2018

FINAL REPORT

TEAM 9: ENGR 339/340 SENIOR DESIGN PROJECT
CALVIN COLLEGE

THOMAS MATHESON, DANIEL MICHAELS, TOMMY MATHESON

Executive Summary

This document provides a detailed description of Team 9's senior design project. The project was proposed by AvaSure LLC, a leading firm in the development of hospital based telemonitoring systems. The team designed a wireless alarm device, called the remote status alarm (RSA), as an accessory device that can wirelessly connect to a telemonitoring tower. The alarm device can be mounted above a hospital door, and an alarm is indicated by a flashing light on the device's face. It connects to a station wirelessly via Bluetooth and the connection is automated using near field communication (NFC), making the device easy and intuitive to use. The RSA also boasts low power consumption and solar harvesting capability.

In addition to designing the alarm device, the team also emulated the functionality of the telemonitoring system using a Raspberry Pi with an NFC shield. On the Raspberry Pi, the team wrote a program that allowed the Bluetooth address of the RSA to be read from the RSA's NFC tag and then get passed to BlueZ,[1] the Linux Bluetooth stack on the Raspberry Pi. Bluetooth pairing was then initiated using the open source Qt framework, a Linux software package. This program allows for the user to interact with the system and activate the alarm once the device has automatically connected, using a custom graphical user interface (GUI) on a capacitive touch screen.

Through the development of the RSA, the team learned a significant amount of information on the topics of software development in the form of real time operating systems (RTOS) and Bluetooth and NFC stacks. The team found that, based on expected usage data, indoor solar energy harvesting should be adequate to keep the unit powered for at least a year, a key design specification. This document contains information regarding the process for determining the feasibility of the project, component selection, and prototype testing of the project as it related to the proposal from AvaSure LLC.

[1] See report in Appendix A.

Table of Contents

Table of Figures

1 Introduction

1.1 Project Description

When a hospital patient takes a fall, it can have a major impact on the physical and emotional health of the patient and cost the hospital tens of thousands of dollars. AvaSure LLC has developed a solution to help mitigate this problem, called the AvaSys® Telesitter system. The system allows a technician to remotely monitor multiple patients at once, and alert on-floor workers if a patient is at risk for a fall. This monitoring is enabled using an AvaSys® Mobile Unit (AMU), which is a movable tower with a camera on it that streams live video footage from the hospital room of an at-risk patient to a monitoring station. Figure 1 shows what an AMU looks like when setup in a patient room.

When a patient is at risk of falling, the technician can activate a "stat" alarm on the AMU to indicate that an on-floor nurse needs to step in. The alarm currently consists of a loud tone playing directly out of the AMU. However, this can turn out to be confusing for responding nurses, because there are times when adjacent rooms each have AMUs inside of them. When the audible stat alarm is activated, nurses can hear the alarm, but have trouble determining which room needs a response, and this leads to longer response times.

Our project was the design of a remote stat alarm (RSA) indicator that will be placed in the hallway outside of each hospital room that has a monitoring unit. The device uses two ultra-bright LEDs to quickly indicate to a nurse which room has the patient that needs intervention. This alarm device has potential to cut down on the time between the alarm activation and the physical intervention, potentially increasing the chance of successful fall prevention.

Figure 1. AMU in patient room

1.2 Project Scope

The project timeline was eight months, starting in October 2017 and continuing to May 2018. Ben Moes, Electrical Engineer at AvaSure, was the technical advisor of the project team, and met with the team as needed. The student team was composed of Daniel Michaels - Hardware Engineering Co-Op within the Product Engineering Team at AvaSure– and his two teammates, Tommy Matheson and Trenton Wells. The team was tasked with developing a functioning prototype of the RSA that could pair and be controlled by a Raspberry Pi for demonstration purposes. Additionally, a prototype housing for the unit was requested for demonstration purposes.

