PILLOW
A Universal System For Pill Monitoring

By

John Schmidt

Robert Weiner

Lohitaksh Gupta

Final Report

ECE 445 Fall 2018

TA: Anthony Caton

Team 28

Abstract

This report documents the design, implementation, and verification of our pill monitoring system. The final product is meant to be used as a medical monitoring device that users can connect to their wireless network and receive important data about their daily, weekly, and monthly usage of medications.

Table of Contents

1 Introduction

1.1 Objective

According to Consumer Reports, more than half of Americans regularly take prescription drugs [1], which results in over 150 million Americans managing their health through medication. But relying on a patient to correctly adhere to their routine isn't perfect: Unexpected life events, accidents, and simple forgetfulness can cause a patient to take the wrong amount of their medication. This can create dangerous consequences: The CDC estimates that over 1 million Americans are hospitalized from adverse drug experiences, many of which are preventable[2]. This can manifest in two ways: The first is overdosing, when a patient consumes too large of a dose within a period a time, which can easily cause unintended consequences. Even over-the-counter medication can be harmful or even lethal at high doses[3]. The second way is underdosing, the less visible of the two. This is when too little of a medication is taken, which can reduce the effectiveness of treatment and prolong the illness. An additional underdosing concern can be seen in the use of antibiotics: If a person doesn't follow their antibiotic regimen (for instance, stopping when they feel better and not when they are done with the pills), they run a heightened risk of becoming reinfected with the illness, this time from an antibiotic-resistant strain that is much harder to get rid of [4]. Our project is meant to prevent under- and overdosing in an effective, yet easy to use and inexpensive way.

1.2 Background

Other attempts to solve this problem have been focused on replacing the actual pill bottles with some sort of automatic dispenser. Because of the (often) motorized components involved, these products can be much larger, more power-hungry, and more expensive than our solution. And the cost is the most pressing concern with these solutions - current dispensers on the market start at hundreds of dollars [5]. The components going into our system are inexpensive and will give the users access to an effective and affordable solution to over- or under-dosing.

Our proposed solution was to create a two-module system with a Pill Scale set underneath a bill bottle transmitting any change in weight to a Pill Hub over Bluetooth. The Hub would use WiFi to communicate with a mobile application that would track usage and provide reminders for the user to stay on schedule. The Bluetooth module proved tough to implement (refer to section 5.2.1), and so we opted to combine the system into one module, weighing and transmitting the data in one, wall-powered device.

1.3 High-level Requirements

i. The 'Pill Scale' will count the number of times a medicine is taken and at what time, transmitting that data to the mobile application.

ii. Users will be reminded daily to stay on their schedule and will receive alerts in the form of push notifications and alarms if they dispense the incorrect dosage. These notifications can be scheduled by the user and can occur simultaneously on their phone or through the speakers and lights on the hub module. Additionally, doctors connected via the application will also be alerted for any deviation from the schedule.

iii. The user's activity will be monitored and shared through an app (to both the user and doctor). This analytics will be quantified by charting usage, which can help both for regular routines or for medicines which don't have fixed dosages (like lactose pills). Doing so will help inform doctors and their patients about the effectiveness of their current pill regimen.

2 Design Verification

2.1 Block Diagram

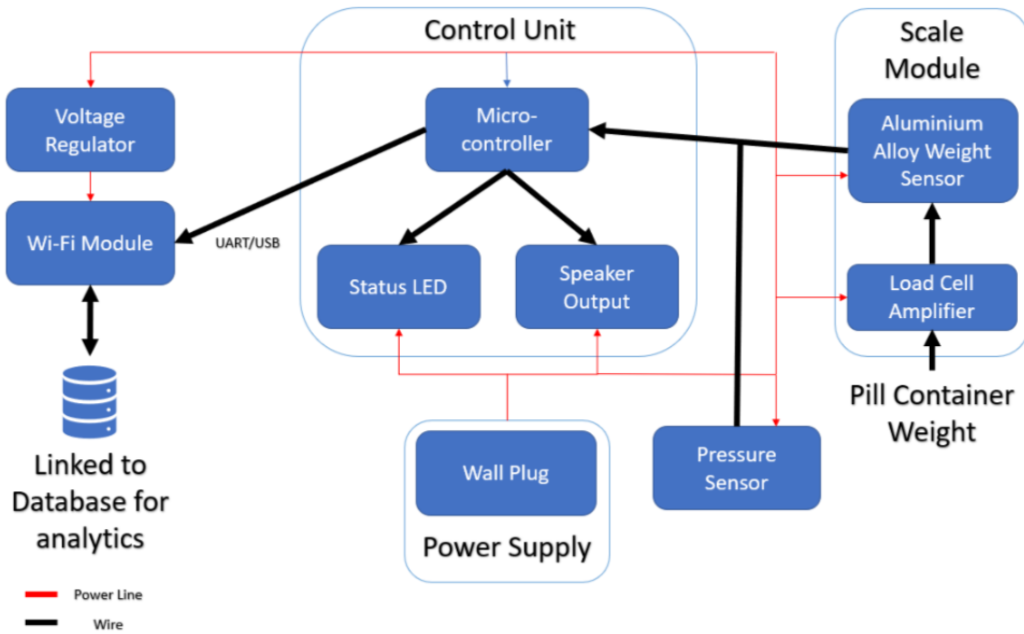
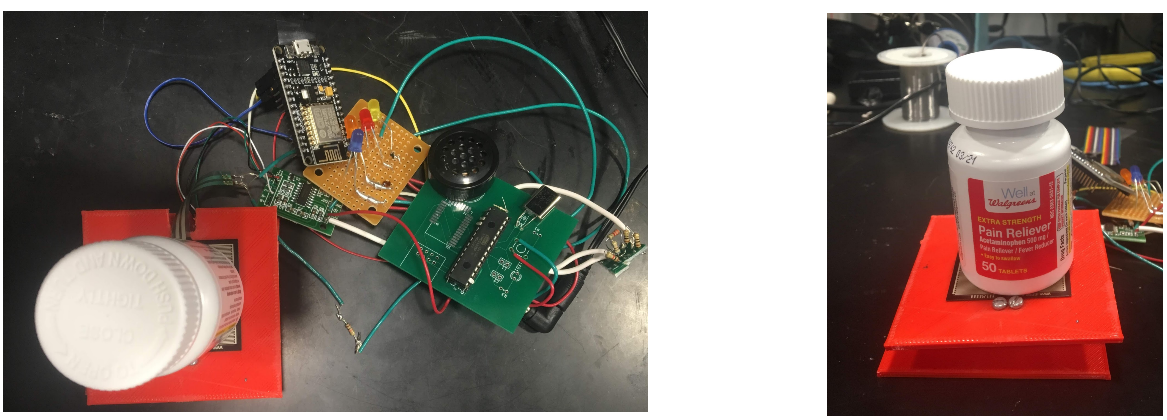


