SLATE SAFE SENIOR DESIGN PROJECT

By

Joe Bianco

John Graft

John Simonaitis

Final Report for ECE 445, Senior Design, Spring 2018

TA: Zipeng Wang

2 May 2018

Project No. 9

Abstract

In the last two decades, online shopping has surged into a 400 billion dollar industry in the United States alone. This number is projected to grow to over 600 billion by 2021 [1]. At the same time, package theft has exploded, directly affecting over one-third of Americans and posing a serious concern to over half of them [2]. Our project, Slate Safe, is an attempted solution to this problem. Using a weight sensor, Slate Safe is capable of sensing whether or not a package or set of packages has been stolen. You can add weight to the device, but as soon as the weight sensors detect that a package has been stolen, a loud alarm will sound and a camera takes a picture of the thief. The user is subsequently alerted when their package has been stolen.

Contents

1. Introduction

The emergence and subsequent boom in online shopping has changed the way consumers shop and buy goods. According to a recent survey conducted by the Pew Research Center, 79% of Americans shop online, and 15% buy online on a weekly basis [3]. Now more than ever before, consumers have access to thousands of products with a tremendous degree of selection, all at the convenience of a few clicks and a standard shipping and handling fee. However, the convenience of online shopping is offset by a lack of security in the delivery process. UPS Ground packages are generally delivered Monday through Friday between 9:00 a.m. and 7:00 p.m., which overlaps heavily with the standard 9:00 a.m. to 5:00 p.m. work day. Hence, most people are not at home during UPS shipping hours, leaving their packages completely unattended. This increase in package delivery coupled with lax protection has led to a surge in package theft. According to a survey done by Xfinity Home, Comcast's home security service, more than 50% of people across the United States know someone who has had a package stolen, and about 30% of people have had it happen themselves [2]. Our own surveys have indicated that over a third of respondents are interested in a package protection system. Clearly, this is a pervasive problem.

Our goal is to design and construct a device that stymies package theft through a weight, alarm, motion, and camera-based security system. We use load cells to precisely measure the weight of a package, and an alarm system that is triggered when the weight of the package decreases past a certain threshold. If a potential thief gets close to the package, a PIR motion sensor is triggered, causing a verbal warning to sound and a camera to take a temporary picture of the person. If the alarm is triggered, the picture is sent via Wi-Fi to an Android application on the user's cell phone. The user can disable the alarm via the app over Wi-Fi or through an RFID tag. The device will also generate and run off its own power through solar cells and rechargeable batteries. We believe that these functionalities will provide secure protection of the user's package. The final product is shown in figure 1 below.

Figure 1: Slate Safe theft deterrence system.

2 Design

In designing the Slate Safe system, we had three general goals. The first was that system needed to eventually be scalable into a real product, and the more direct this transition could be, the better. This meant that we made sure everything put into the product, such as the chassis and boards, could easily be done to scale. We also wanted the ability to add and remove features easily so focused strongly on independent modular design. The second goal was to be it to be as inexpensive as possible. A poll we posted on local homeowner websites received over 200 responses, and all interested customers were only interested in paying less than $100, and 80% of those were only willing to pay less than $50. Other package protection products cost over $200, and the cheapest we could find is $80 excluding shipping. Thus we believe we have an opportunity to provide a low-cost solution people actually want. The third goal was to make the system as convenient as possible, being easy to setup for customers, easy to use for delivery people, and aesthetically pleasing, taking a minimalist design approach as this system is meant to be noticed only when needed.

From an engineering perspective, we planned to reach this through three high level requirements. These are the following:

- Build a robust weight sensing module capable of measuring at least 0.5lb and up to 50lb, without triggering false alarms
- Effective theft deterrence via a loud alarm, camera, and warning message
- Fully self-powered through solar cells and rechargeable batteries

Our initial high-level block diagram is shown below in figure 2. Though this system worked, several design changes reflected in the new block diagram in figure 3.

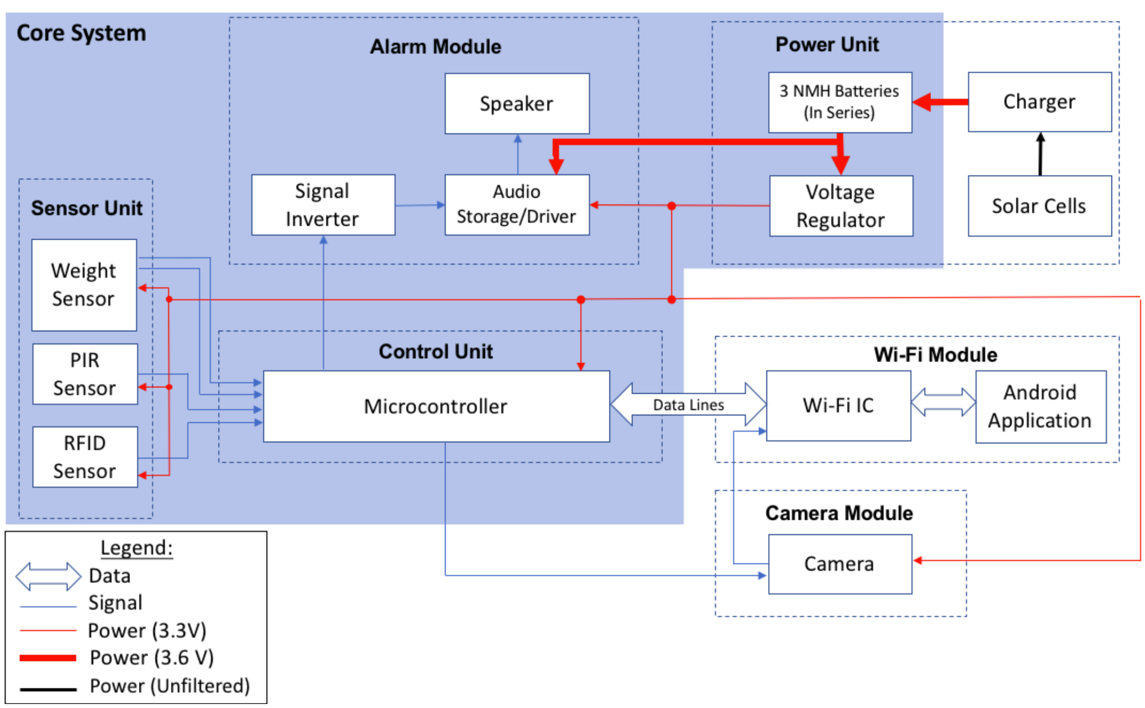

Figure 2: Original block diagram.

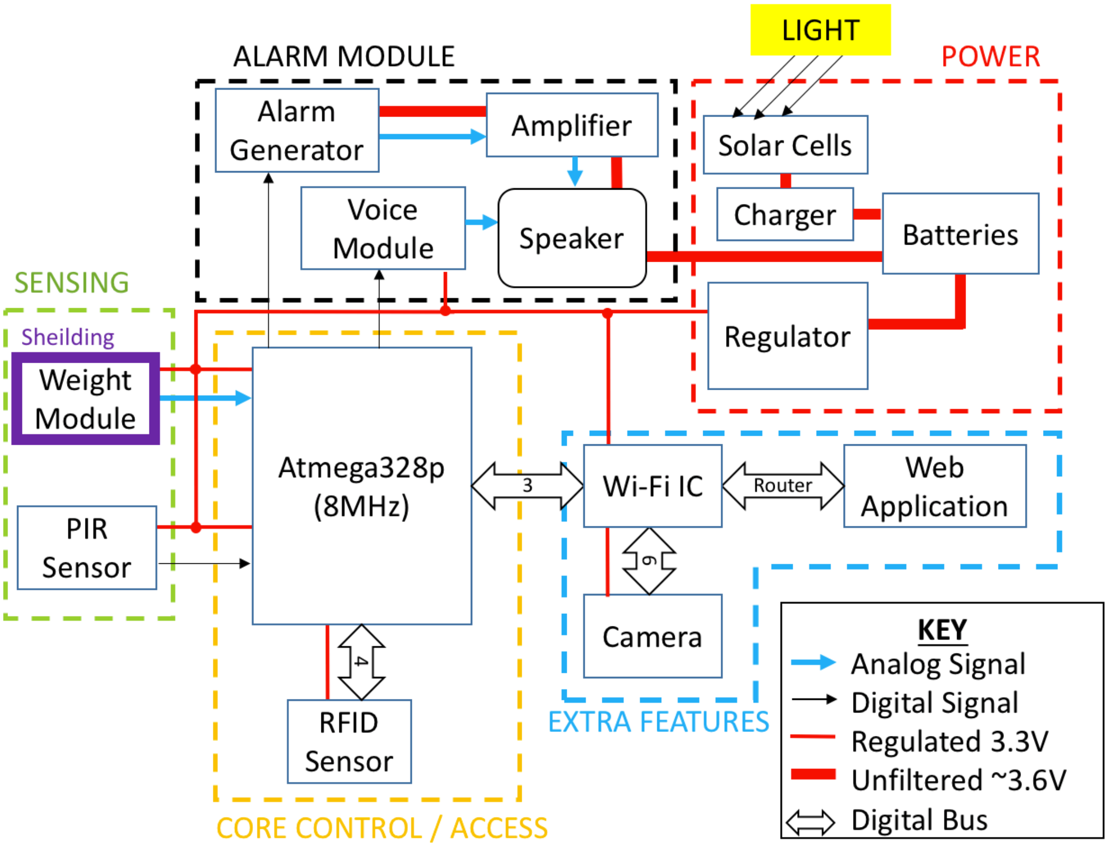

Figure 3: Final block diagram.