1.3 The Team

Daniel Michaels

Daniel Michaels is currently the Hardware Engineering Intern/Co-Op on the Product Development Team at AvaSure. After graduation, he will be joining GE Aviation in Grand Rapids as a part of the Edison Engineering Development Program. He enjoys playing music and running. One day he hopes to incorporate his passion for embedded systems and music into a career.

Trenton Wells

Trenton Wells is currently an Electrical Internal Development Intern at ADAC Automotive. He enjoys reading and is an amateur blacksmith in his free time. His passion for making and learning blends into his profession and he enjoys keeping up on the latest technology and innovations.

Tommy Matheson

Tommy Matheson is currently a Controls Engineering Intern at Dematic North America. He is also an amateur DJ, and this, sparking a love for music equipment and technology, helped lead him down the path of electrical engineering. Apart from his engineering life, Tommy is an avid swing dancer and rock climber.

2. Project Management

2.1 Team Organization

AvaSure, as the project sponsor, dictated the project requirements and specifications. AvaSure assumed the role of team lead for project and directed the course of the project. However, the team was given a great deal of autonomy and self-management as to how the goals were to be met and which member of the team provided resources and solutions to them. The team performed the design research, implementation, and feasibility studies of the project.

In addition to support from AvaSure, the team had many external resources available to help. Through the senior design class, the group was paired with Professor Mark Michmerhuizen, who served as a mentor and was the recipient of the class-assigned deliverables. An industrial consultant, Eric Walstra, also agreed to meet with the team, and provided the team advice and help in determining the project timeline and aspects of the design process.

The team decided to host team documents on a Calvin College SharePoint site. This was chosen because of the integration with the full Microsoft Office 365 software package. This allowed the team to fluidly collaborate across all development platforms and access all files, both in the cloud and locally. The site also allowed for consolidated storage of the team's written reports and presentations. It also contained a research folder where relevant external documents, like chip datasheets, were kept. For code base management and sharing it was decided that using a Github account for the group would increase the ease in which collaboration was facilitated throughout the project.

2.2 Schedule

For the scheduling of tasks and estimating completion time, the team originally opted to use GanttProject, a free alternative to Microsoft Project that could also be imported into Microsoft Project. This allowed full integration with the SharePoint site and instant access to the team schedule and current tasks through the cloud.[2] There were some limitations to this approach, because the team was not familiar with the operation of this scheduling software and lacked experience in estimating task completion times.

After consulting with Walstra, it was decided that the team would be more successful using major milestones to track progress rather than a Gantt chart. The advantage of this was that it ensured that the team was continually discussing what everyone was working on and staying on track, without requiring the maintenance of an overly detailed scheduling system. With a small team and only a few tasks at a time this approach relieved the stress of a more rigid schedule as the time required for a task was not completely understood. The team estimated that the time required for each of the members per week was 4-6 hours to meet the milestones that were placed for the completion of the project. Over the course of the semester, the team found that more time every week was needed, and after an January where milestones were consistently missed, the team had to increase the amount of time spent per week.

This increase in time spent was largely spent on software development, which took considerably longer than expected. Software had to be developed for both the Raspberry Pi host and the alarm device, and these two tasks were originally slated to take two months in parallel from start to finish. This was significantly under estimated because of the unforeseen complexity of utilizing an unfamiliar system architecture and understanding wireless communication protocols. For the future, the team now has a better understanding of the amount of time required to achieve a similar goal.

[2] See Schedule in Appendix A.

2.3 Budget

The budget was managed by AvaSure LLC. When the team needed to purchase an item, a purchase order was submitted to AvaSure and reviewed. The team was not given a defined budget but AvaSure expressed that given logical reasoning for a purchase, it would be made for the team.[3]

2.4 Method of Approach

The design methodology was based on the team's previous experience as research engineers for their respective employers. This provided as a well-formed basis for work and was instrumental in allowing for much of the self-taught nature of the project. The team's main form of communication was face to face meetings inside and outside of class time. Electronic means of communication included Facebook Messenger and email. Both methods provide records of the conversations for future reference and referral.