Figure 1: Revised Block Diagram for Pill Scale

2.2 Mechanical Design



‘

Figure 2: Complete Pill Scale

2.3 Software Flowcharts

2.3.1 Main Program

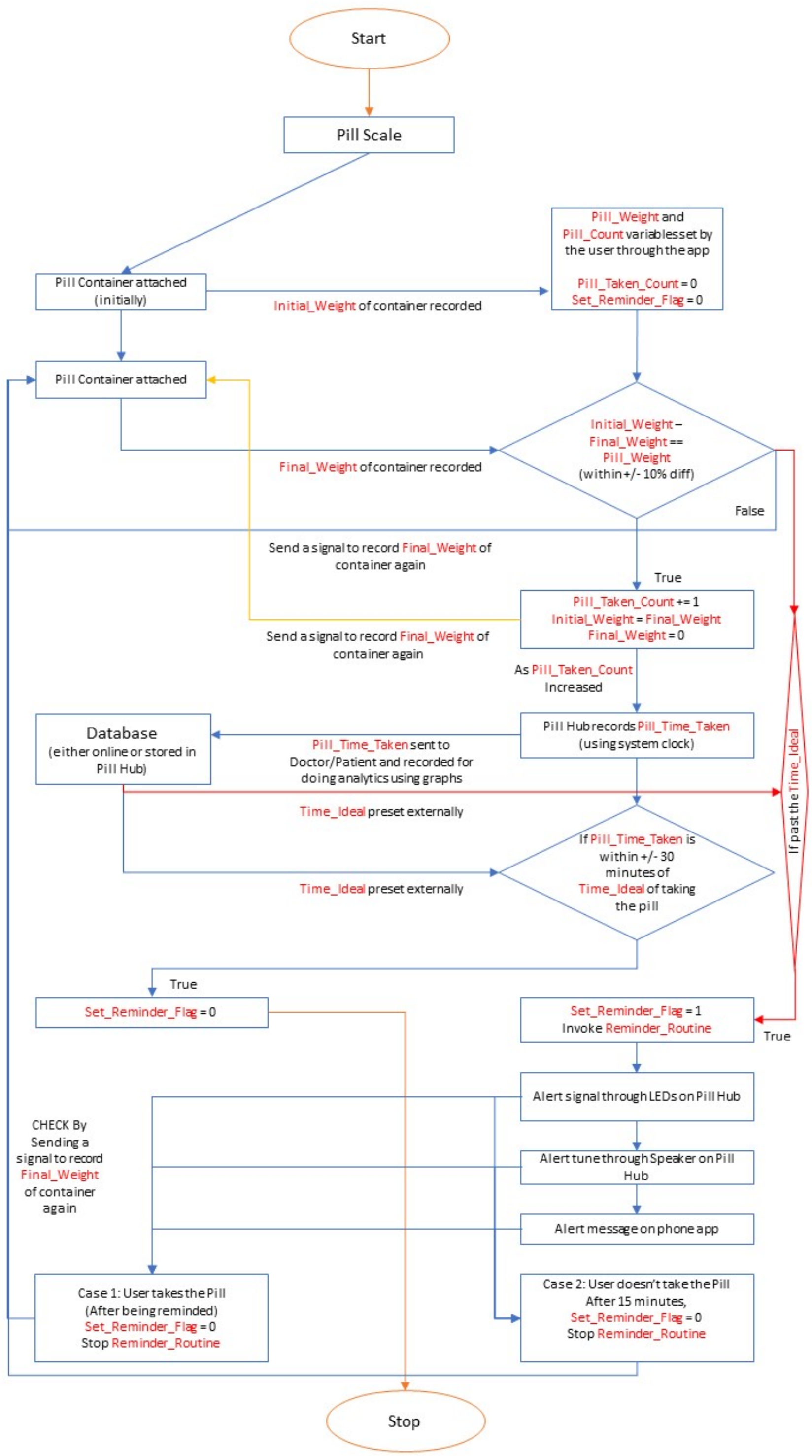


Figure 3: Main Program For Detecting Change In Weight

2.3.2 Server Communication

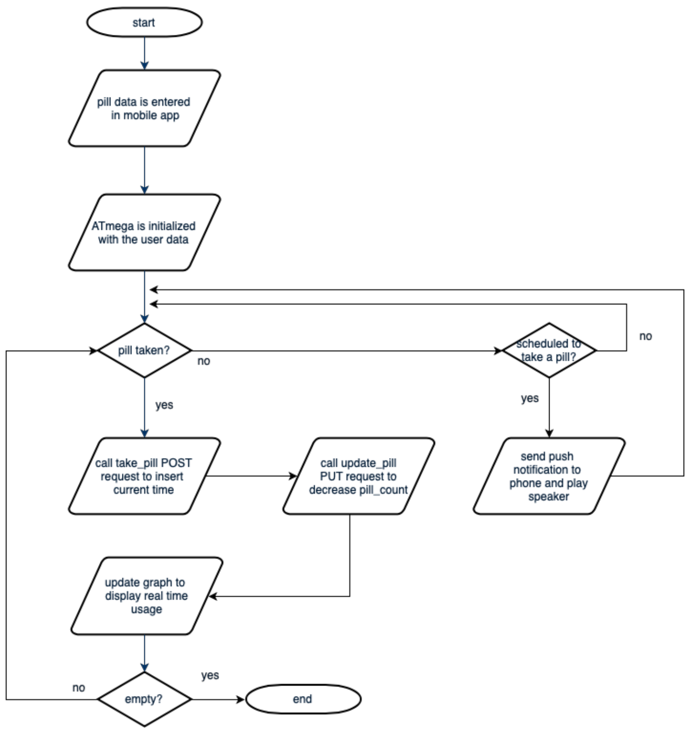


Figure 4: Communication between server, app, and Wi-Fi module

2.4 Schematic

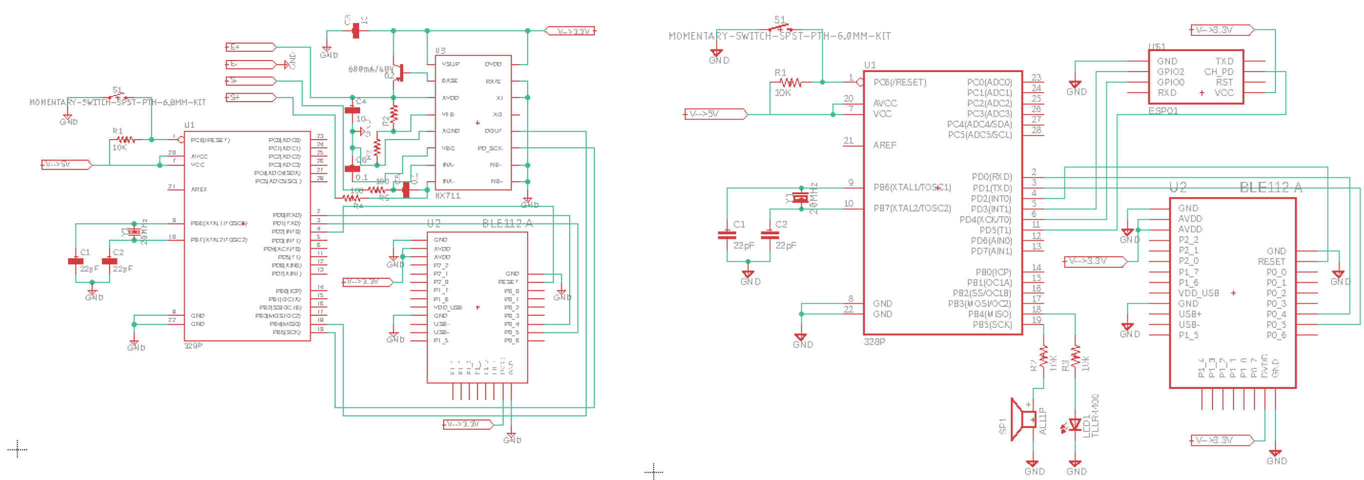


Figure 5: Schematics for Original Scale and Hub modules

2.5 Tolerance Analysis

In order for this system to work appropriately, the most important element of tolerance comes down to the scale and the modules that end up sending that input into the microcontroller. To start, we need to define the desired tolerance for the scale itself. For this design to work, the scale needs to have enough resolution to accurately measure when even a single pill is removed from the bottle. Finding the actual weight of a singular pill is difficult, especially since the quoted weights for most medication is actually the strength of the active ingredient, not the weight of the pill itself. For instance, this source [6] displays low-strength Xanax pills, which come in strengths of 0.25 mg, 0.5 mg, and 1.0 mg in pills of similar size (and clearly much heavier than a single milligram). Therefore, quoted "weights," as demonstrated, don't give a clear picture of the weight we should expect. Further, there doesn't seem to be a widely available reference for the actual physical weight of any pill, so getting a clear picture of an average or a range of values for pill weight would require us to personally test and weigh a wide variety of pills. This wasn't realistically feasible for just this prototype design, so we opted to instead calibrate the system to a single bottle of acetaminophen we had on hand. We used an electronic scale to measure ten pills individually. Below is a table of those trials.

Pill	Weight (g)
1	0.582
2	0.581
3	0.586
4	0.577
5	0.577
6	0.580
7	0.579
8	0.583
9	0.584
10	0.580

Table 1: Pill Weight Trials

These trials resulted in a calculated average of ~581 mg per pill (for reference, the strength was advertised as 500 mg). So, the next step was to use this information and devise an appropriate tolerance for our scale. The actual pills differ by only a few milligrams, so we decided to give ourselves a somewhat large margin of error of +/-100 mg. This gave us much more time to focus on the integration of our system over calibrating the scale to some minute level. But most importantly, this is more than enough tolerance to ensure that each pill removed registers a distinct difference in weight.

