     else     window.onload     =
createIframe'.$row['viewed'].';
                                    // </script>
                                    //                                 <div
id="iframefor'.$row['viewed'].'"></div>
                                // ';
                                echo '
                                <iframe      width="450"       height="250"
frameborder="0" scrolling="no" marginheight="0" marginwidth="0"
                                src="'.$url.'">
                                </iframe><br                            /><a
href="'.$url.'"><small>Large View</small></a>';

                                echo '</div>';

                            }
                        ?>
                    </div>
                </div>
<?php
require_once("./footer.php");
print_footer();
?>
```

19.6. DETAILDATA.PHP

```
<?php
require_once("./header.php");
require_once("./makePlot.php");
print_header();
$databasehost = "localhost";
$databasename = "global-clinic-test_zxq_testdb2";
$databaseusername ="655453_root";
$databasepassword = "gloclinic";

$con = mysql_connect($databasehost,$databaseusername,$databasepassword) or
die(mysql_error());
mysql_select_db($databasename) or die(mysql_error());
mysql_select_db("global-clinic-test_zxq_testdb2");

$currentuser = $_SESSION["id"];
$vieweduser = $_GET["uid"];
$sensor = $_GET["sid"];

$sensorid = array(
    '0'    => "Orientation",
    '1'  => "Temperature",
    '2'  => "Humidity",
     '4'    => "Heart Rate",
    '3'  => "Body Temperature",
    '5'  => "Pedometer",
);

//create the data for the plot iff the user is allowed to view it.
$result=mysql_query("SELECT        *        FROM        VIEWABLE        WHERE
viewer='".$currentuser."' AND viewed ='".$vieweduser."'");
 if (mysql_fetch_array($result)){
     print_plot($sensor, $vieweduser);
     $viewable = true;
 }
 else $viewable = false;
?>

</center>
<div          id="appLayout"          class="demoLayout"          data-dojo-
type="dijit.layout.BorderContainer"  data-dojo-props="design:  'headline'"
style="height:800px; width:600px; margin-left:auto;
margin-right:auto;">
                  <div            class="centerPanel"            data-dojo-
type="dijit.layout.TabContainer"     data-dojo-props="region:     'center',
tabPosition: 'top'">
                       <?php
                            if           ($sensor=='0')              echo
'<br><center><table><tr><td><img     src="img/fallicon.jpg">  Red
represents    a    fall.<br><img    src="img/standicon.jpg">  Blue
represents                                                  '.$vieweduser.'
standing.</td></tr></table></center><br><br><div            id="chartNode"
style="width:550px;height:200px;"></div>';
                            else       echo       '<div       id="chartNode"
style="width:550px;height:400px;"></div>';
```

```
                              if ($viewable)
                              {
                                        echo          '<div          data-dojo-
type="dijit.layout.ContentPane"                    title="'.$vieweduser.'\'s
'.$sensorid[$sensor].'">
                                                  <p>                <center><a
href="mainData.php?sid='.$vieweduser.'"><table><tr><td><img
src="/img/back.jpg"></td><td><h1>             Back             to
overview</h1></td></tr></table></a></center>';

                                   echo '</p></div>';
                              }
                              else echo '<h3>Error: User mismatch</h3>';
                         ?>

                    </div>
               </div>
<?php
require_once("./footer.php");
print_footer();
?>
```

19.7. MAKEPLOT.PHP

```
<?php

function print_plot($sensor, $vieweduser)
{
     //database connection
     $databasehost = "localhost";
     $databasename = "global-clinic-test_zxq_testdb2";
     $databaseusername ="655453_root";
     $databasepassword = "gloclinic";
     $con                                                                   =
mysql_connect($databasehost,$databaseusername,$databasepassword)         or
die(mysql_error());
     mysql_select_db($databasename) or die(mysql_error());
     mysql_select_db("global-clinic-test_zxq_testdb2");

     //getting the data
     $data=mysql_query("SELECT * FROM SENSOR WHERE userid='".$vieweduser."'
AND sensorid='".$sensor."' ORDER BY made_on");

     //defining chart type and implementing data
     if ($sensor=="0"){

          $plotdata1 = 'var plotData1 = [';
          $plotdata2 = ' var plotData2 = [';
          $plotdata3 = ' var plotData3 = [';
          $i = 0;
          while($datum   =   mysql_fetch_array($data))//hardcoded   for   demo
purposes!
          {
               $i += 1;
               if ($datum['val']=='0')
               {
```