[3] See Purchases in Appendix A.

[redacted]

[redacted]

[4] See Appendix A

4 Schedule and Task Breakdown

The project was broken down into several key stages and sub stages. This allowed the team to be agile and productive when a single task encountered a delay. The task breakdown lays out a hierarchy for level of completeness on various stages of the project. The major milestones were research, design, development, and testing.[5]

The research section was the first stage, and through research, the team determined how a system such as the one proposed could be accomplished. Much of this task was completed from October to November of 2017. However, the task remained open through a majority of the project as a higher and higher degree of understanding was needed throughout the project. As the main goal of the research section was to develop an understanding of the technologies that were implemented in the design, the research phase was split into five sections.

The research was broken down into: Bluetooth Low Energy (BLE), Near Field Communication (NFC), Solar Harvesting, Battery Management, and LED control and selection. These different sections were researched in parallel by the team and resulted in a great deal of information and design options.[6] The remaining time was spent developing a much greater understanding of the information and a deeper view of how a single system can utilize all the options as efficiently as possible.

The design phase began after the initial research was assembled and consists of five sections: hardware design, software, power requirements, housing design, and documentation. The hardware and software phases were done in parallel as the needs of the systems hardware architecture and software architecture are intricately linked in an embedded system. Software dominated the team's time and became the largest undertaking of the project. The software written is not a substantial code base but the underlying principals, protocols, and a plethora of new tools and techniques became a major part of the implementation process. Utilizing the major milestones and weekly in-depth progress reports, the team developed an acute understanding of the need for teamwork and documentation.

An addition to the original proposal of building the RSA, AvaSure and the team decided on the need for the development of a host system emulation. This stage added a substantial amount of additional work for the team but fit within the scope and timeline of the project. This system was determined as a need by AvaSure and the team to fully demonstrate the device at the Calvin College Engineering Senior Design Night, due to the non-disclosure agreement (NDA) between the two parties.

The final work and test reports were done during the final stages of the hardware prototype assembly and software optimization. During discussions with AvaSure they expressed a much greater desire for fully realized test reports and software documentation than for the hardware or enclosure to be fully realized and the team adjusted accordingly.

[5] See Appendix A.

[6] See design options in Appendix A.

5 System Architecture

[redacted]

5.1 Remote Stat Alarm System

[redacted]

[redacted]

5.2 Host System Implementation

The emulation of the host system was determined to be a low-cost system for demonstration and feasibility studies only. Therefore, the decision was made to use a Raspberry Pi with built in Bluetooth Low Energy, coupled with an NXP NFC Dev Kit. The reason behind this style of implementation was accessibility and system cost, in which there were no cheaper alternatives bearing the Raspberry Pi's level of functionality. The addition of a seven-inch capacitive touchscreen to the host system allowed for a highly compact and streamlined host experience.

The Raspberry Pi is a fully developed Linux system and allows for a highly integrated system as shown in Figure 4. This allowed for us to utilize several high-level software packages to enable the functionality that we desired. The system utilized the Raspberry Pi's integrated hardware abstraction layers (HAL) accessed through the D-Bus layer, which is the Raspberry Pi's data bus. Utilizing D-Bus streamlined the system software development significantly.

The outermost layer of software was built on the Qt framework.[7] The NFC system driver and documentation were used to create a custom graphical user interface (GUI), which allowed for the monitoring of the state of the peripheral device including alarm and battery status.

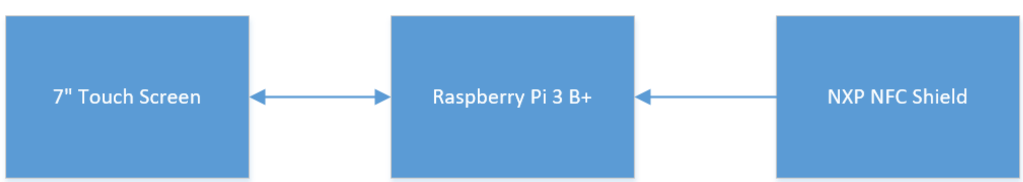

Figure 4. Host System Architecture

The MCU on the RSA was controlled via Bluetooth by the Raspberry Pi. Once the two devices had been paired, the RSA could be triggered by a signal delivered via Bluetooth from the Raspberry Pi, much like toggling an I/O. In addition, the RSA periodically transmitted its battery status back to the Raspberry Pi. Overall, the Raspberry Pi acted as the controller for the system.