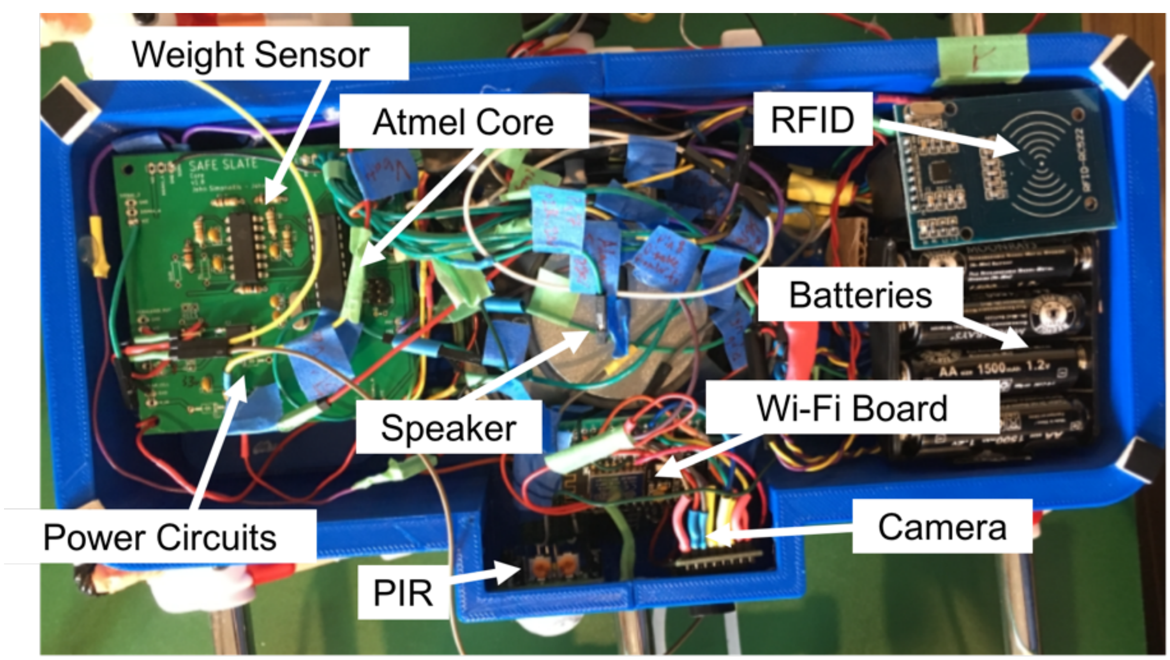

Figure 4: Physical implementation of the block diagram. The audio module is attached to the underside of the battery pack.

The new diagram makes four primary changes which improved performance and simplified the design. The first modification was the introduction of shielding to the weight module. Given we use a 1000x gain amplifier, we underestimated the significant effects of static electricity, parasitic capacitance, and electronic noise from other components. In order to prevent these effects, we used grounded aluminum shielding on all weight module electronics.

The next two modifications were simplifications taking advantage of the Atmega328p chip used. First, we completely removed the need for a comparator in the weight sensing circuit by changing the Atmel sleep mode design, then using the regular weight ADC to tell if a package had arrived to wake the system. Second, we replaced the zener diode/BJT circuit with p-channel MOSFET, which used the ADC to regulate battery voltage to prevent overcharging from the cells.

Finally, we realized the low audio fidelity and amplification requirements of our audio system meant that a simple BJT amplifier was all that was needed, reducing both cost and power consumption. All of these modifications are addressed in their corresponding sections.

2.1 Core Atmel + PIR +RFID

The core Atmel Processor, RFID sensor, and motion sensor are combined into one module for simplicity, since both the RFID Sensor and PIR sensor were separate off-the-shelf boards that we purchased [21-22]. Because of this, the only things we did with them were properly power, calibrate, and interface them with the microprocessor. Verification of these modules simply consisted of making sure they worked and did not interfere with other modules.

The core microcontroller was based heavily on the Atmega328p datasheet [4] and Arduino Mini design [5]. The only modifications come in the form of reducing the power consumption by setting the input power to 3.3V, halving the clock speed from 16MHz to 8MHz, and removing the programmer chip. These concepts were largely inspired by a SparkFun article [6]. To reduce costs, we directly place the circuit on the main PCB. The module, with all of the required inputs and outputs, is shown below in Fig. 5.

The most original aspect of this module is the code that controls the microprocessor, shown as a high-level flowchart in Figure 6. This code includes actions for disabling the alarms, detecting and warning potential thieves, and sensing if a package has been taken. We significantly reduce power consumption through our code by disabling features such as extra counters and brownout detection, and enable the device to enter sleep mode when there is no package on the system. However, we did not reach the power goals set for this chip due to an underestimation of current used by the peripherals. This is explained in the verification section.

As mentioned before, for our Kickstarter, we plan on offering two products. The first is an expensive version implementing all of the features mentioned in this proposal, such as Wi-Fi, a camera, and application for around $100. The second version is a “basic” module that includes only the weight sensor, alarm, and RFID unit with the core processor shown below. Because of this intended difference between the final products, we decided two separate microcontrollers would be optimal for streamlining the entire design process.

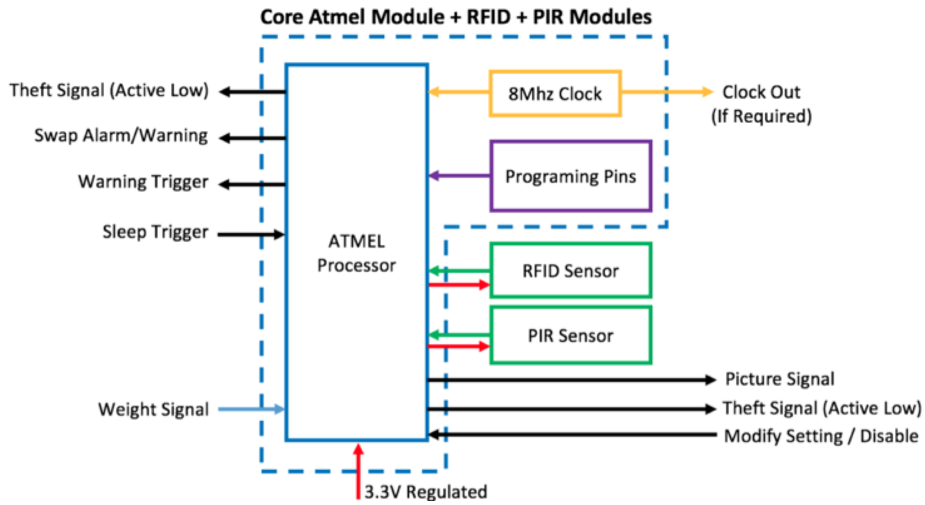

Figure 5: Core microprocessor with RFID and PIR interfaces.

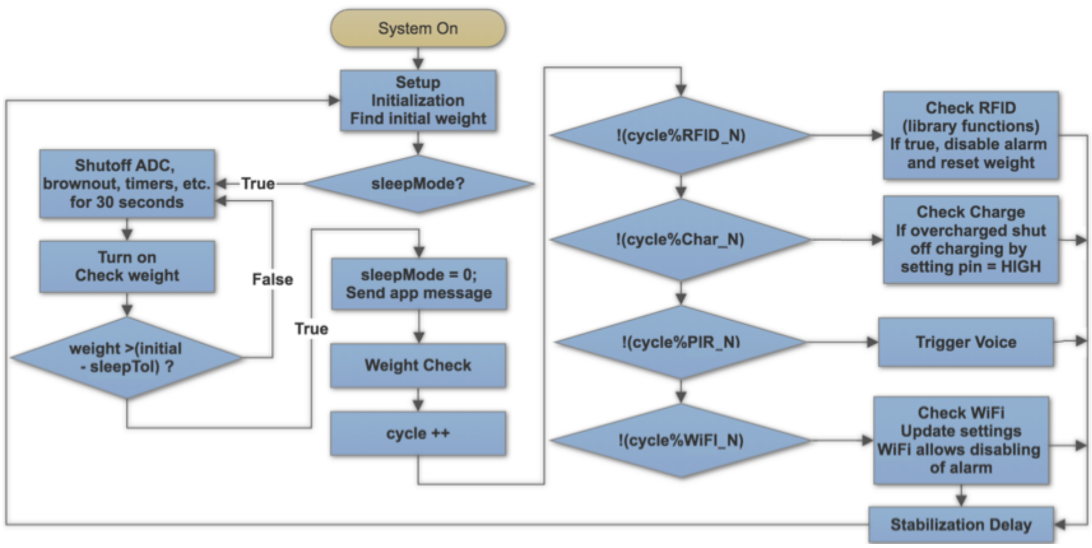

Figure 6: High level Atmel code overview, demonstrating sleep mode, cycling of the different features, and stabilization of the power source before weight measurements.

2.2 Weight Sensor

The weight sensor is the crux of this project's design. For our product to be successful, the weight sensor must be robust, inexpensive, and accurate. In order to accomplish these goals, we used 50kg full bridge micro load cells [8]. We considered using force sensors but found that their minimal improvement in precision was offset heavily by their cost and fragility [7]. Almost all commercial electronic scales use load cells, which led us to finding very inexpensive modules.