```
                    $plotdata1 .= '{id:'.$i.', y:1}, ';
                    $plotdata2 .= '{id:'.$i.', y:0}, ';
                    $plotdata3 .= '{id:'.$i.', y:0}, ';
               }
               else if ($datum['val']=='1')
               {
                    $plotdata1 .= '{id:'.$i.', y:0}, ';
                    $plotdata2 .= '{id:'.$i.', y:1}, ';
                    $plotdata3 .= '{id:'.$i.', y:0}, ';
               }
               else
               {
                    $plotdata1 .= '{id:'.$i.', y:0}, ';
                    $plotdata2 .= '{id:'.$i.', y:0}, ';
                    $plotdata3 .= '{id:'.$i.', y:1}, ';
               }
          }
          $plotdata1 .= '];';
          $plotdata2 .= '];';
          $plotdata3 .= '];';
          echo '<script>';
          echo $plotdata1;
          echo $plotdata2;
          echo $plotdata3;
          echo '</script>';
          ?>
               <script>
                    require([
                         "dojox/charting/Chart",
                         "dojox/charting/themes/MiamiNice",
                         "dojox/charting/plot2d/StackedColumns",
                         "dojox/charting/plot2d/Markers",
                         "dojox/charting/axis2d/Default",
                         "dojo/domReady!"
                    ], function(Chart, theme) {
                         // Create the chart within it's "holding" node
                         var chart = new Chart("chartNode");
                         chart.setTheme(theme);
                         chart.addPlot("default", {
                              type: "StackedColumns",
                              markers: true,
                         });
                         chart.addAxis("y", { min: 0, max: 96,
                              majorLabels: true, majorTicks: true, majorTick:
{length:10},
                              minorLabels:       false,       minorTicks:true,
minorTick:{length:6},
                              majorTickStep:24,
                              minorTickStep:4,
                              labels: [{value: 0, text: "0:00"}, {value: 24,
text: "6:00"},
                                        {value:  48,  text:  "12:00"},  {value:
72, text: "18:00"},
                                        {value: 96, text: "0:00"}
                                        ]
                                        }); //actually the x axis
```

```
                      chart.addAxis("x", { min: 0, max: 200});//actually
the y axis
                      chart.addSeries("set1",plotData1);
                      chart.addSeries("set2",plotData2, {stroke: {color:
"red"}, fill: "red"});
                      chart.addSeries("set3",plotData3, {stroke: {color:
"yellow"}, fill: "yellow"});
                      chart.render();
                 });


             </script>

     <?php
     }
     else{
          echo '<script> var plotData = [';
          while($datum = mysql_fetch_array($data))
          {
               echo $datum['val'].', ';
          }
          echo ']; </script>';
          echo '<link rel="stylesheet"
href="/path/to/dijit/themes/claro/claro.css" media="screen">
             <script>
             require([
                   // Require the basic chart class
                  "dojox/charting/Chart",

                  // Require the theme of our choosing
                  "dojox/charting/themes/MiamiNice",

                  //  We want to plot Lines
                  "dojox/charting/plot2d/Lines",

                  // Load the Legend, Tooltip, and Magnify classes
                  "dojox/charting/widget/Legend",
                  "dojox/charting/action2d/Tooltip",
                  "dojox/charting/action2d/Magnify",

                  //  We want to use Markers
                  "dojox/charting/plot2d/Markers",

                  //  We will use default x/y axes
                  "dojox/charting/axis2d/Default",

                  // Wait until the DOM is ready
                  "dojo/domReady!"
             ], function(Chart, theme, LinesPlot, Legend, Tooltip,
Magnify) {
                  // Create the chart within its "holding" node
                  var chart = new Chart("chartNode");
                  // Set the theme
                  chart.setTheme(theme);
                  // Add the only/default plot
                  chart.addPlot("default", {
                       type: LinesPlot,
```