[7] Qt is licensed under the LGPL 3.0 open source license. LINK

6 Design

6.1 Design Criteria

6.1.1 Components

[redacted]

[redacted]

6.2 Design Norms

All the design norms were important to the project; however, the following were especially relevant to the design.

Delightful Harmony – It was important for our design to be beautifully simple and intuitive. When it comes to the hardware design, a minimal selection of parts in a simple layout was desired. This is the reason that we chose to go with a chip that had an integrated Bluetooth module. On the software side of things, we chose to facilitate the Bluetooth pairing using NFC. This made the product easy for the end user to operate and implement as a natural step in the fall-prevention process.

Stewardship – Incorporating sustainable elements into the design like solar power and low power components decreased the impact of the device on the environment and were greatly desired. At the beginning of the project, it was thought that if we could get the alarm device to run solely off the ambient light in the hospital hallways, it would mark a great success in terms of power sustainability. However, because it was also desired for the device to exhibit delightful harmony within the hospital ecosystem, it needed to allow for alternative charging methods if the hospital's lighting was not adequate to support this initial goal. At the end, both were achieved, given optimal lighting conditions in the hospital.

Justice – The design needed be good for all involved parties, including the stakeholder. This meant that the design needed to use components that were suitable and economical. Alternatives needed to be considered for components that were deemed too powerful for the job. Over the course of the project, we became aware that some components, especially the microcontroller, were overpowered, and for production purposes, there could be more economically viable options. However, the use of such a powerful microcontroller does allow the client the overhead to add features to the device in the future. The team has decided that in the low to medium volumes, the price advantage of a less powerful microcontroller is negligible, and thus satisfies the design norm.

6.3 Design Decisions

Several design decisions were made regarding the system architecture and system hardware chosen.

6.3.1 Solar Power

To incorporate sustainability into our system, and to also introduce the potential for a perpetually charged device, a method of charging our battery while the device was in use was sought. The only real choice for this was to harvest ambient light from the hospital hallways using solar power. An alternative would have been having the device plug into a wall outlet, but this would have decreased the usability of the device, making above door mounting difficult, and not lending itself to sustainability. In addition, harvesting energy from vibration or from the NFC communication, though possible, was deemed insufficient due to its extremely low yield in comparison to solar.

The specific solar cells used by the team were chosen after extensive testing of solar cell options. This testing is discussed in more detail in Sections 7.2 and 7.3. [redacted] As size is an important constraint for the project, the ability to use only two solar cells was a huge success for the design.

6.3.2 Bluetooth LE

Bluetooth LE is a form of the Bluetooth protocol that uses lower power than the standard Bluetooth protocol. Since the design needed to have a minimal power drain over time, it was an easy choice to go with this protocol, and for this reason, was required by AvaSure. Additionally, with a lower powered system, the design became more sustainable. Lower power components made the design more expensive but were necessary to achieve the performance requirements as detailed in section 3.3.

6.3.3 Microcontroller Choice

[redacted]

[redacted]

6.3.4 Charge Controller Choice

The charge controller was one of the more important component choices for the design. [redacted]

6.3.5 Battery Read Circuit

Due to the voltage limits of the MCU's I/O, a resistor network was required to provide a node with a scaled battery voltage value that was within the readable range of the MCU. This is a simple circuit but adding it as a permanent path between the output of the battery and ground meant adding a constant power load on the system. To reduce the power consumption of the system, a switching circuit was developed using a BJT driven by the MCU to pull the gate of a PFET to ground allowing current to flow to through voltage divider resistor network. In affect this meant that only when the MCU was reading the voltage on that pin, that power would be consumed.