The weight sensor takes in 3.3V regulated power and output an analog power signal rated from 0-3.3V, corresponding to the weight of the package, as well as a digital output telling the user whether there is a package on the device or not. This digital output turns off the other modules in our system when there is no load, resulting in much lower idle power consumption most of the time. The circuit design for this is shown in figure 7. This design was inspired by a SparkFun article that we found [9], and its gain equation is shown in equation 1, which is standard for an instrumentation amplifier [10]. Its value was decided from determining the gain required to get full voltage clipping (3.3V) at 60 pounds of weight. Equation 2 shows the error propagation formula for load cells [11], and verified they would be accurate enough under a range of temperatures from -30°F to 120°F.

$$A_v = \left(1 + \frac{2R_1}{R_g}\right)\frac{R_3}{R_2} \approx 1100 \tag{1}$$

$$\varepsilon > \sqrt{\varepsilon_C^2 + \left(\frac{\varepsilon_t \times L \times N}{W_1} \times t\right)^2 + (\varepsilon_S \times t)^2} \tag{2}$$

One significant flaw in our initial design plan was an underestimation of the noise caused by static electricity and other electronics with an amplifier of such high gain. We solved this by shielding the precision amplifier, which is shown in the right picture of figure 8, and through shutting off extra electronics when not in use. This is shown clearly in the verification section of this report.

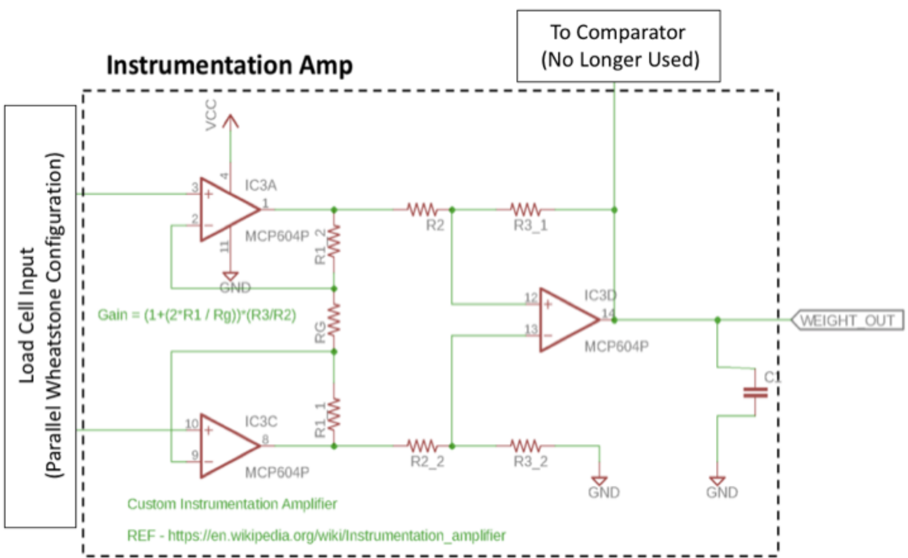

Figure 7: Zoom in on the instrumentation amplifier, showing the inputs and now unused comparator output.

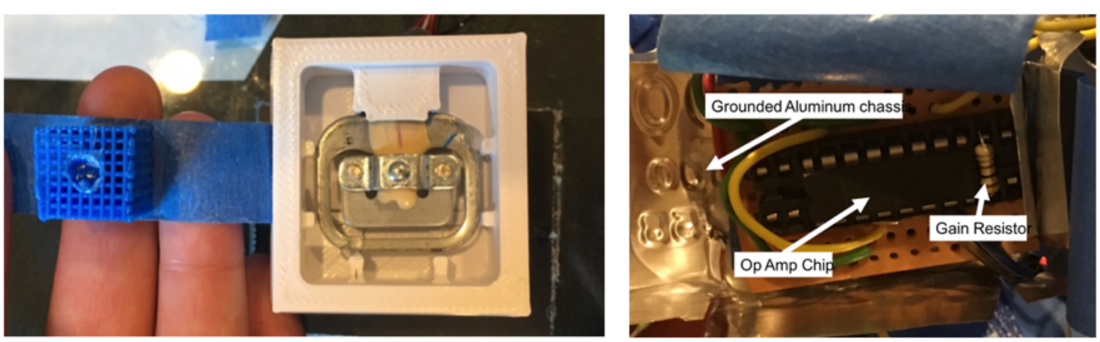

Figure 8: To the left is the load cell housing with the 3D printed buffered foot. To the right is the instrumentation amplifier circuit inside a grounded chassis.

We also used digital filtering techniques to further improve the weight data, and to detect whether a package was truly taken or if there was a carelessly dropped package or vibrations nearby. This was done by taking N samples from the ADC input, inserting it into an M element array, then finding the median of these elements. We keep the median and compare it to the last calculated median and an old “anchor point” from the past (generally from about 5 seconds ago real time), and if the value is outside of our set tolerances the alarm is set off. We show this in Figure 9.

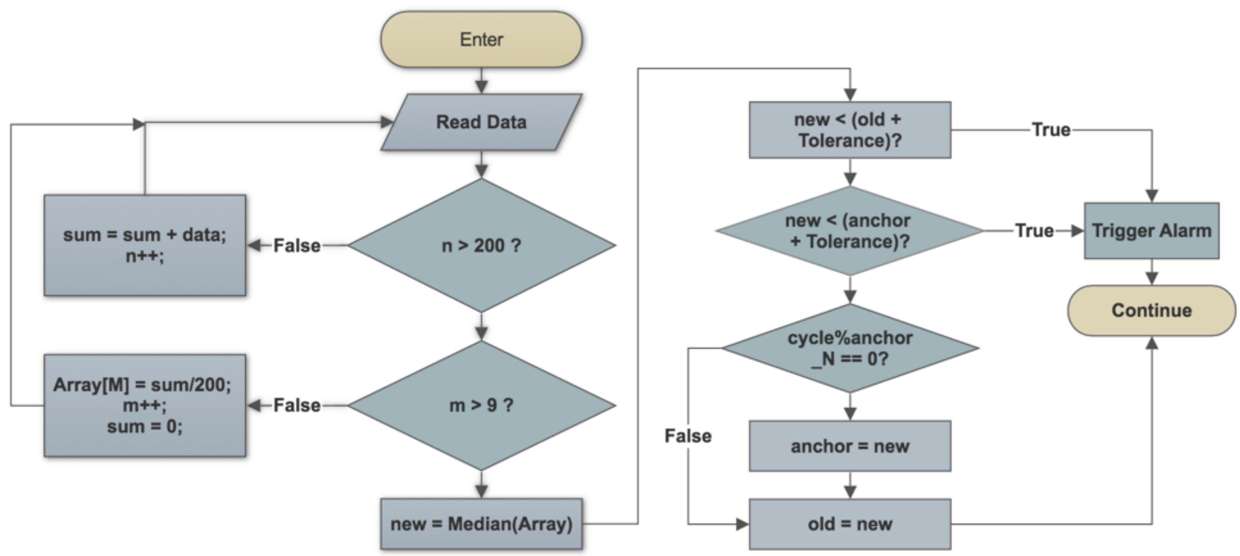

Figure 9: Control flowchart of the weight sensor.

2.3 Audio Module

The alarm module is the primary deterrence measure we implemented in this project. Its features include a loud speaker capable of producing a voice to warn people of its protection and a siren of the owner’s choice. This module is also built to be self-sufficient and highly durable, still going off even if the entire apparatus is smashed because it is directly attached to the battery unit. For prototyping purposes, we also designed the amplifier to have adjustable gain to get a desired output volume.

The module was broken down into three closely related circuits: the siren generator, the voice player, and the amplifier and signal selector. The siren generator was based off of the UM3561 chip, a very common and inexpensive chip used in cheap toys but which outputs a very low power signal [12]. The voice player was inspired by cheap recordable greeting cards, and from those we found the ISD1820 chip, which could record up to 10s of audio at a 6.4kHz sampling rate, which was adjustable [13]. We were able to program it using audio input from a computer, and although it was low quality of a message, with proper voice distortion (manually cutting out high frequency elements) we were able to get a very clear warning message from the unit.

The initial design used an audio amplifier chip and high frequency inverter to drive the voice warning and siren generator. This worked well, but we soon realized was overkill given we only needed low fidelity x30 gain for the siren to meet our targets, and no amplification of the voice module. This resulted in the simplified design of figure 9, where we have the voice connected to all floating high impedance nodes when the siren is off, allowing direct driving of the speaker. When the siren is on, which is the default value with no signal from the Atmel, first Vcc is turned on at point 1, which then uses a simple zener diode circuit at point 2 to ensure proper supply to the chip without a regulator. The driver then outputs a signal which is amplified accordingly by the BJT at point 3.

Some basic calculations guided our amplifier design. The speaker we purchased had a rating of 104dBa, which corresponds to a loudness of 104dB when driven at 1W as measured 1 meter away from the alarm. Early testing of our enclosure demonstrated to us that when played at 1W of input power in our chassis, the speaker would output around 95dB measured 1 meter away. Thus, we determined that we would need 100mW of power to generate 85dB of power, and that we could reduce the gain further if needed. At 3.6V powering the unit, 100mW corresponded to a required current of 30mA ≈ 100mW/3.6V, well within the gain and current capabilities of a simple BJT-based amplifier. This verified our new design, and for the final amplification we simply adjusted the BJT gain, given by equation 3 below, to final value of ~30.