With our resolution selected, we now needed to choose components for our system that could be calibrated within this range.

2.5.1 Micro Load Cell

This sensor is doing the heavy lifting of the system (literally), so we needed to make sure that we chose the right one. To keep the footprint of the design small, we opted for a 100 g micro load cell. 100 g is a perfectly suitable capacity for this project, given that this is enough to comfortably hold a pill bottle and dozens of pills - the almost-full bottle of acetaminophen we used to test our design weighed only 32 grams. The specifications for this specific load cell[7] list plenty of additional notes on the various tolerances of this load cell, but resolution isn't one of them. According to this primer on load cells[8], resolution truly isn't a property of load cells, but instead of the equipment used to measure them. Indeed, by itself the load cell cannot be used as an analog scale. In a simple test of the cell's output (mostly to confirm that it was registering any change of voltage), the VCC wire was connected to 5 V, and the GND wire to ground. When probing the O+, or positive output, of the cell, the output voltage was ~2.81 V, with some noise in the millivolt range. After applying a significant amount of force onto one of the cell, the output read 2.79 V, again with noise in the millivolt range. This difference was nowhere near adequate to parse the difference between a single pill. Indeed, even the entire bottle registered a negligible change in output voltage. This was expected, which is why we needed to introduce an amplifier to the equation.

2.5.2 HX711 Load Cell Amplifier

This module is used to actually increase the resolution of the load cell. By feeding the output from the load cell into the Channel A of the amplifier, we realize a gain of 64, resulting in a differential input voltage of ±40mV. This newly amplified signal is passed out of the amplifier as a 24-bit 2's complement number. With this amplifier in place, the tiniest change to the weight detected by the load cell can be measured. There isn't much we could do with this bit of hardware to realize the resolution we want, since the previously analog signal was now translated into a digital signal. Experimenting with the different gain values to change our resolution would have been possible, but would require an entirely different PCB and a lot of work for the potential of a better resolution. With this being said, the output from the amplifier is fed directly into the microprocessor to be calibrated.