```
                        markers: true
                    });
                    // Add axes
                    chart.addAxis("x");
                    chart.addAxis("y", { vertical: true, fixLower: "major",
fixUpper: "major" });
                    // Add the series of data
                    chart.addSeries("data",plotData);
                    // Create the tooltip
                    var tip = new Tooltip(chart,"default");
                    // Create the magnifier
                    var mag = new Magnify(chart,"default");
                    // Render the chart!
                    chart.render();
               });</script>';
     }
}
```

19.8. ACCOUNT.PHP

```
<?php
require_once("./header.php");
print_header();

$function = $_GET['fcn'];
$error = $_GET['err'];
$currentuser = $_SESSION["id"];

$databasehost = "localhost";
$databasename = "global-clinic-test_zxq_testdb2";
$databaseusername ="655453_root";
$databasepassword = "gloclinic";
$con = mysql_connect($databasehost,$databaseusername,$databasepassword) or
die(mysql_error());
mysql_select_db($databasename) or die(mysql_error());
mysql_select_db("global-clinic-test_zxq_testdb2");

if ($error=='1')
     echo 'Login failed. Username/password combination is not in our
database.';
?>

<?php
if ($function=='login'){
     echo'
          <form action="lib/login.class.php" method="post" name="frm1"
id="frm1">
          <input type="hidden" name="ac" value="login">
          <div class="fieldmaindiv">
               <div class="fieldlt">Username:</div>
               <div class="fieldrt">
                    <input type="text" name="id" id="id"/>
               </div>
          </div>
          <div class="fieldmaindiv">
               <div class="fieldlt">Password:</div>
```

```
            <div class="fieldrt">
                <input type="password" name="pwd" id="pwd"/>
            </div>
        </div>
        <div class="fieldmaindiv">
            <div   class="fieldlt"><input   type="submit"   name="login"
id="login" value="Login"/>      
            </div>
            <div    class="fieldrt"><a     href="account.php?fcn=forgot"
style="color:blue;">Forgot password</a>
            </div>
        </div>
        </form>';
}
else if ($function=='signup'){
    ?>
        <script>
        function check()
        {
            if     (document.getElementById("userName").value==""     ||
document.getElementById("email").value==""                         ||
document.getElementById("password").value=""){
                alert ("Missing fields");
                return false;
            }
            else                                                      if
(document.getElementById("password").value!=document.getElementById("passw
ord2").value){
                alert("Password match failed");
                return false;
            }
            else                                                      if
(document.getElementById("email").value!=document.getElementById("emailCon
firm").value){
                alert("Email match failed");
                return false;
            }
            else return true;
        }
        </script>
        <div class="centerPanel" style="height:800px; width:400px">
        <div   align="left"><p   style="font-size:   x-small">*   denotes
required fields.</p></div>
        <form   action="lib/login.class.php"   method="post"   name="frm1"
id="frm1">
            <input type="hidden" name="ac" value="new">
            <div align="left">
                <label for="userName">Username*</label>
                <input    type="text"    name="userName"    id="userName"
value="" />
            </div>
            <div align="left">
                <label for="firstName">First Name</label>
                <input   type="text"   name="firstName"   id="firstName"
value="" />
            </div>
            <div align="left">
```

```
                    <label for="lastName">Last Name</label>
                    <input   type="text"   name="lastName"   id="lastName" 
value="" />
               <div align="left">
               </div>
                    <label for="email">Email Address*</label>
                    <input type="text" name="email" id="email" value="" />
               </div>
               <div align="left">
                    <label for="emailConfirm">Confirm Email*</label>
                    <input type="text" name="emailConfirm" id="emailConfirm" 
value="" />
               </div>
               <div align="left">
                    <label for="password">Password*</label>
                    <input  type="password"  name="password"  id="password" 
value="" />
               </div>
               <div align="left">
                    <label for="password2">Confirm Password*</label>
                    <input  type="password"  name="password2"  id="password2" 
value="" />
               </div>
               <input   type="submit"   value="Sign   up"   onclick="return 
check();"/>
          </form>
     </div>
<?php
}
else if ($function=='acctedit'){
     echo 'this functionality not available';
}
else if ($function=='forgot'){
     echo 'this functionality not available';
}
else if ($function=='validate'){
     echo 'this functionality not available';
}
else echo 'Error, unknown function';
?>