The final step of the audio module was creating code to control both the voice and siren outputs. The alarm code worked by entering a timed while loop which constantly polled the WiFi and RFID sensors to disable the alarm. The voice mode was more complicated. When the voice module plays, it introduces a lot of noise to the rest of the circuit, most significantly to the weight sensor circuit which sometimes then triggered false alarms. Because we still need to protect the package while the voice message is playing, we must still check the package weights. Our solution was to increase the threshold tolerance of the circuit temporarily (to around 0.8lb), to prevent false voice triggering, before returning the tolerance to its normal value of 0.4lb.

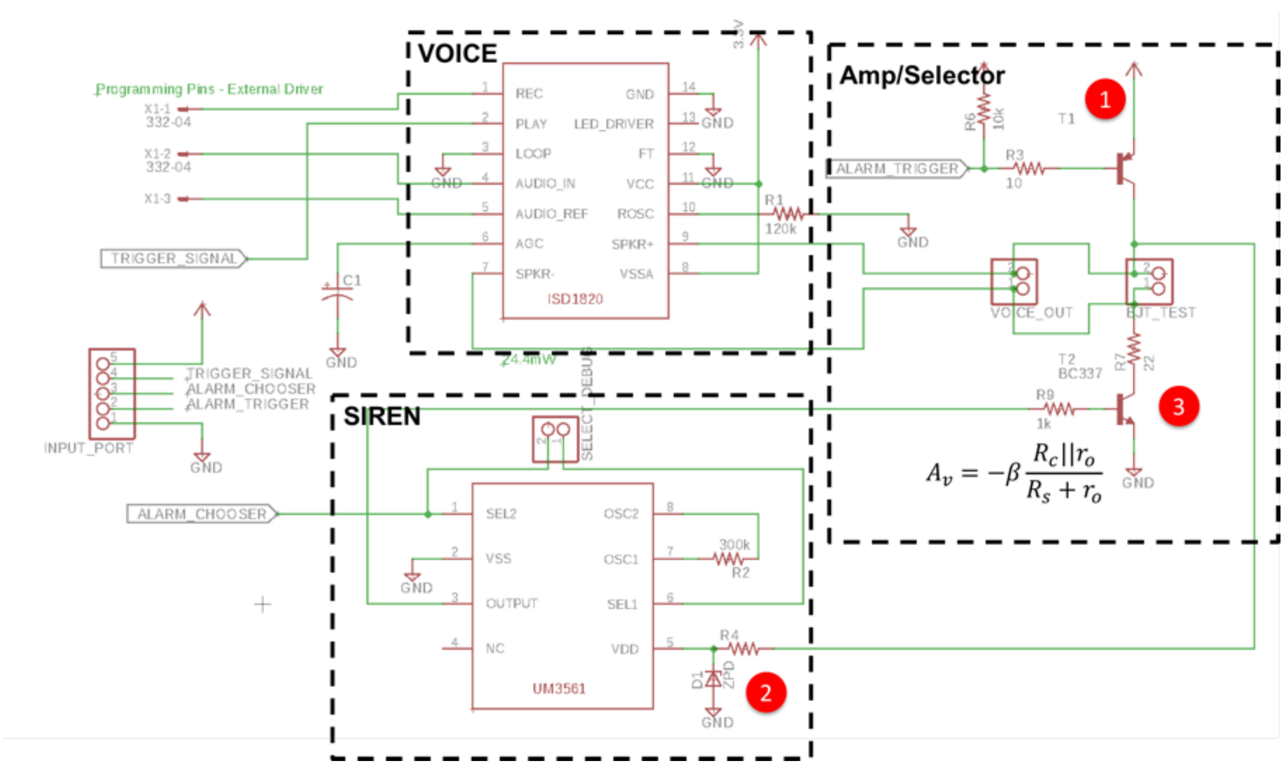

Figure 10: Final design of the audio module.

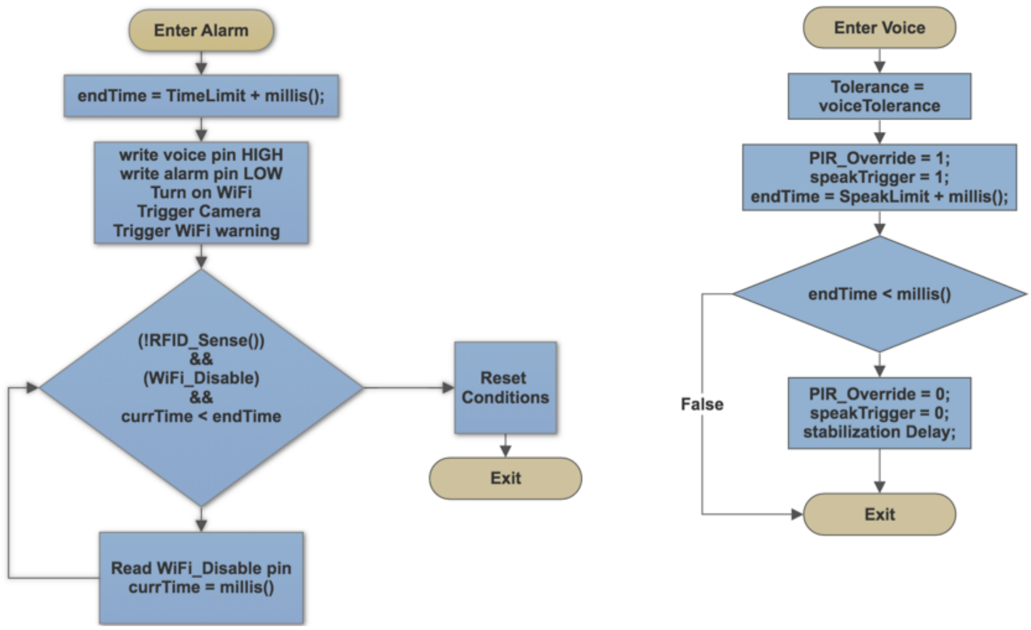

Figure 11: The code controlling the alarm unit and voice unit.

2.4 Power Circuit

The power circuit had four parts: solar cells, a charging circuit, batteries, and a voltage regulator. Each of these worked together to fully power all operations and safely charge the batteries even on cloudy days or when most of the cells were covered with packages. In our proposal and design document, we laid out the expected power consumption of each unit to show that the chosen solar cells would be capable of powering the circuit with only the equivalent of two hours of full sunlight per day. Most of our units were under the expected power consumption by significant amounts, leading to us more than reaching our goal in this regard, though for length purposes we have excluded this table.

The solar cells were designed so that each corner cell (seen in figure 1) would be capable of charging the batteries even in low light conditions. Under full lighting, their maximum voltage was 6V, and on cloudy days this reduced to 4.8V, still enough to charge our three batteries at 3.6V even with the drop due to the protection diode.

The original charging circuit was fully self-powered and independent of other elements, using a zener diode and BJT combination to stop the batteries from overcharging. This worked well, but after some thought we came upon a clever design that reduced the complexity, cost, and losses of the charging circuit. This new circuit is shown in figure 12, where the solar cells and battery are shown as ports. This circuit uses a p-channel MOSFET so that the default state is to charge the batteries, such as when the power is so low the microprocessor shuts off. When the voltage is high enough at the set limit as measured by the voltage divider to the right (since $V_{batt} > V_{cc}$), the Atmel core shuts off the P-Channel MOSFET in order to protect the circuit from overcharging.

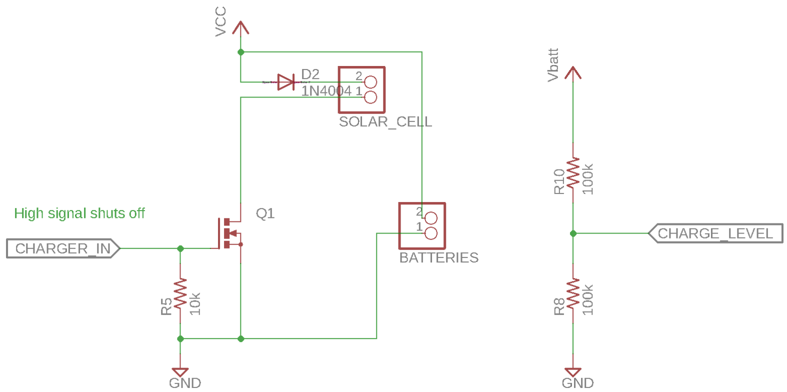

Figure 12: New microcontroller-based charging circuit.

2.5 Application Block

The final module of our project are included the WiFi, camera, and application modules. They are described below, and were designed extend the protection and user friendliness of the Slate Safe.

Wi-Fi Module and Application

The Wi-Fi module consists of an ESP8266 Wi-Fi transceiver to allow easy control over the Slate Safe device [14]. It serves four main functions: first, alerting the user when a package has arrived, second, alerting the user when their package has been stolen, third, uploading a picture of the thief to the web application, and fourth, allowing the user to enable or disable the alarm. When a package arrives, the weight sensor module sends a signal to the Wi-Fi transceiver which in turn sends a request to a website [15] that sends a custom text message to the user's cell phone. The same thing happens when a package is stolen, but when the package is stolen, the custom text message includes a link to the webapp so that the user can see the picture of the thief.