2.5.3 ATMega 328p

The final step in extending the tolerance of the load cell is in the programmable microcontroller itself. At the software level, the scale system needed to be calibrated further. The scale correction can be calculated from the following equation, derived from here[9].

$$Correction = (outputRating * gain * ADCRange)/maxRange$$

Where "outputRating" is the output from the load cell, gain comes from the amplifier (64), ADCRange is the range of values for the ADC output (2^24 -1), and the maxRange is maximum rating for the load cell (100g). This equation allows us to get a baseline for the necessary correction. The next step was to use known weights, in this case the pills we had already measured, and calibrate the software to accurately read the weight. We first used one of the pills of known weight, placing it on the scale and recording the average weight returned by the system. Once we had calibrated the scale correctly for the single pill, we incrementally added four more pills and recorded the minimum and maximum seen result over 20 seconds of monitoring. As shown by the graph below, we were able to show a tolerance of around +/- 0.1 g, which exactly fit our requirement for this tolerance.

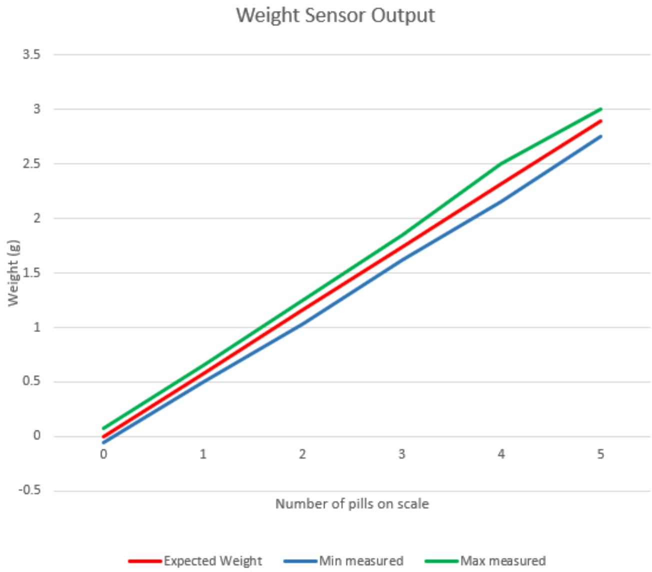


Figure 6: Output from weight sensor from pill test

With these three components, comprising both the hardware and software portions of this system, we were able to correctly calibrate the scale to respond to weights with a resolution of +/- 100 mg. Given the pills we used to create this prototype were almost six times that value, this satisfied our requirements and allowed us to continue with testing the functionality of the other elements of the design. As a future consideration, more resolution in the scale would be extremely helpful. Pills come in a variety of weights and sizes, so more resolution is almost always going to be a good thing in ensuring that the system recognizes when the difference in weight is so small. But for this prototype, our analysis concludes that we have a sufficiently operable scale.

3 Cost and Schedule

3.1 Labor

We assume a labor cost of $35 per hour. With three partners working approximately 10 hours per week for the next 10 weeks left in the semester, we come out to the following equation:

$$3 * \$35/hr * 2.5 * 10\,hrs/week * 10\,weeks = \$26{,}250$$

3.2 Parts

We are only factoring in the cost of building a single prototype. Buying these parts in bulk and introducing some form of production line would greatly reduce the costs associated.

Part	Unit Price	Quantity
Microcontroller (ATMega 328p)	$2.14	2
Wi-Fi Module (ESP8266)	$6.95	1

Bluetooth Module (BLE 112-A-V1)	$11.89	2
Load Cell	$8.48	1
Load Cell Amplifier (HX711)	$9.95	1
3V Lithium Ion Battery	$1.60	1
Total	**$55.04**	

Altogether, most of the cost of this prototype will be realized in labor, for a total cost of $26,305.40.

4 Design Verification

4.1 Power Supply

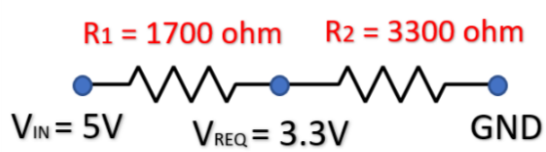
4.1.1 Wall Plug Adapter

A wall plug adapter was used as a power supply. It was connected to an AC power supply or outlet at home.

Requirement	Verification
1. Input Voltage: 100-240V 2. Output Voltage: 5V	1. Place a digital multimeter in parallel with the output of the wall outlet adapter to measure the voltage 2. Place a digital multimeter in parallel with the output of the wall outlet adapter to measure the current

4.1.2 Voltage Regulator

This voltage divider circuit provides the required 3.3V to the Wi-Fi module. This chip must also be able to handle the maximum input from the power source.



$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Figure 7: Voltage Divider Circuit and Equation

Requirement	Verification
1. Input Voltage: 5V 2. Output Voltage: 3.3V 3. Shouldn't get hot. It should be thermally stable	1. Place a digital multimeter in parallel to measure the input and output voltage 2. Use a digital thermometer and take readings every hour

To verify the thermal stability we set a timer to go off every hour and we took measurements of the current temperature of the component. As denoted in the table below after 3 hours of being in use the temperature of the voltage regulator stays constant at around 65.2 °F.

Time Past (HH:MM)	Temperature (°F) (Room Temperature = 65°F)
00:00	65.00
01:00	65.15
02:00	65.28
03:00	65.40

Table 2: Thermal Stability Verification

4.2 Control Unit

A control unit prepares data to be sent over UART to the Wi-Fi and Scale module and it manages the computation, as proposed in the software flowchart. The microcontroller provides a simple user interface with a status LED and speaker. The microcontroller we chose is the Atmel ATMega328.

4.2.1 Microcontroller

The microcontroller communicates with the Wi-Fi, Bluetooth, and Scale chip via UART and reads the SD card cache via SPI.

Requirement	Verification
1. Successfully enters sleep mode when no user input is detected (i.e. when no module is actively transmitting data) 2. Low Power Consumption 3. Must have enough I/O ports to	1. Using an oscilloscope, check if any signal is being transferred when on sleep mode. 2. Use a multimeter to record the significant drop in current usage 3. Must have enough I/O ports to accommodate Bluetooth and Weight

accommodate Bluetooth and Weight module or Bluetooth and Wi-Fi module	module or Bluetooth and Wi-Fi module

Power Source	State	Current at 5.0V @16 MHz
RAW Pin	Active	19.9 mA
RAW Pin	Down Power	3.14 mA

Table 3: Low power consumption verification table (with sleep mode)

Other modifications that can be done in order to further lower the power consumption:

- No Power LED; On an active state, LED takes approximately 3mA. Thus at Down Power mode, the consumption is approximately 0.15 mA
- Removing the Linear Regulator
  - $P_{wasted} = (V_{in} - V_{out}) * I$
  - Although, without the linear regulator, the input and output power is same.