<?php
require_once("./footer.php");
print_footer();
?>
```

20. APPENDIX I: SENSOR TESTING INTERFACE

The Sensor Testing Interface, a Java program that runs on a PC, was developed for the calibration and debugging of new sensors. The program receives input from the microprocessor in the format specified at the end of section 7.1 and visualizes the input. The program is a modified version of an open source program called LiveGraph, which is designed to allow real-time graphing of data from multiple sensors. LiveGraph supports user controlled selection of which sensors to plot, which colors to assign the plots, and various transformations that can be applied to the plots (Figure 41).

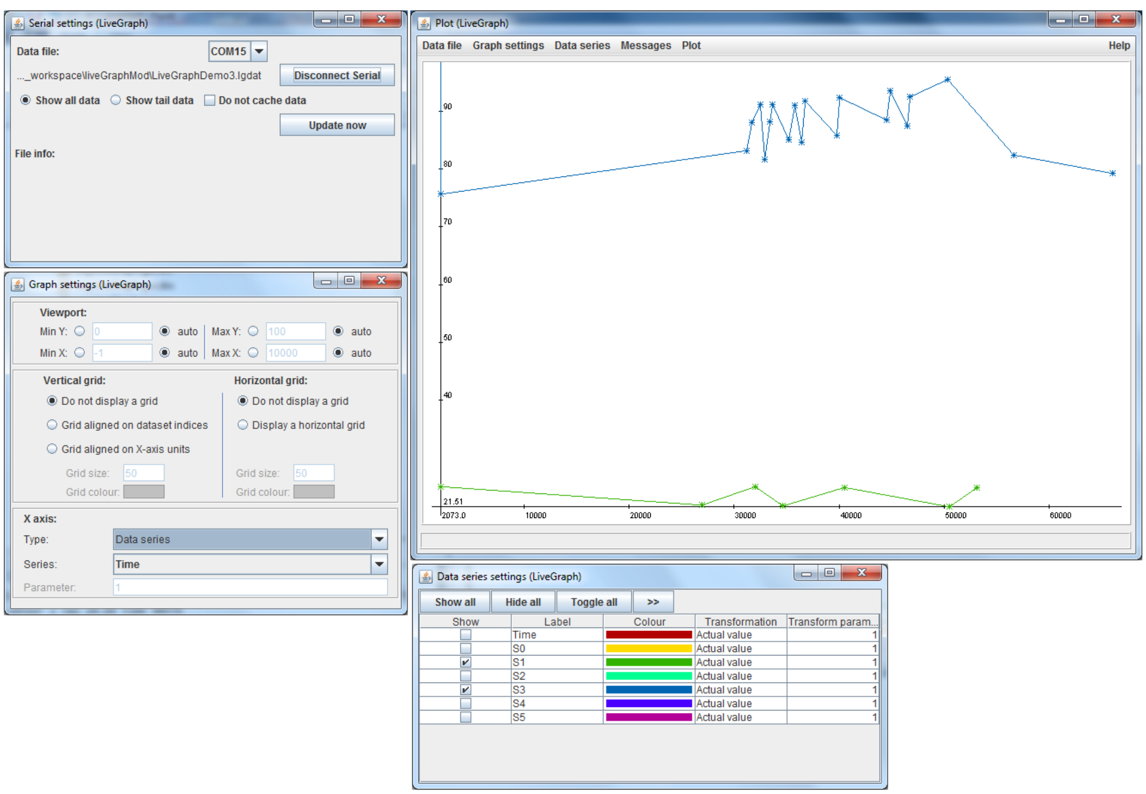


Figure 41: Sensor Testing Interface

The main modification made to LiveGraph was the reading of data from a serial port connection in the format that it is transmitted in by the microprocessor. Due to the large size of the code file, it is not copied here. The complete code can be found at: https://www.dropbox.com/sh/85yjcqbvb2fw9wi/_XVptTeRL6.

21. APPENDIX J: CANE PROTOTYPE CAD SCHEMATIC

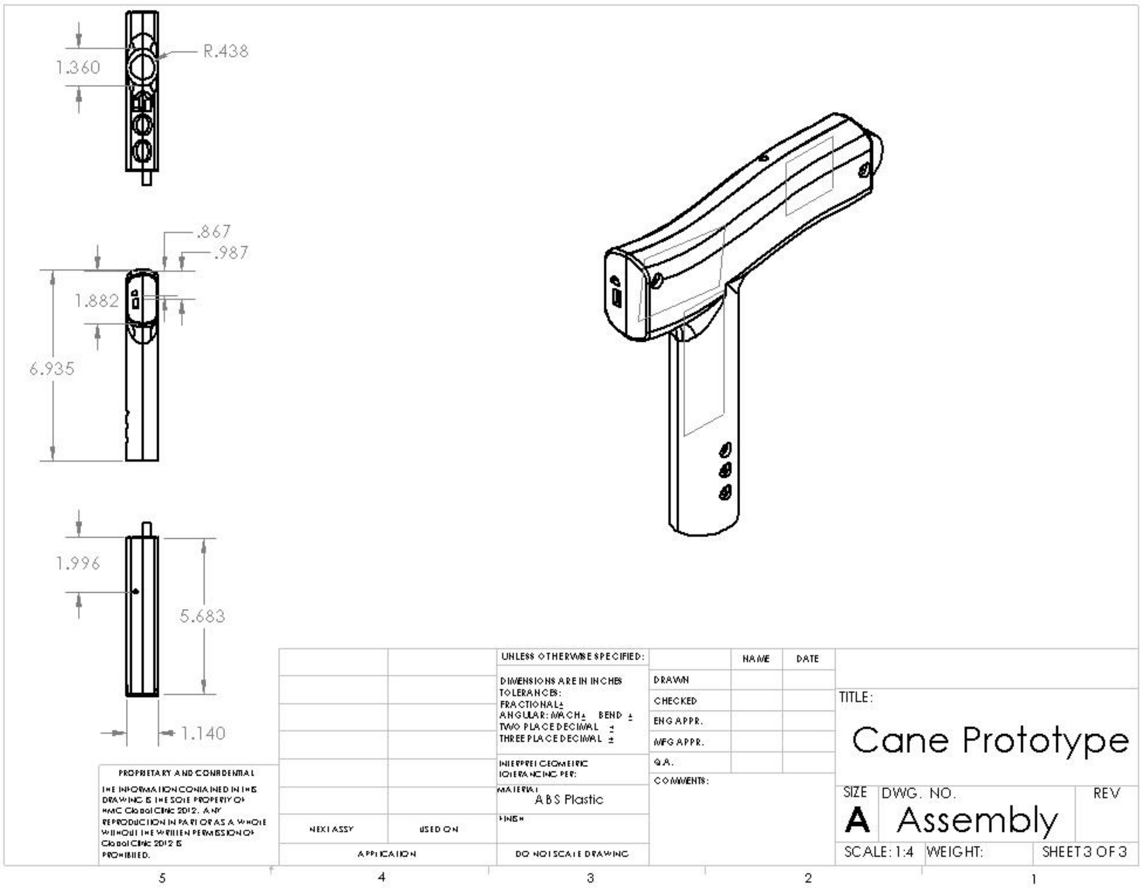