The application consists of an option to enable and disable the alarm, an option to see the pictures the camera has uploaded, an option to change the resolution of the camera, and an option to manually clear the data. The user connects to the web application by typing in the IP address of the Wi-Fi network they are using. We also developed an Android application, but had some trouble configuring the ESP chip with the Android app. The web application accomplished every feature we set out to accomplish through Wi-Fi, and allowed the user to connect to it using any device (iPhone, Android, computer). Pictures of the web application UI and the Android app UI are included in the verification section.

Camera Module

The camera module interfaces heavily with the Wi-Fi module. The ArduCAM Mini camera takes a photo of the thief and uploads it to the web application via a CMOS OV2640 image processor and Wi-Fi connection. It is angled upwards on the physical enclosure to capture a picture of the thief's face, not their lower extremities. The camera is triggered with the motion sensor, meaning if the motion sensor detects significant movement, a picture is taken and stored on the on-chip memory. The picture is only uploaded to the web application when the alarm is triggered. If pictures do not get uploaded to the web application, they are automatically cleared from the on-chip memory when it reaches its memory limit.

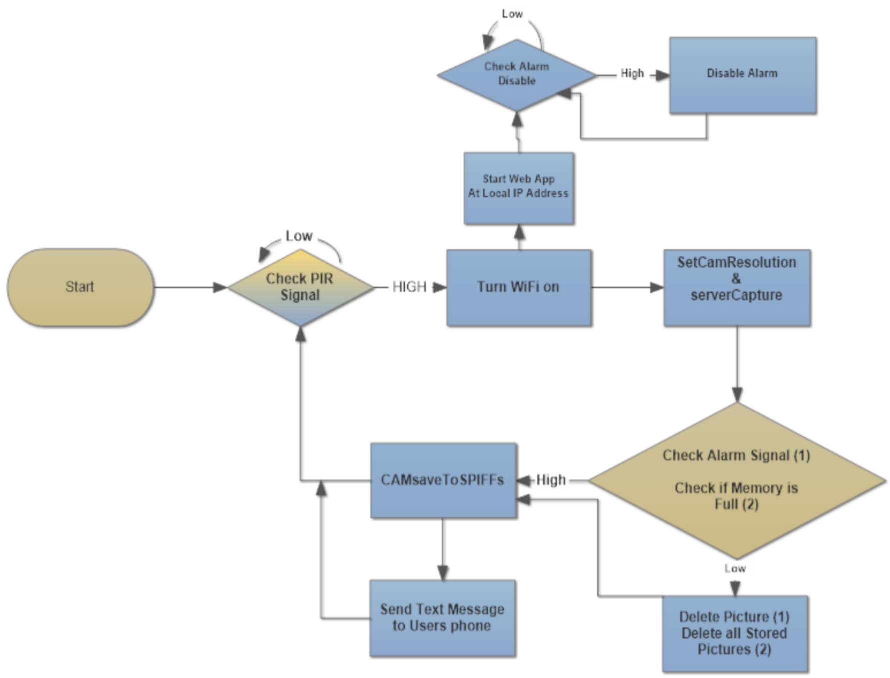

Figure 13: High Level flowchart of WiFi/Web App Software [16-18]

3. Design Verification

A significant portion of this project was spent testing and verifying each module of our project first individually, then as an integrated package. While the individual testing of most modules was straightforward, integration of the whole unit was much more difficult then expected. This was primarily due to the ultra high gain of the weight-sensing module picking up noise from other components. Shielding and clever algorithms largely solved this issue, and in our verification of the weight sensor we describe this in more detail.

3.1 Core Atmel + PIR +RFID Verification

The Atmel was verified implicitly in the proper functioning of the whole unit, while the RFID was verified by disabling the alarm and PIR by triggering the voice module when armed. Due to length constraints, we will not show any quantitative data for this since the main verification is simply that it worked as expected.

3.2 Weight Sensor Verification

Testing, verifying, and optimizing the weight sensing circuit was by far the most time intensive process of this project. Issues with noise from static electricity and other electronics, parasitic capacitances, and power draw from other components greatly increased the sophistication of our design and required a lot of work to solve. Extending the sensing capabilities to discriminate weight differences as small as 0.2lb while developing a robust defense against false-triggering also took significant effort.

The first step we did was to verify our overall load cells and instrumentation amplifier circuit using a breadboard and very basic scale setup. We then tested the raw full-scale voltage swing of the system under full 50lb loading and varying levels of filtering, the results being shown in figure 14 below

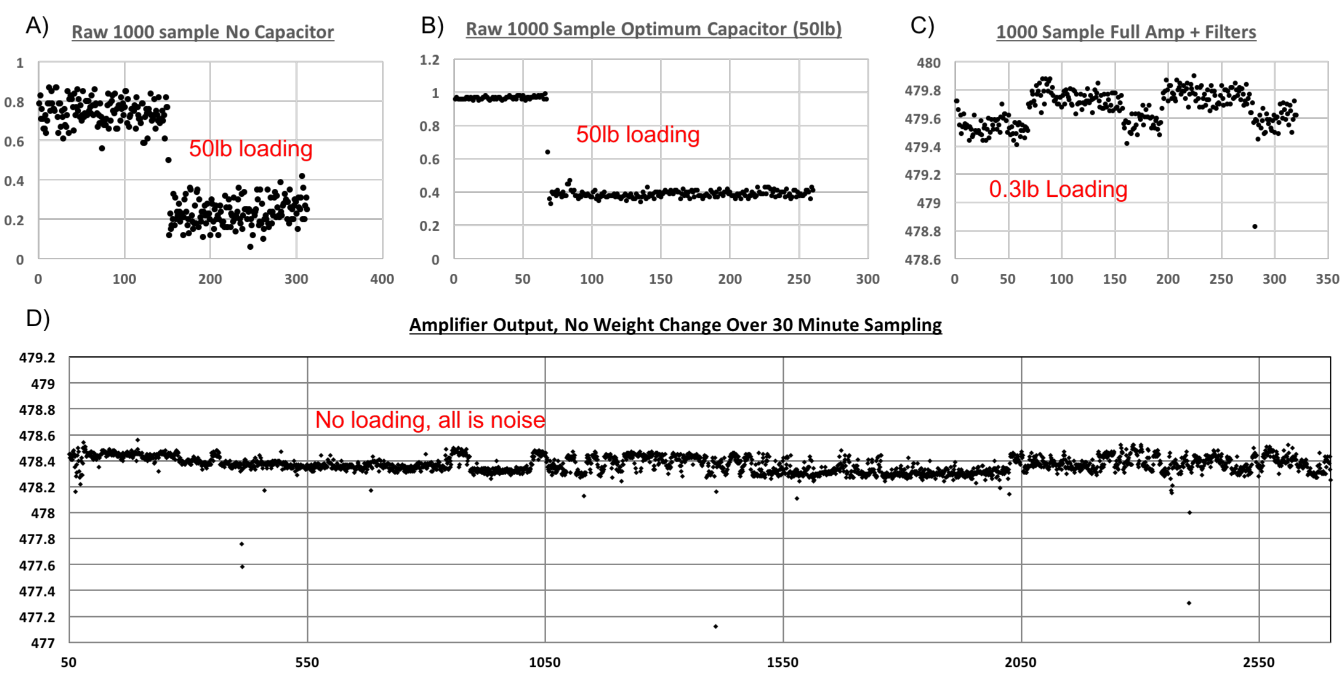

Figure 14: Different tests of the breadboard amplifier and scale.

Figure 13 shows several interesting aspects of the load cells themselves and the amplifier. The first thing to note from Fig. 13(A-B) is how noisy the load cells seemed to be, without any other electronics involved. This inspired us to put filtering on the input side, which improved results dramatically. Once the amplifier circuit was added, we were able to clearly see loading down to 0.3lb as well in Fig. 13(C) which was a very good validation of our design. This data was very noisy however, and a 30-minute run of it without loading revealed very concerning drift, noise, and bistable behavior. This behavior is seen in Fig. 13(D). We decided that this was likely due to it being on a breadboard so more susceptible to noise, and that the PCB would significantly improve the circuit.

Next, we assembled the circuit on the PCB and began further testing, using the serial port as a monitor since that is what our algorithms would eventually see. We saw that noise was significantly improved, but there was still unstable and "jumping" behavior in the circuit, and that the 0.5lb limit was quite hard to distinguish under the noise. When other electronics were turned on in the Slate Safe this got even worse, with massive voltage jumps when they turned on and off. This is seen in the top of Fig. 14. Once proper shielding and algorithms to reduce turn on power fluctuations from other modules were introduced, this noise reduced significantly enough to see weight changes as small as 0.1lb, seen in the bottom of Fig. 14.

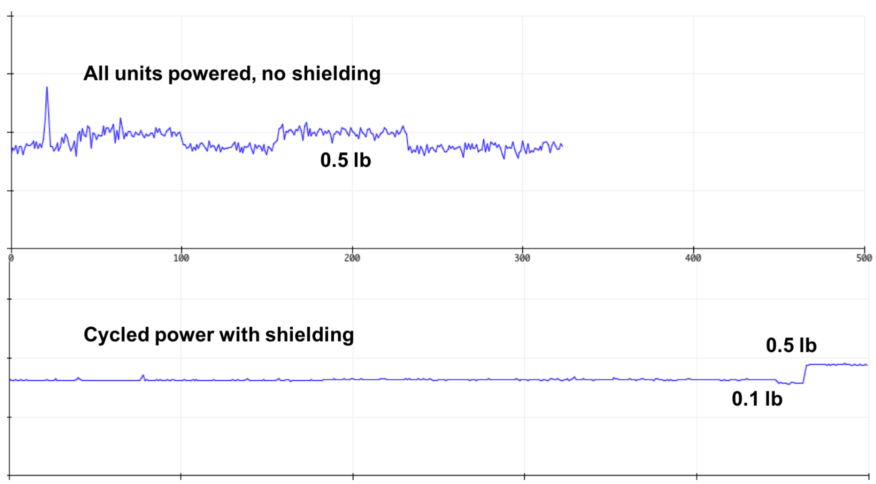

Figure 15: Noise levels with and without shielding and power stabilization.