4.2.2 Status LEDs

The 4 status LEDs will display the following four statuses: WiFi connectivity, Reminder Routine, Pill container on top of Pill Scale, and Power.

Requirement	Verification
1. Clearly visible from over 1m away with a drive current of 10 mA ± 5mA	1. Measure current through LED with a digital multimeter. Ensure current is within the required range (10 mA ± 5mA) 2. When left in a room, with no obstructions of view, measure the distance where the LED is no longer visible (unable to determine what color it is or what color it switches to).

4.2.3 Speaker Output

Speaker will be able to give a scheduled reminder to the user about his or her medicine.

Requirement	Verification
1. Clearly audible at 60 dB (normal conversation level) from over 1m away	1. When left in a room, with no obstructions, measure the distance where the decibel level drops below 60 dB using a decibel meter application found in the App Store

4.3 Scale Module

Scale module reports the weight of the pill container in grams to the microcontroller. It gives a milligram (mg) precision. The scale module is also accompanied with a pressure sensor on top of the scale.

4.3.1 Aluminum Alloy Weight Sensor & Load Cell Amplifier

We will use 0-100 grams electronic scale aluminium alloy weight sensor. The voltage difference resulting from the weight sensor needs to be amplified in order to be quantifiably processed by the microcontroller. SparkFun load cell amplifier (HX711 - https://bit.ly/2nOrBYq) can be used for this purpose. This will give a mg precision.

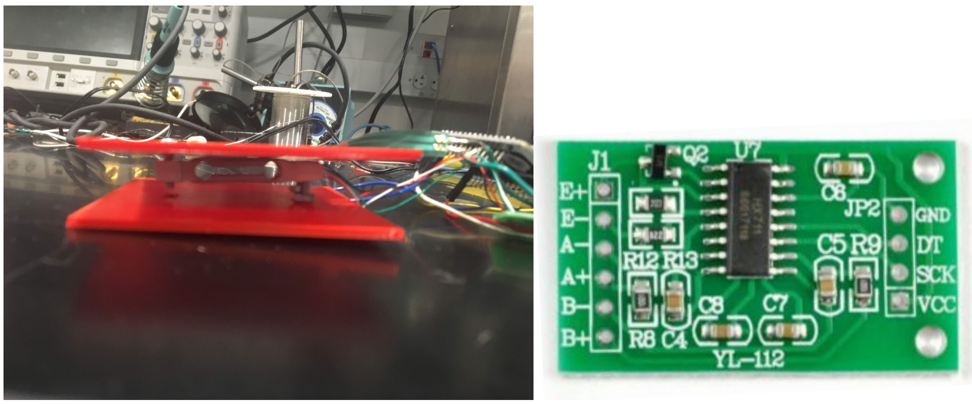


Figure 8: Aluminium Alloy Weight Sensor (Left); HX711 Load Cell Amplifier (Right)

Requirement	Verification
1. Weight sensor responds to stress 2. After connecting the amplifier with the microcontroller, the current through amplifier should increase as the weight increases 3. Weight sensor should be calibrated 4. Should be able to detect mg changes in weight 5. Capable of sensing a weight change of fewer than 0.5 grams	1. A tiny voltage fluctuation is observed with the multimeter 2. Increase in current is measured through the multimeter 3. Calibration is monitored by putting a known weight on the sensor and see if it gives the expected weight through the serial monitor. 4. Observed through the serial monitor 5. Apply a weight greater than 0.5 grams to the module, then remove at least 0.5 grams (one pill, for example). The status LED will flash when a weight of 0.5 grams or less has been removed (indicating that a “pill” was removed from the weight)

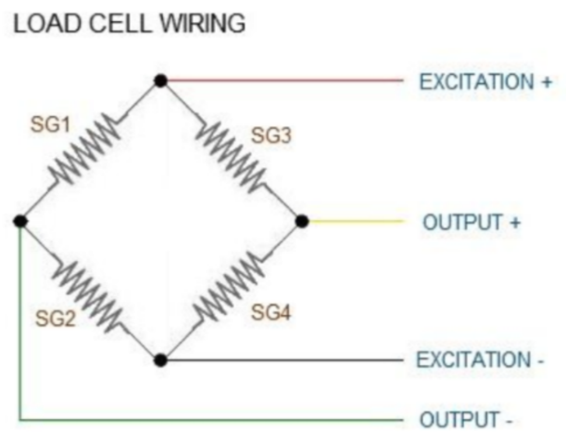


Figure 9: Load cell circuit: Four strain gauges (SG1, SG2, SG3 and SG4) assembled in a wheatstone bridge formation

4.3.2 Pressure Sensor

Pressure sensor helps in putting the microcontroller in sleep mode. Also, it puts weight the sensor and load cell amplifier (HX711) in active mode only when there is nothing on top of it (when pill container is picked up by the user). Overall, this helps in saving energy.

Requirement	Verification

1. Required Coverage area: 1.5”x11.5”
2. Input: 3.5V RM = 10k, ((see Figure 10 for further analysis)
 	
3. Force Test
    a. No Pressure
    b. Finger Pressure
    c. Bottle Filled
	1. Measured using Scale
	2. Works, output: 0V
	3. With Multimeter
	    a. Output: 0V
	    b. Output: 3.5 V (max)
	    c. Output: 2.4 V