6.3.6 NFC Tag IC

The team spent a significant amount of time searching for an NFC Tag IC. [redacted] The reason an active NFC chip is required as opposed to a passive NFC tag is due to power consumption. The original approach that the team considered was using a passive NFC tag which would only hold the devices Bluetooth address. However, using a passive NFC tag would require the BLE module to always be advertising when not connected, which would keep the system from ever entering a low-power standby mode. For our system, active refers to the ability for the NFC tag to drive an output either high or low that can be read by the MCU. These active NFC ICs require a source voltage in order to drive that output so they must be connected to the regulated output of the energy harvester.

7 Testing

7.1 Development Kits

[redacted] These tests are more thoroughly documented below. The solar cells were tested to see which configuration would provide the most power in a given lighting setting. The charge controller was tested to ensure it functioned as expected, and delivered enough power on the regulated supply to power the [redacted]

7.2 Solar Cell Testing

Extensive testing was performed with several solar cells. An official test report was written for AvaSure for this testing. An abbreviated summary of that report is included in this section. This references the "Urumi Solar Cell Test Report.docx"[12] in the Appendix.

7.2.1 Unit Under Test

The parts tested for the solar cell test report were evaluated for maximum performance under florescent lighting conditions that were determined to be like that which will be seen in a hospital hallway.

7.2.2 Test Criteria

This was a feasibility study and helped determine the criteria for future applications.

7.2.3 Test Setup (Diagram, Photographs):

The purpose of this was to test the practical operation of solar cell options for use on the peripheral as well as the viability of solar charging in hallway/office setting.

[redacted]

Figure 5. Solar Cells for Testing

[12] Full Test Report in Appendix A

The three solar cells that were tested are shown above in Figure 5. All three cells are advertised [redacted]

[redacted]

Figure 6. Test Environment

Highlighted by the red circle in Figure 6 is the surface on which the measurements of each cell were taken. Figure 6 also shows the lighting environment that was used.

Figure 7. Quantifying Environment Lighting

The Lux meter, as seen in Figure 7, was used to measure the strength of the ambient light on the test surface, which was found to be 420 Lux.

Using the following conversions, the available Watts at the test surface could be calculated[13]:

The luminous flux Φ_V in lumens (lm) is equal to 0.09290304 times the illuminance Ev in lux (lx) times the surface area A in square feet (ft2):

$$\Phi_{V(lm)} = 0.09290304 \times Ev_{(lx)} \times A_{(ft^2)}$$

The power P in watts (W) is equal to the luminous flux Φ_V in lumens (lm), divided by the luminous efficacy η in lumens per watt (lm/W):

$$P_{(\mathrm{W})} = \Phi_{V\,(\mathrm{lm})} / \eta_{(\frac{\mathrm{lm}}{W})}$$

Resulting in the following equation:

$$\mathrm{P}_{(W)} = \frac{0.09290304 \times Ev_{(lx)} \times A_{(ft^2)}}{\eta_{(\frac{\mathrm{lm}}{W})}}$$

Using 420 lux, an approximated surface area of 0.00174 ft^2, and approximately 60 lumens/watt which is typical of fluorescent tube lights, we calculated that the power available for a solar cell to harvest is 1.132 mW.

Test Criteria:

The Energy required to drive 3 single LED strings at 15mA with a 3.6V forward voltage for the RSA was calculated using a LED current of 0.045, LED voltage of 3.6 vdc, and an LED on-time to be 75 seconds.

The LED On-Time was calculated based on a 25% duty cycle with the alarm being triggered 5 time a day for 1 minute. Using a tool provided by TI, the energy consumption of the CC2640R2F chip assuming 100% connection time, and advertising 4% of the time resulted in 22.3uW of power consumption.