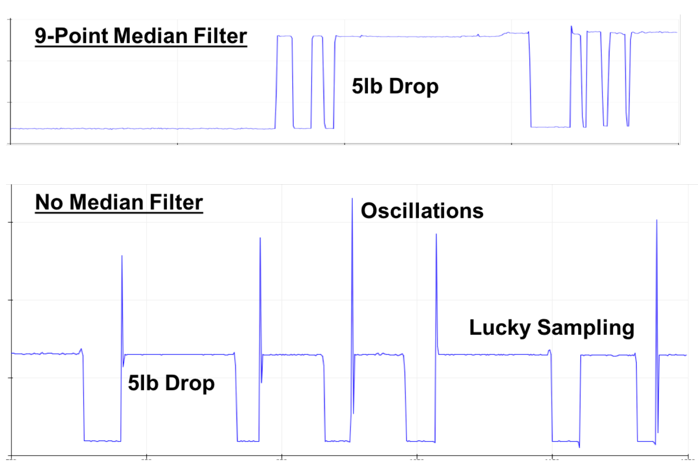

Figure 16: Data from dropping a 5lb package from 3 inches above the Slate Safe.

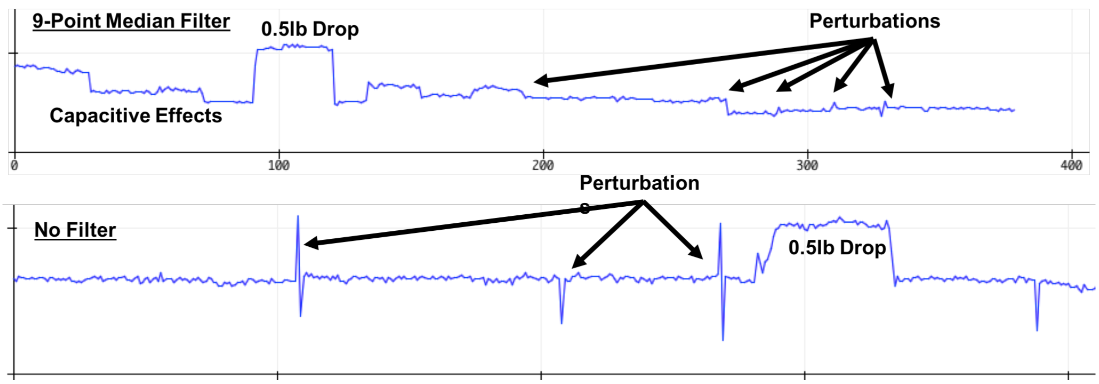

Figure 17: Graphs of the perturbations caused by dropping a 5lb package from 1 foot in the air right next the the Slate Safe.

After improving the hardware, we began testing the Slate Safe under different conditions. It was easily able to detect a 0.5lb package that was gently placed, so then we began testing and writing algorithms to detect more difficult cases such as carelessly placed packages, packages sliding around, big vibrations, and trying to trick the Slate Safe by swapping packages. We tested many different algorithms, and the best was the median filter algorithm described in the design section. Figures 15 and 16 show the substantial improvements of this algorithm in the cases of dropping a 5lb package carelessly on the Slate Safe and dropping a weight right next to it to cause vibrations. These improvements validated our design and were directly verified in testing of the module.

3.3 Audio Module Verification

This module was quite simple to verify, and consisted of checking that both the siren and voice could play from the same speaker, that the siren was loud enough, and that the voice warning was clear enough. We also wanted to verify that the alarm would be triggered when the Atmel shut off or the alarm circuit was disconnected.

Verifying that the alarm went off when unplugged was as simple as unplugging it, and in fact became quite irritating since every time we uploaded new code it would go off. Both modules correctly playing through the speaker was implicitly verified by the proper circuit operation.

The graphs in figures 17 and 18 are directly from the Decibel X application on the Apple App Store [22]. In figure 17, we verified that the alarm was as loud as required. One can clearly see different modulations of the waveforms from the different siren noises, and though it is hard to see, all are above the 85dB point at around 100dB. For the final demonstration, we significantly reduced the noise for safety, though never recorded another nice figure showing all of the different siren type volumes.

Next we tested the voice module. Using a microphone, we first directly uploaded the sound of several claps into the recording module and Decibel X application simultaneously at increasing volumes. We then played those claps back from the chip, observing much higher background noise and no increase in clap loudness. We then uploaded speaking tones from the computer for the final product, and verified this simply by loading the Slate Safe then walking in front of it to trigger a voice.

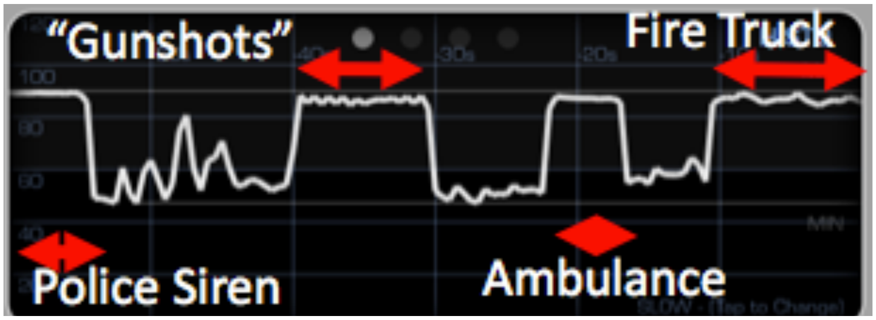

Figure 18: Demonstration of the siren's loudness with different outputs, measured from 1 meter away.

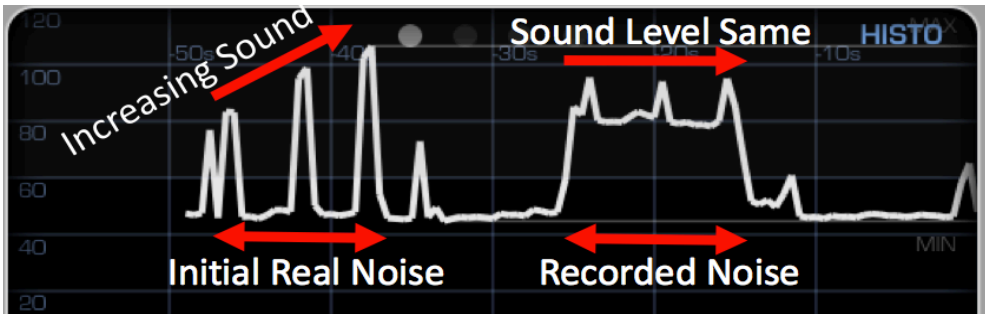

Figure 19: Verification of the voice recording chip.

3.4 Power Supply Verification

The verification of the charging circuit rested on three criteria. The first was that the solar cells were capable of producing enough voltage to charge our circuit even on cloudy days. Given the 3 batteries in our design are "fully charged" at around 3.8V, and that the protection diode and control MOSFET have drops of around 0.7V together, that would mean at least 4.5 volts are required on a cloudy day to fully "trickle charge" our device. Measuring this value, we saw ~6.3 V on a sunny day and ~4.6 on a cloudy day, confirming that the trickle charge would work.

Second, we needed to verify that our batteries could stably source the required power then recharge. We tested this by discharging our batteries though a 50 Ohm resistor overnight, sourcing approximately 100mA constantly for 15 hours. The batteries, though warm, never overheated to failure. Then we put our cells under an indoor bright lamp in order to recharge the batteries fully and check whether or not the circuit would overcharge them. We periodically checked the battery voltages, resulting in the graph in figure 16. We charged 4 batteries simultaneously, since this was more difficult then charging 3 and we believed we might need all of them in the final design. This graph also verified the overcharging protection circuit. The voltage divider setup worked quite well at charging to 4.9V, and though the voltage fluctuated somewhat stayed within 5% of this value which was more than the required accuracy.

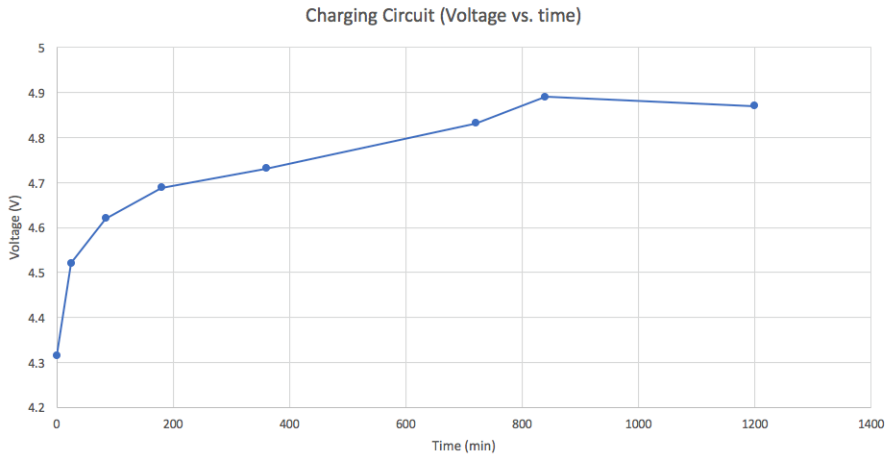

Figure 20: Charging of batteries from fully depleted to the maximum voltage set by our microprocessor.