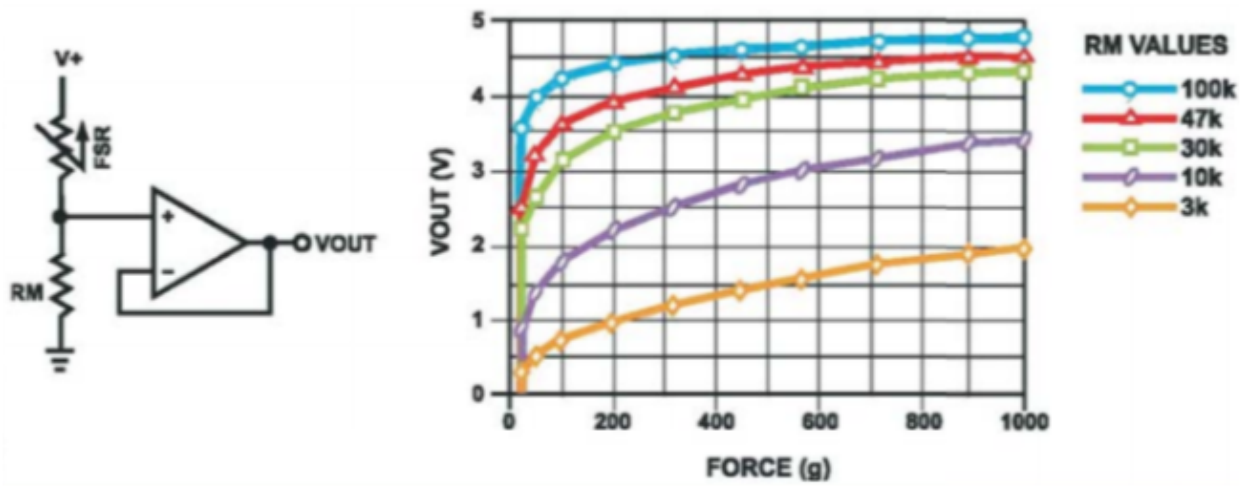


Figure 10: Pressure sensor circuit (Right); RFSR varies in direct proportion to the pressure on top of it. The graph on the left shows the variation of voltage (VOUT (V)) across the RFSR (Ω) with Force (g) for 5 different RM values (Left)

4.4 Wi-Fi Module

Data from the control module is sent via UART to be accessed on a Wi-Fi network. The Wi-Fi IC, the ESP8266, is a self-contained SOC with integrated TCP/IP that gives our microcontroller access to the users Wi-Fi network. The Flash holds the program memory for the Wi-Fi IC.

Requirement	Verification
1. Must be able to communicate over IEEE 802.11/b/g/n at greater than 100kbps 2. Must be able to communicate over both UART and SPI	1. Detect and connect to this Wi-Fi Network from phone and other Wi-Fi enabled devices. If Wi-Fi microchip is IEEE 802.11b/g/n- compatible, it should be connected successfully 2. Check on the receiving side, whether any data was received or not

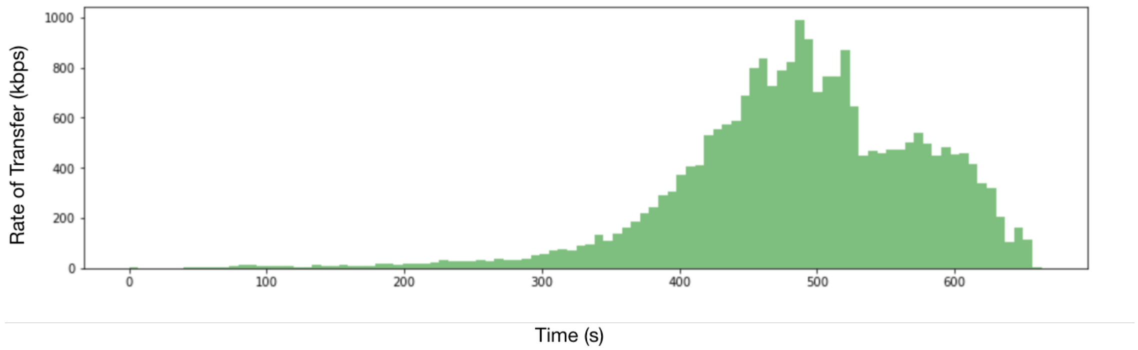


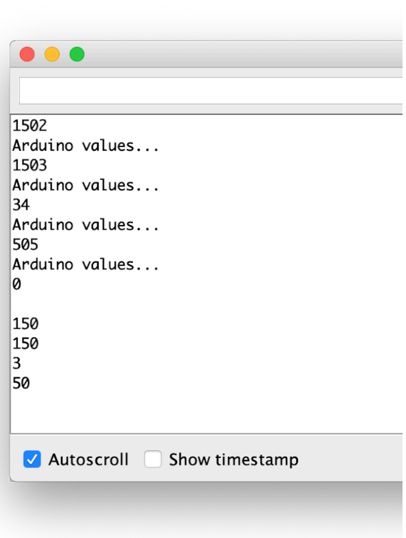
Figure 11: Rate of Transfer test

To verify that the Wi-Fi module could communicate data from the server to the ATmega in a reasonable amount we ran a test to confirm how much data we could retrieve from our server. Towards the beginning of the test we retrieved small sets of data (users unique id) which had a rate of transfer of 550 kbps. With requesting larger datasets (all of the users pill history for the last month) the rate of transfer decreased slightly to 340 kbps. Averaging all of these responses from the server we were received a 490 kbps rate of transfer.

To verify that the Wi-Fi module was transferring data to the ATmega we retrieved the data from the server in the form of a json document. On the ESP8266 we flashed the chip to parse the incoming responses it receives from the server.

In Figure 12, the values are sent to the microcontroller as their value concatenated with the index. With the use of the concatenated indices the microcontroller can easily decode and assign the initial values to begin the main program and start tracking any users pill routine.