[redacted]

Test Analysis:

[redacted]

[redacted]

[redacted]

[13] Equations and explanation source in Appendix A

7.3 Energy Harvester Development Kit Testing[14]

7.3.1 Test Setup

The purpose of this test was to evaluate the real-world performance of the solar cells in charging a [redacted]

[redacted]

Figure 8. Li-Po charging setup.

Figure 8 above shows the setup for [redacted]

[redacted] and the availability of solar cells, only these three cells were chosen based on their size, Voc and Isc.

[14] Full report can be found in Appendix A

Test Criteria:

The initial test requirement was that the power scheme needed to be capable of harvesting enough power to run for one year independently. It also needed to be capable of being powered for one week without being able to harvest energy. Based on our estimated power requirements, this equates to the system being required to harvest at least 14 Joules per day to be net neutral on power.

One of these setups [redacted] In addition to that, despite their high price, another option that was considered was [redacted] See the full report in Appendix A.

7.4 Prototype Testing

The team developed and implemented a prototype testing plan that included hardware verification, software verification, functional testing for enclosure and LEDs, and power management testing. The test summary and results can be seen below.

7.4.3 Test Setup

[redacted]

Figure 9. Prototype with yellow positive battery connection short through DMM.

[redacted]

Figure 10. DMM in AVG. mode to measure current.

[redacted] The current measurement is shown in Figure 10.

The individual components of the prototype system were measured for current and voltage and from there, the consumed or harvested joules from each could be calculated. First, the solar cells were measured while lying flat on the desk in the housing, as had been done in the original solar cell tests performed earlier in the project.

As outlined in the report, the solar cells can harvest ~25 joules per day. Recall, the original solar cell testing and power calculations had determined, based on now obsolete assumptions that the solar cells would need to be able to harvest at least 14 joules per day. These assumptions were based on a poor understanding Bluetooth Low Energy which placed the expected energy consumption of the Bluetooth microcontroller at ~2 joules/day. To test this assumption, the average current consumption was recalculated using [redacted] would use exponentially more energy at 82.5 joules/day.

The final components that were tested were the LEDs. By design, the LEDs are powered with 2.8V with no current limiting resistors. This is extremely close to their forward voltage at which point they pass approximately 15mA each. Through the software, the team made the LEDs be driven with a "heart beat' or "breathing" effect by modifying the PWM duty cycle. The effective average duty cycle is 17.989%. Based on data from AvaSure it is also known that the stat alarm is activated on average for 1 minute five time a day. Using the duty cycle and the activated time, we can calculate that the LEDs will be on for 53.97 seconds per day.

For understanding expected battery life, the capacity of the battery had to be calculated in joules as well. This was done by using an equation found at: http://www.rc-electronics-usa.com/battery-electronics-101.html. As the report states, it was found that a 1200mAh battery with nominal voltage of 3.7V contains 15,984 joules.

Test Analysis:

Based on these measurements and calculations, we can see that our net energy is $E_{solar} - E_{Micro} - E_{LED} = -62.04\, Joules$. With a battery that has 15,985 joules, this equates to 257.493 days or 0.705 years of operation. However, there was a problem with this calculation. In testing, cell phones have had to be used to connect and control the prototype device when it is at AvaSure. Both iOS and Android devices use a default connection interval of less than 10mS. Because of this, 7.5mS was used as the connection interval for the calculation. This interval can be set as high as 4 seconds, but that is controlled by the central device. The peripheral unit can choose a range of connection interval max and min values but it is ultimately up to the host device to decide. The RSA does not need to have a connection transfer anywhere near as frequently as 10ms. If we rerun the calculation with something far more optimal such as 100ms, we see as in the report that this has a drastic change on the energy consumption of the microcontroller.

If the microcontroller energy consumption can be decreased to 2.71 joules/day through software, then the net energy consumption per day would become net positive. This means that the system would be able to fully charge the battery and sustain itself without any charging as the total system load would be 7.245 joules while the solar cells can provide 25 joules in reasonable lighting conditions.