3.5 Application Block Verification

The verification for the web application involved combining the camera and texting code, running it on the Arduino IDE, and checking print statements on the Serial Monitor to indicate whether the ESP was connected to the network. If it was, we could ensure the ESP was communicating with the web application, and if it was, we simply typed the IP address in the address bar and verified that all the features worked. For example, we checked if the alarm disable button worked by setting off the alarm and pressing the button on the web application.

The user interface for the Android application is shown in Figure 18 on the right. I successfully added buttons, but could not configure it properly to the ESP8266-12E chip.

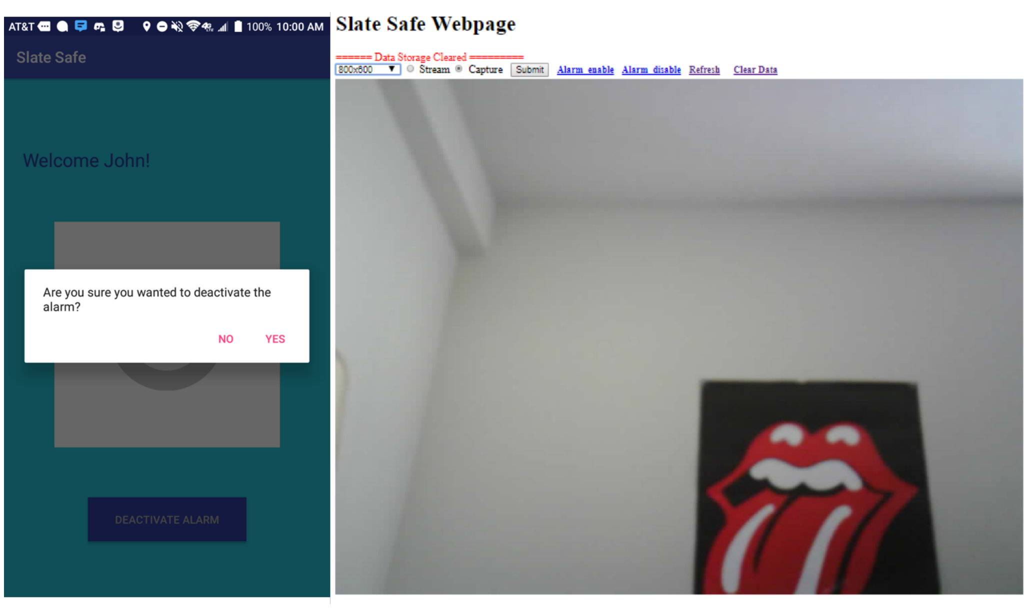

Figure 21: Web application (left) and Android Application (right).

4. Costs

4.1 Parts

Table 1: Cost of all parts together. Note that the optional Camera and Wi-Fi board make up the majority of the cost, so the final version excluding them will be significantly less expensive.

Part	Manufacturer	Retail Cost	Bulk Purchase Cost	Actual Cost
Camera	ArduCAM	\$25.99	\$20.99	\$25.99
Wi-Fi Board	Espressif Systems	\$8.99	\$6.97	\$8.99
PIR sensor	DIYmall	\$1.80	\$1.19	\$1.80
Microcontroller	Microchip/Atmel	\$2.50	\$1.80	\$2.50
Passive Elements	Various	<\$5	<\$1	<\$5
Poly-Si solar cells	Aoshike	\$0.25	\$0.10	\$0.25
LM1117 Voltage	Texas Instruments	\$0.99	\$0.40	\$0.99
NIMH Batteries	Panasonic	\$2.00	\$1.50	\$2.00
Load Cells	Galoce	\$10.00	\$6.00	\$10.00
Op Amps (4)	Microchip	\$0.67	\$0.30	\$0.67
Audio Op Amp	Mouser	\$0.77	\$0.33	\$0.77
Voice Recording IC	Sunkee	\$0.60	\$0.60	\$0.60
Alarm Chip	Spiratronics	\$0.99	\$0.75	\$0.99
Total			**\$41.93**	**\$56.44**

4.2 Labor

An ECE Illinois graduate can earn around \$40 an hour working in industry, so we assume that each of our three team members' time is valued at \$40 an hour. Given that the first two-three weeks were spent going to lecture and getting a handle on the expectations of ECE 445, we estimate that it will take 13 weeks to deliver a working prototype. We also assume that each of us works 15 hours per week. This leads us to a final labor cost calculated by:

$$3[\text{people}] \times \frac{15[hrs]}{[week]} \times 13[weeks] \times \frac{40[\$]}{1[hour]} \times 2.5 = \ \$58{,}500$$

4.3 Schedule

Table 2: Schedule followed.

Week	John S.	John G.	Joe B.
2/5	Design the audio, weight sensing, and MCU	Brainstorm key functionalities we want to have in deluxe version	Research camera boards with memory units.
2/12	Determine the best RFID reader, Atmel chip, and parts for purchase	Look into energy storage and recharging options, investigate app development	Find parts to purchase for camera, PIR, and Wifi.
2/19	Create first round circuit schematics for audio, weight, and MCU	Do power consumption calculations, purchase power components.	Design camera unit and Wifi unit. Purchase all necessary parts.
2/26	Breadboard weight measurement circuit, voice circuit, and alarm	Work on app. Look into ways to minimize power consumption on board	Set up experiments to test requirements of parts.
3/5	Test audio and weight circuits on Arduino, test amplifer	Continue app development. Test battery recharging circuit on a	Estimate time of arrival for parts. Start using parts on
3/12	Debug code, alarm, and weight modules. Finish final PCB design.	Test and verify the remainder of the power circuit as described in the R and V table. Finish PCB.	Optimize design. Finish code to communicate between boards if possible. Debugging
3/19	**Spring Break**	**Spring Break**	**Spring Break**
3/26	PCB arrives. Build core processor first and debug.	Debug power circuit. Continue app development, work with	Finish up all software involved in camera and W-fi
4/2	Finish soldering all weight and sound modules with microcontroller triggering.	Develop app, test solar cells, finish CAD and 3D print it. Write MCU software.	Work on configuring Wi-Fi-Camera module with John G.'s application.
4/9	Test PCB. Optimize audio gain and load sensitivities	Test and debug application and PCB. Write software for MCU.	Integrate camera and Wi-Fi board with rest of Slate Safe
4/16	Test, debug with full integrated package	Test and debug application, add bonus features, test MCU software	Test, debug groups design. Add any additional. features.
4/23	Test, debug with full integrated package	Prepare final presentation and final report	Test full design.
4/30	Present project and finalize report for submission		

5. Conclusion

5.1 Accomplishments

In the end, nearly all of our requirements were met except for a minor current requirement for the main Atmel chip. We clearly documented this success with videos for each element of our RV table. During the beginning of the demonstration, our produce worked perfectly as well. However, after taking it apart to show the contents clearly and reassembling, it failed to work. Simultaneously we had a massive issue with the laptop we were using

During this project, our team was also able to receive significant media attention from the College of Engineering and The Daily Illini, shown below in figure 22.

Figure 22: Media attention from the College of Engineering and The Daily Illini.

5.2 Uncertainties

Although our project was successful, we failed to achieve full demo points during the in-class demo due to a faulty soldering connection in the front of our enclosure, which falsely triggered the alarm. At the beginning our system worked very well, but after taking it apart to show off the internal hardware and putting it back together, it suddenly stopped worked. Eventually we learned this was due to two soldered wires that were constantly being bent when the unit was assembled, that eventually broke. In bringing this product to market, we need to use more robust electronic connections such ribbon cables connecting parts to avoid the mess of wires and decrease the odds of failure. Another small issue we ran into was the connectivity of the Wi-Fi chip; sometimes, the station Wi-Fi failed to connect to certain networks. In the future, we will set up the Wi-Fi chip in AP mode so that it does not have to rely on connecting to outside networks. Rather, it will connect to the internal network built into the chip.

5.3 Ethical considerations

In completing this project, we had several ethical concerns dealing with safety and ensuring we acknowledge that a similar product to ours already exists on the market. From the safety perspective, three issues were most prevalent. The first dealt with ensuring our alarm was safe for all people, and that it wouldn’t be so loud as to cause hearing loss in users or even thieves. We did this by maxing our volume at 85dB, below the threshold for hearing loss, and ensuring the volume was low when testing it. The second concern stems from waterproofing. Outdoor electronics pose a legitimate safety risk, and we intend to mitigate this risk by following ISP68 waterproofing guidelines in the Slate Safe. Third, the electrolyte in the Ni-MH batteries is a strong colorless solution which is corrosive and capable of causing skin damage if leaked. We should stop using the batteries immediately if it leaks.

In doing market research for this product, we found a similar product idea to ours, a device called the “PackageGuard” [19]. In doing this work, we must make sure to adequately cite any influences from their ideas, and inform the ECE445 staff of this product, which we have done in previous reports.