Figure 12: Data Received by Microcontroller

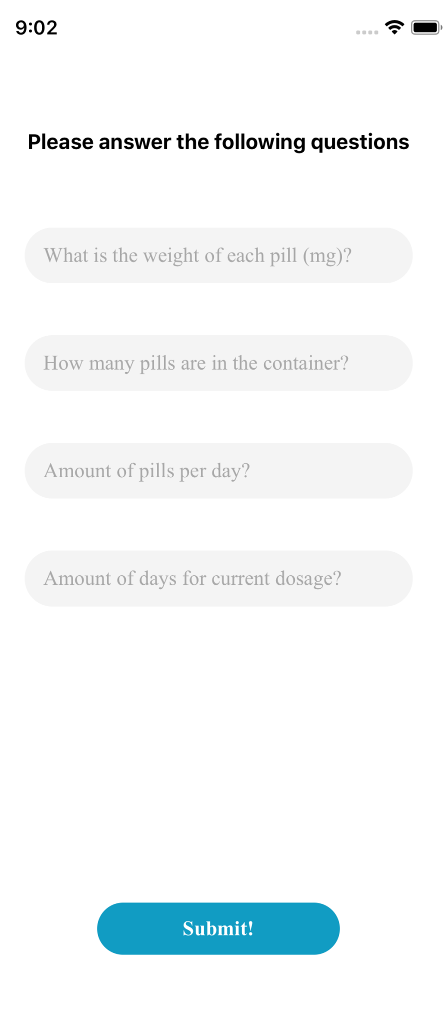


4.5 Server Communication

Requirement	Verification
1. Must be able to POST to a MySQL database with data from the app inputs 2. Must be able to GET most recent pill data entered by the user and then communicate over SPI to the ESP8266 to transfer data 3. Data should be encrypted	1. When filling in inputs from app, check database for the most recent entries 2. When filling in inputs from app, check the outputs to the monitor to confirm they match the most recent data stored in the database 3. Check both sides of the communication on whether the data is encrypted or not

To satisfy our requirements for the server behavior and how it communicates with the ESP8266 we performed some checks to ensure the data sent and received by the server was correct.

Figure 13: App Interface



In Figure 13, the user inputs the corresponding data to the medication that they would like to sync with the PILLOW app. When the data is submitted, a POST request is submitted to store the inputs in a MySQL database where the entry is associated to a unique user index.

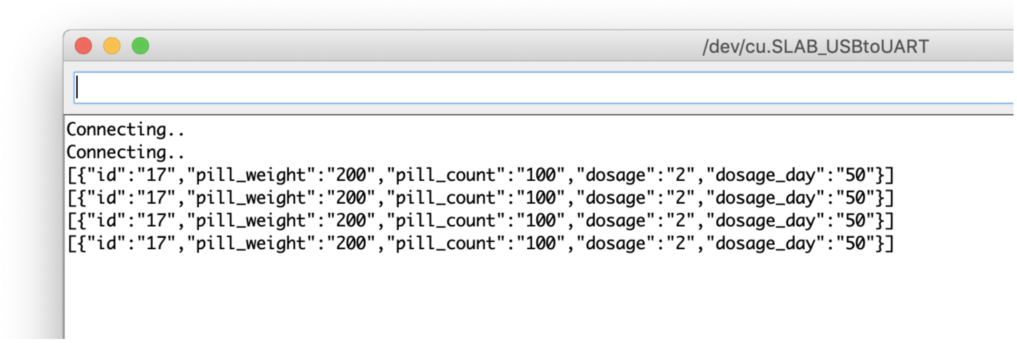


Figure 14: GET Request For Most Recent Pill Data

In Figure 14, the data that was entered in Figure 13, is retrieved from the server and output the serial monitor, this way we can confirm if the received json document is correct. For these requirements, we had a 100% success rate on GET and POST request calls to the server.

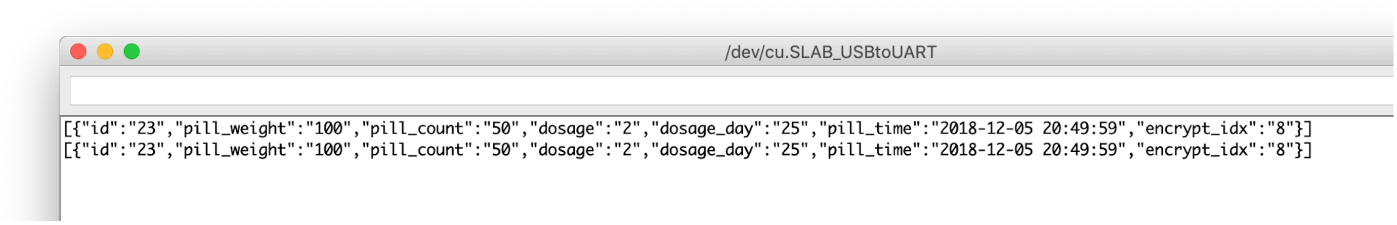


Figure 15: GET Request For Encryption Index

In our most recent implementation, shown in Figure 15, we added a simple encryption layer to our data to ensure that if any Wi-Fi communications were intercepted, the attacker on the other end would not have any use for the data. The encryption works with an array of hex values. One copy of the array lives in the mobile application and another copy of the same array on the microcontroller. When the user inputs a new medication a random hex value is multiplied by all of the inputs and these values are stored in the server. When the ESP8266 submits a GET request to the server it receives an encrypted json document with which it parses and sends the encrypted values to the microcontroller. Once the values are received by the microcontroller it then decrypts the values with the "encrypt_idx" value.

5 Conclusion

5.1 Accomplishments

All the high level requirements for the project were met. We were able to get the data from the the user through the app and store it in our server. The data was encrypted and categorized as per the user's ID. The data received in the server was successfully transferred to the microcontroller via the WiFi module. The scale module and microcontroller were able to successfully detect whether the pill is taken or not, how many pills have been taken and at what time. This data was successfully transferred to the server via the WiFi module and insightful graphs could be populated in the iPhone Pillow app. Also, if the pill was not taken on its appropriate time, the user was reminded about it through the speaker, LED and push notifications.

Overall, our system was able to complete the basic functionality needed to demonstrate its viability as a completed prototype. With the added integration of the mobile application, we have a completed prototype that meets the high level requirements for our design.

5.2 Uncertainties / Issues

5.2.1 Bluetooth Module

Probably the most evident of the issues facing our design is how the final prototype differed from our initial design proposal. The original plan for this design was to have two discrete devices. One was the scale module, which would contain all the components to weigh the pills, and be powered by batteries. The other was a hub module, which would be connected to an outlet and transmit the data through WiFi. The two were to be connected through Bluetooth modules on each of the PCBs. When it came time to configure the Bluetooth components, we immediately ran into issues. In order to set up the modules, we needed to first solder the chips to a breakout board, and then use an Arduino connected to the computer to update the firmware. The problem was that the program that was supposed to do this kept hanging in the middle of execution, and wouldn't complete. We sunk a lot time trying to debug this and add the modules to the final design, but it ultimately was a waste of effort. This was especially distressing since we didn't have time to redesign the PCB, so we had to solder wires to connect the pins correctly. As described in section 5.4, getting this part of the design working is a priority of future work, as it falls in line with the overall vision of the product.