7.5 Software Testing[15]

Software testing consisted of several iterations of team reviews and meetings with AvaSure to detail what steps were being taken to ensure a successful project. In these meetings, the software was reviewed line by line and the developer (team member responsible for the code) provided detailed reasoning behind the style, usage, and implementation documentation for AvaSure. This was coupled with the prototype testing where the device was loaded with the software and utilized in several real use cases.

The use cases stressed were: pairing a device to the host using NFC, triggering an alarm through the host GUI, reading an accurate battery percentage from the peripheral on the host device, and pairing a new device with the host while a device was in the connected state with the host already.

Due to the prototype nature of the device no other software testing was conducted. The team has proposed additional testing in the form of long term stability, functional timing, and unit testing to be conducted after a final hardware revision is met and prior to final production. The team believes that this will ensure that any issues would be found prior to the device being implemented into the medical environment.

7.6 Electromagnetic Compliance Testing

The original plan for EMC testing is as follows. The team would have taken the design through the first stage of electromagnetic compliance (EMC) testing with E3 Compliance out of Grand Rapids, Michigan. E3 Compliance has the equipment and personnel trained to provide design guidance and full UL compliance[16] reports that were within the scope of the project. EMC testing would have provided the team with information regarding the electromagnetic interference that the device created and would have been susceptible to. This would have allowed confidence in the robustness of the design and knowledge that it would safe for use in a hospital setting where there are life critical devices.

However, due to the not-production-ready nature of the team's design, it would have been a frivolous expense to test the design for EMI. For this reason, this testing was not completed for the project.

[15] See Source Code in Appendix B

[16] See UL Specifications in Appendix A.

8 Final Prototype

The team designed and 3D printed a housing to showcase the product in the final presentations, and this housing can be seen in Figure 11. The housing contains adequate space inside of it for the system components, and the boards are mounted so that the LEDs are centered underneath light diffusion bar that stretches across the housing. The housing also features sockets for the solar cell mounting, and the socket section was designed at an angle that points them towards the center of a hallway ceiling.

Figure 11. Final Protypo 😉

9 Conclusion

This project required the team to use everything they have learned through their school and work experiences to meet the requirements and requested deliverables. Many of the systems designed and used for this project were outside the realm of expertise of the team. As a result, the team developed a great deal of understanding in the specified technology and other areas of engineering. This is a direct result of completing all aspects of the project and was realized through a reliance on the fundamental knowledge learned at Calvin College and the perseverance of the team through a year of self-taught system architecture design and implementation details. These skills will serve each of them well in their own personal hobbies and professional careers. The team believes that the system proposed, developed, demonstrated, and delivered to AvaSure clearly depicts our deep understanding of the needs of the customer through a well-balanced Christian engineering education. The delivered system meets the needs of the customer as well as advanced each of the team members levels of technical competence and has prepared them for successful careers as engineers.

10 References

[redacted]

[17] Document included in Appendix A

11 Appendix

11.1 Appendix A

Raspberry Pi NFC Report
https://calvincollege.sharepoint.com/sites/group-engineeringseniordesignteam9
UL Guidelines for Electronics in Hospital Settings
https://standardscatalog.ul.com/standards/en/standard_60950-1_2
[redacted]
Raspberry Pi 3
https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
Bluetooth Secure Simple Pairing Using NFC
http://members.nfc-forum.org/apps/group_public/download.php/18688/NFCForum-AD-BTSSP_1_1.pdf
Team Website
http://engr.calvinblogs.org/17-18/srdesign09/
Prototype Test Report
[redacted]
Solar Cell Test Theory Source
https://www.rapidtables.com/calc/light/how-lux-to-watt.html

11.2 Appendix B

Full RSA Source Code
[redacted]

12 Acknowledgements

AvaSure - for their financial support and technical mentorship, as well as for their trust in our team to help with their development.
Mark Michmerhuizen – for his continued feedback and guidance on the project, and for challenging us to increase our scope further than we had initially thought.
Eric Walstra – for encouragement and affirmation in our design process and goals, and for feedback on project management.