Lastly, by designing and building this device, we have gained a better understanding of the principles of electrical engineering. Therefore, IEEE Code of Ethics #5 is central to the nature of this project: "to improve the understanding of technology; its appropriate application, and potential consequences" (IEEE Code of Ethics #5) [20].

5.4 Future work

With the funding won from startup competitions, we plan to continue with Slate Safe into the summer, and eventually put the device onto Kickstarter. As of now, we are doing in depth market analysis to determine what people want, reaching out to manufacturers to get pricing estimates, and figuring out what makes a successful Kickstarter. We are also looking into alternative anti-theft concepts to make our product more discrete and/or more effective.

Appendix A: Requirement and Verification Table

Requirements	Verification	Worked?
When the alarm is triggered and all parts of the circuit are loaded, the batteries must successfully unload enough current for all functions to operate	Charge batteries. Given open circuit voltage measured using a voltmeter, calculate the needed resistive load to source a current of 400 mA. Discharge the batteries through this load for 1 minute, measuring the current with a DMM, and ensuring that the current stays greater than 400mA for the entire duration.	**Y**
In optimal conditions (sunny day), the solar cells in series must have a voltage > 4.8 V	On a sunny day, measure voltage output of series/parrallel connected solar cells with a DMM	**Y**
In optimal conditions the solar cells in series must have a current output of >300mA to provide the needed charge for our batteries	On a sunny day, measure current of series connected solar cells with a DMM	**Y**
Safely charges three series-connected NiMH batteries to full charge with solar cell charging circuit	Connect battery charging circuit to three fully discharged batteries in series, after measuring the voltage to be <3.4 volts with a DMM to ensure full discharge. Allow batteries to charge for 5hrs at 300mA (total capacity 1500mAh), and measure the output voltage to ensure it is greater that 3.8 volts.	**Y**
Does not overcharge the batteries	Connect a 6.0V DC Power Supply to the input where the solar cells connect, and attach charged (>3.6V) series connected batteries. Measure voltage and current with multimeter, ensuring that series voltage does not exceed 4V. Next, connect solar cells and repeat.	**Y**
The weight sensor is capable of sensing a minimum weight change of 0.5lb.	Gently apply a 0.5lb weight to the module, then remove and the alarm should be triggered	**Y**
The sensor is capable of sensing over 45 lbs of force without an alarm triggering error	Load apparatus to 45lb, then add another 5 lb which we then will remove to trigger the alarm	**Y**
A package weight of 1lb is capable of waking the apparatus from sleep mode	Apply 1lb weight to apparatus, then walk in front of the apparatus. If the system is out of sleep mode, this should trigger an audio warning to play a message warning of the Slate Safe system	**Y**
A person walking nearby the unit will not trigger the alarm	We will test this by placing the apparatus on a commercially available folding table we bring. We will then apply 5lb or 45lb to the Slate Safe to put it into armed mode, or no weight at all. Then, we will drop a 5lb mass one foot away from the apparatus from a 3-inch height, to cause vibration. In all of these cases, the alarm should not be triggered	**Y**
A carelessly placed package will not trigger the alarm to be set off	First, reset the Slate safe system with an RFID tag, and remove all packages. Then, drop a provided 5lb carboard package on to the Slate Safe from 3 inches above the system. This should not trigger the alarm.	**Y**
The alarm circuit is capable of choosing between at minimum a police siren noise and security alarm noise, either manually or through an application	Demonstration of triggering the alarm under both settings	**Y**
The module plays a warning message when under load and the motion sensor is triggered	Place a 5lb weight onto the apparatus, then walk within 3 feet of the front of the sensor	**Y**

The alarm produces a noise of 75dB to 85dB as measured 1 meter away from the device	Using a decibel meter application (DecibelX), we measure at least 75dB from the alarm module on a phone at a 1m distance and verify it is in our desired range	**Y**
The alarm is capable for playing at least 1 minute continuously	After triggering the alarm by removing weight, time the alarm to 1 minute using a phone timer	**Y**
The core microprocessor successfully enters sleep mode when no package is applied	Placing low voltage on the "sleep" input results in the PIR sensor not giving a verbal warning	**Y**
When the RFID tag is triggered, the system becomes disabled	After flashing the RFID tag, the indicator LED turns on to show it is disarmed, and the package is removed without the alarm triggering	**Y**
The Atmel processor consumes <5mA when active and <2mA while in sleep mode	Test current draw into Atmel processor in both modes using an ammeter placed between the regulation and Atmel Vdd pins	**N**
PIR module outputs digital high signal of >3V to the microcontroller when triggered	Walk towards PIR sensor at a slow walk (1mph), brisk pace (3mph), and jog (5mph) and see if the sensor picks up movement. If it does, the Slate Safe should give an audio warning	**Y**
Sensitivity is adjusted so that the device is only triggered when significant movement is detected 3±0.5 meters from the sensor	Connect PIR sensor to 3.3V source for power. Connect LED to PIR sensor for detection output. Use a tape measure to have subject stand and walk by sensor at predetermined distances in increments of 1 ft. Record when PIR sensor is triggered by observing LED	**Y**
Must take a photo after a capture command via SPI port. Picture is stored on off-chip memory	Send a command to camera to take a picture. Check if picture was stored on off-chip memory	**Y**
Resolution of photo must be >= 1280x960, and we must be able to make out facial features from photos of people standing 3 +/- 0.5 meters away (in high light conditions).	Hook up camera to power supply for power. Use a tape measure to have subject stand predetermined distances of one, two, three and four meters from the camera. Send capture command from SPI interface. Analyze results of each image, record results, making sure to state if subject is recognizable.	**Y**
Wi-fi IC must be able to control the alarm via the developed application	Connect the Wi-Fi module to the same Wi-Fi network as the device using the app. Turn on the developed Android application. While the alarm is triggered, click the "disable" button to shut it off within 15s as measured by a stopwatch.	**Y**
Wi-fi IC must be able to give an alert that the package has arrived via the developed application	Connect the Wi-Fi module to the same Wi-Fi network as the device using the app. Turn on the Android developed application, and ensure internet connectivity. When a package is placed on the module, within 30 seconds warning should arrive via the application as measured by a stopwatch.	**Y**
Wi-fi IC must be able to upload the photo of the thief within 5 minutes	Connect the WiFi module to a laptop or phone "router" sharing its connection. Turn on the Android developed application, and ensure internet connectivity. When a package is taken from the module, within 5 minutes a photo should arrive as measured by a stopwatch.	**Y**

References

[1] "U.S. e-Commerce Market Size 2015-2022." *Statista*, www.statista.com/statistics/272391/us-retail-e-commerce-sales-forecast/.

[2] Weise, Elizabeth. "Package theft hits nearly one-third of Americans. Is video surveillance the answer?" *USA Today,* 20 Nov. 2017.

[3] Smith, Aaron, and Monica Anderson. "Online shopping and purchasing preferences." *Pew Research Center: Internet and Technology*, 19 Dec. 2016.

[4] "ATmega328/P Datasheet" http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

[5] "Ardiuno Mini 05" https://store.arduino.cc/usa/arduino-mini-05

[6] "Reducing Arduino Power Consumption" https://learn.sparkfun.com/tutorials/reducing-arduino-power-consumption

[7] "Force Sensing Resistor Integration Guide and Evaluation Parts Catalog" https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf

[8] "GML670 flat load cell electronic scale parts" https://www.alibaba.com/product-detail/GML670-flat-load-cell-electronic-scale_60674361375.html

[9] Sarah Al-Mutlaq"Getting Started with Load Cells" https://learn.sparkfun.com/tutorials/getting-started-with-load-cells

[10] "Instrumentation amplifier." *Wikipedia*, Wikimedia Foundation, 18 Feb. 2018, en.wikipedia.org/wiki/Instrumentation_amplifier.

[11] "Determining Load Cell Accuracy" ADM Intrument Engineering, 13 Feb. 2018, https://www.adminstrumentengineering.com.au/determining-load-cell-accuracy

[12] "UM3561A Three Siren Sound Generator Datasheet" http://www.e-ele.net/DataSheet/UM3561.pdf

[13] "ISD 1800 Series Datasheet" http://elecfreaks.com/store/download/datasheet/breakout/voice/ISD1800-datasheet.pdf

[14] "ESP8366 DataSheet" https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[15] "If This Then That" https://ifttt.com/discover

[16] "ArduCAM now Supports ESP8266 Arduino Board with WIFI Websocket Camera Demo." ArduCAM.com, 4 Oct. 2015.

[17] Pieter P, "A Beginner's Guide to the ESP8266", The ESP8266 as a microcontroller – Hardware

08 Mar. 2017.

[18] dmainmon, “ARDUCAM MINI ESP8266 WEB CAMERA”

21 Feb. 2017, http://www.instructables.com/id/ArduCAM-Mini-ESP8266-Web-Camera/

[19] “The PackageGuard” https://www.thepackageguard.com/

[20] “IEEE Code of Ethics.” IEEE - IEEE Code of Ethics, www.ieee.org/about/corporate/governance/p7-8.html

[21] RFID Datasheet - http://www.hobbytronics.co.uk/datasheets/sensors/MFRC522.pdf

[22] “PIR HC-SR501 datasheet” https://www.mpja.com/download/31227sc.pdf

[22] “Decibel X” https://itunes.apple.com/us/app/decibel-x-db-dba-noise-meter/id448155923?mt=8