5.2.2 Packaging

And as a further consequence to the difficulty in 5.2.1, removing the Bluetooth modules and replacing them (effectively) with a load of wires made packaging untenable. We had planned to 3D print a simple box along with the two plates we used to secure the load cell, to wrap around the components while still granting us access. With the added physical complexity and overall fragility of the system, we couldn't complete this objective. This is definitely another important aspect of the future work for this project, as it would add not only to appeal of the design, but also the safety and overall effectiveness.

5.3 Ethics and Safety

The most important safety concern with this project is the fact that its use is directly tied to the health of our users. The most glaring hazard revolves around a malfunction somewhere in the circuit that results in an incorrectly perceived output. The main concern here is the resolution of the scale, as determined by the load amplifier and its connection to the microprocessor. As shown in Section 2.5, the tolerance analysis of the scale was very important to the design functioning correctly.

As with any project involving electrical components, exposure to electricity is a safety concern. Luckily the modules we are working with require a relatively tiny voltage to operate, from 3.3 to 5 volts. Still, with the design connected to the wall outlet, the power sources could potentially become dangerous with a wiring mishap. In order to ensure the safety of ourselves and the eventual end user, we followed the guidelines given by the university for electrical safety [11]. Additionally, we have all passed our lab safety training, and followed the rules set out in the University's Lab Safety Guide[12].

Ethically, the most obvious concern with our system is that we are dealing with medical data from our users. We are therefore making ourselves responsible for the health of our users. Both the IEEE [13] and ACM [14] codes of ethics emphasize doing no harm (numbers 9 and 1.2, respectively). In addition, in order to remain compliant with the government's HIPAA Privacy Rules [15] (and 1.6 "Respect Privacy" from ACM's ethical code), we need to ensure that the medical information connected to our users is kept confidential, while still allowing the user the ability to share their information with physicians, caregivers, or pharmacists. Making sure that a user's medical information is under complete control of the individual is tantamount to us making sure that this system follows ethical and legal rules. To do this, we encrypted all of the data in our database. Physicians wishing to participate in this system need the express consent of the patient before being able to access the relevant data from their own, separate account. As an added security measure, we encrypted the information sent to the application, and made sure that it only contained the changing weight of the scale. Doing this ensure that even if a hacker wanted medical information, they wouldn't be able to glean anything useful from what is transmitted by this system.

Overall, the whole point of this project is to be beneficial to the health of our users. Designing in a safe and ethical way, and following the codes and guidelines enumerated above, is absolutely key to ensuring that our users benefit from this system.

5.4 Future Work

The most important addition for this system would be to incorporate the Bluetooth modules as initially designed. This allows the components of the system to be split into a scale and a hub, thereby separating the load to each of the modules. This could also result in the user operating several scales with a single hub, which would be advantageous for someone with multiple medications. The reason why this is the most important addition for this project is that it more closely aligns with the ethos of the original proposed solution: Offering a lower cost option to pill delivery systems. Splitting up the modules in this way means that the hub can be expensive and complex, with the scales being mostly mechanical and less power-hungry. Just adding the Bluetooth modules, and packaging the system neatly, would greatly increase the usability and overall worth of the design.

On the software level, there are also plenty of improvements to be made. Finding a dataset of pill weights and creating a searchable database would greatly help the users. This, in concert with a mode to measure the weight of one pill using the scale would take any requirement of the user or their doctor to enter in that data themselves. There is also the possible increased functionality of the application at large, adding more customization to schedules and reminders for both the users and their doctors. Finally, another future development to consider is the integration of our system with smart devices like the Amazon Echo. These devices are much more advanced than our hub design, and therefore would increase the usability of our devices by a significant margin.

In conclusion, there will always be ways to improve upon a design. While we didn't completely implement our proposed system, we added most of the required functionality and, warts and all, have a functioning design.

References

[1] Carr, Teresa. “Too Many Meds? America's Love Affair With Prescription Medication.” *Consumer Reports*, www.consumerreports.org/prescription-drugs/too-many-meds-americas-love-affair-with-prescription-medication/.

[2] “Medication Safety Program.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 7 June 2018, www.cdc.gov/medicationsafety/program_focus_activities.html.

[3] “What Over-the-Counter Medications Can You Overdose From?” *Desert Hope*, deserthopetreatment.com/drug-overdose/over-the-counter/.

[4] *NHS Choices*, NHS, www.nhs.uk/common-health-questions/medicines/what-should-i-do-if-i-miss-a-dose-of-antibiotics/.

[5] “Automatic Pill Dispensers.” *Automatic Pill Dispensers - Never Forget Medications Again*, www.epill.com/dispenser.html.

[6] “Xanax Pill Images - What Does Xanax Look like?” *Drugs.com*, Drugs.com, www.drugs.com/xanax-images.html.

[7] *Micro Load Cell (0-100g) - CZL639HD*, www.eztronics.nl/webshop2/catalog/index.php?route=product%2Fproduct&product_id=423.

[8] “Load Cell Primer.” *Spatial Phidget - MOT1101_0 at Phidgets*, www.phidgets.com/docs/Load_Cell_Primer.

[9] Bogde. “Calculating the Proper SCALE Value · Issue #20 · Bogde/HX711.” *GitHub*, github.com/bogde/HX711/issues/20.

[10] “24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales.” *Avia Semiconductor,* PDF File, *r*https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf

[11] “Electrical Safety in the Laboratory.” *Bases Hydroxides - Safety Library | Division of Research Safety - Illinois*, www.drs.illinois.edu/SafetyLibrary/ElectricalSafetyInTheLaboratory.

[12] ”Laboratory Safety Guide University of Illinois at Urbana-Champaign.” *Division of Research Safety,* https://www.drs.illinois.edu/site-documents/LaboratorySafetyGuide.pdf

[13] “IEEE Code of Ethics.” *IEEE - Advancing Technology for Humanity*, www.ieee.org/about/corporate/governance/p7-8.html.

[14] “The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.” *Association for Computing Machinery*, www.acm.org/code-of-ethics.

[15] HHS Office of the Secretary, Office for Civil Rights, and OCR. “Privacy.” *HHS.gov*, US Department of Health and Human Services, 16 Apr. 2015, www.hhs.gov/hipaa/for-professionals/privacy/index.html.