Figure 42: Assembled Schematic of Cane Prototype

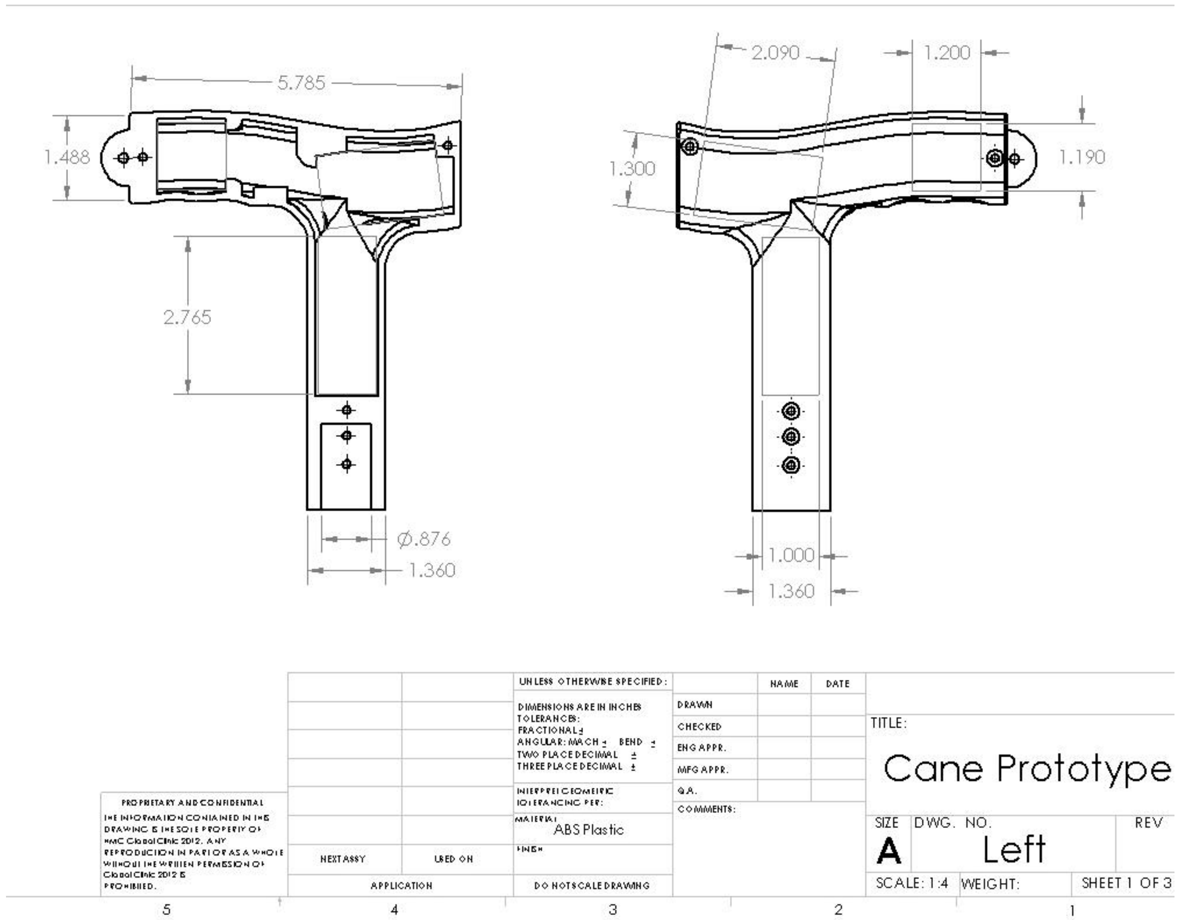


Figure 43: Left Half Portion of the Cane

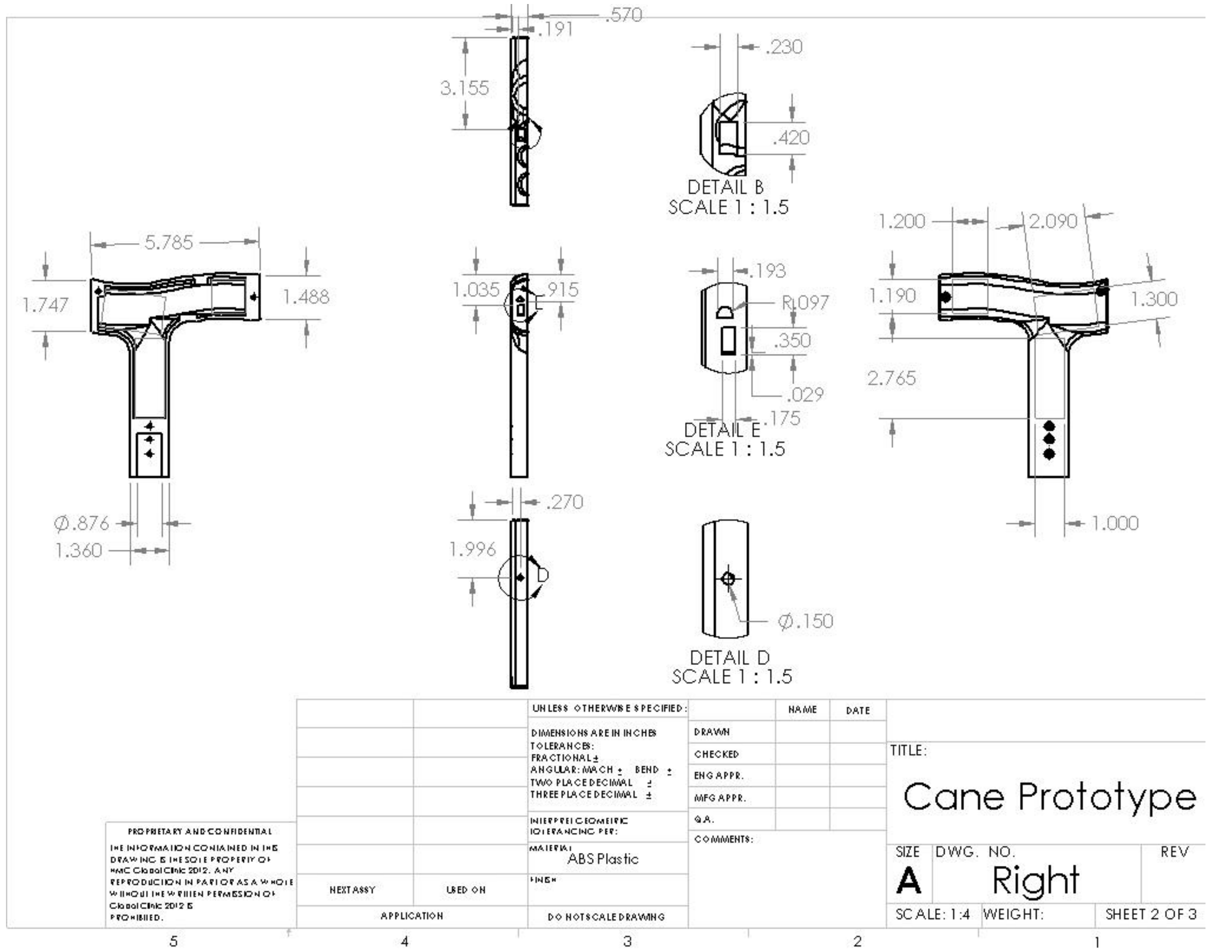


Figure 44: Right Half Portion of the Cane Prototype

22. APPENDIX K: CHRONOLOGICAL PROGRESS OVERVIEW

22.1. SUMMER 2011 PROGRESS

In August 2011, the HMC group traveled to Tokyo in order to become acquainted with the KU group and lay the groundwork for future collaborative work necessary for the Global Clinic project. During this period, the Global Clinic team visited Japanese companies and consulting centers to gain a better understanding of the Japanese elderly population and market. Initial design proposals and the team charter were also introduced before the HMC group returned to the United States. During this period, the team decided on a few possible directions that the project can take.

22.2. FALL 2011 PROGRESS

In accordance to the team charter, the fall session goes from September 2011 to November 2011 with Benjamin Liu being the HMC team leader. The corresponding team leaders on the KU group are Masahiro Kato and Yohei Katayama. During this session, the KU group performed extensive research on the Japanese elderly population and brought up several proposals for project directions to the HMC group. With this information, the HMC group developed an initial prototype. A suite of sensors was tested and incorporated into a working breadboard, which feeds information wirelessly into a GUI that has been coded. The HMC group has also jumped ahead of schedule by developing a PCB for the initial prototype, something that had originally been planned for the winter session.

22.3. WINTER 2011 PROGRESS

The winter session goes from December 2011 to February 2012 with Noah Duncan being the HMC team leader. The corresponding team leader on the KU group is Yohei Katayama. During the winter session, the KU group continued to research and develop form factors for the sensor platform. The HMC group progressed on further circuitry development, PCB designs, and programming for software interfaces. The KU group visited HMC during the end of February and early March, during which the full team conducted a joint presentation at the spring clinic presentations. The visit also culminated in the production of a set of first generation form factor prototypes.

22.4. SPRING 2012 PROGRESS

The spring session went from March 2012 to May 2012 with Edward Wang s the HMC team leader. The corresponding team leaders on the KU team were Yohei Katayama and Kazuma Ouchi. During the spring session, the Global Clinic team developed a sensor debugging and calibration software platform, a third party user interface, and a second generation prototype for the cane form factor with integrated sensors and a functional printed circuit board. The spring session also resulted in a final report and a Projects Day presentation at HMC. At the end of this session, the Harvey Mudd College group's senior members will graduate. Thus, any further project development will be carried on by the Kogakuin University group under the leadership of Kazuma Ouchi and Kyosuke Kusaka until their clinic's conclusion in January 2013. However, Benjamin Liu and Samantha Ipser of the Harvey Mudd College group will travel to Japan over the summer and continue with developing and testing the system while working in conjunction with the Kogakuin University group.