Pill Pal: A Medication Tracker and Dispenser

By

Deonna Flanagan (dflanag2)

Jerry Chang (jerryc2)

Pallavi Narayanan (pallavi3)

Final Report for ECE 445, Senior Design, Spring 2021

TA: William Zhang

05 May 2021

Project No. 42

Abstract

Our project provides a solution for the growing elderly population and the shortage of available caretakers by automating the task of dispensing pills. Our pill dispenser is controlled by a caretaker through a Web server connection and is capable of dispensing up to two different types of pills given certain timings per day of the week. Catering to the elderly, the control is abstracted away from patients and placed in control of the caretaker. The dispenser also includes alerts for the patient and added safety precautions such that the unit remains locked even when removed from power. This paper explores the design process and choices made to complete our pill dispenser, Pill Pal.

Contents

1 Introduction

To achieve the goal of automated pill dispensing, we needed to design a pill dispenser that is reliable, precise, and easy to configure. Designed correctly, an effective pill dispenser should be able to dispense pills quickly, on schedule, and with correct dosages that alert users. Pill Pal aims to serve as a supplement to existing caregiver-patient relationships, such that the act of administering medication may be more streamlined and less mundane. After many individual trials and tests, our modules were integrated to be delivered as a product in whole. Pill Pal has been designed to be not only a proof of concept for mechanical and electrical construction, but also as a successful execution of theory to product.

1.1 Motivation

Modern medicine allows us to alleviate the ailments that come with advanced age and has consequently allowed us to extend our life expectancy well beyond limits seen decades ago. As expected, reaping the benefits of old age requires us to maintain regularly scheduled medication intake. Older generations have an especially difficult experience keeping track of pills taken and they may have many different pills to take at different times of the day. With growing numbers of an elderly population in the United States[2], it is anticipated that there will be more elders than caregivers in the coming years. Thus, it may be seen that an overall control of specific medication taken at a time and the remembrance of taking such batches of medication will be a challenge for the elderly. Complications with medication, whether it be overdosing, under-dosing, or mismanagement, are potentially dangerous.

1.2 Background

Modern medicine helps us alleviate ailments that come with old age and has consequently allowed us to extend our life expectancy beyond that of generations before. However, while various medications could be helpful for older generations, they can experience difficulties keeping track of the times of dosages for these medications. With a growing elderly population in the United States known as the impending "Silver Tsunami", it is anticipated that there will be more elders than caregivers in the coming years. Thus, it may be seen that addressing such an issue of a patient to caretaker imbalance is imperative for keeping our elderly population healthy and well. Complications with medication and any mismanagement is potentially dangerous and a product that works under such parameters must be designed with control in mind.

1.3 High-Level Requirements

- Caregiver inputs the prescription specification of the medication into the mobile application. This information includes the number of pills to take and at what time of day. The dispenser should run with a scheduler and should dispense dose within a minute (60 sec) of inputted time. The dosage must be outputted based on application specifications 99% of the time.
- Track whether the pills have been taken through application interface of caregiver. User's profile should show whether or not the medication was retrieved and receive an update on status, within 3 minutes, about retrieval from the dispenser. This is maintained through a weight sensor capable of $0g - 100g$ precision needed for mg weight of pills. Additionally, warning about pills not taken will show up on the application 5 minutes after dispensing.
- Configure dispenser remotely from mobile application by allowing caregiver to input the type of medica-

tion put into the dispenser, the number of pills to take, and at what time the pills should be dispensed. There will also be an update when the pills must be refilled on the application so the caregiver can unlock the dispenser on the app and then refill the medication.

1.4 Block Diagram

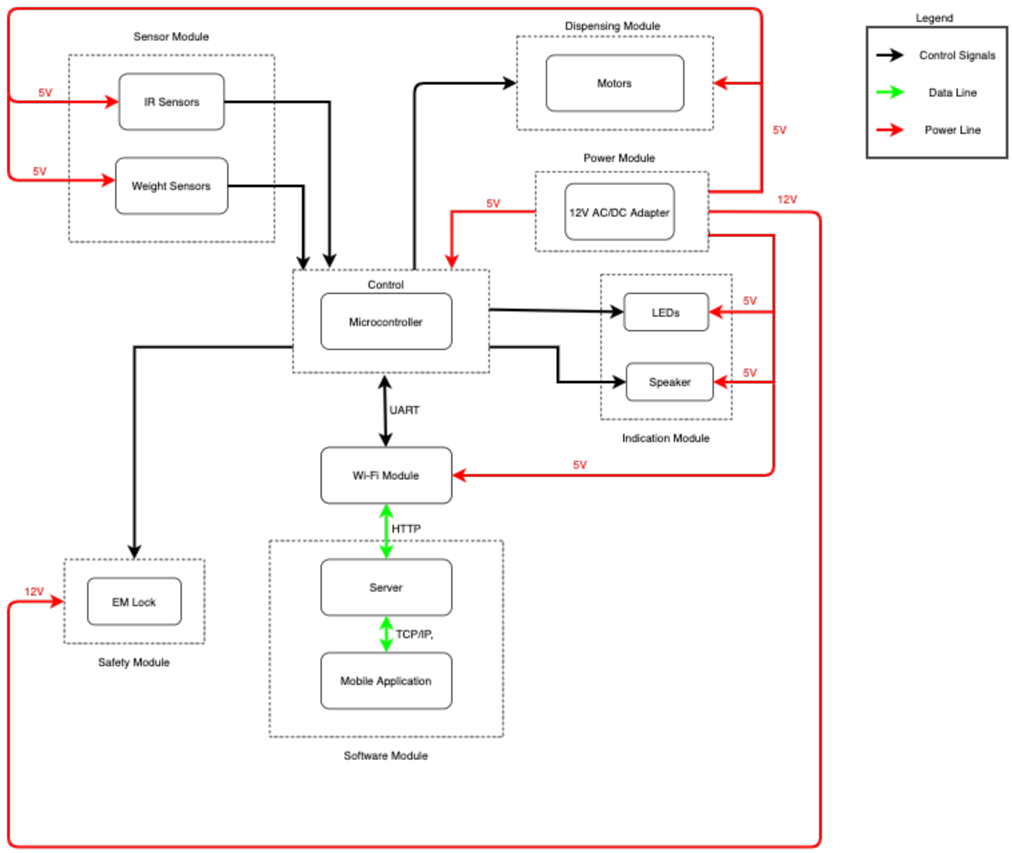

Figure 1: Overall Block Diagram

1.5 Block Descriptions

1.5.1 Control

The control for our project is implemented on an Atmega328-PU chip which we bootload with the ArduinoISP for easy programming. The microcontroller, as we can see in Figure 1, takes in inputs from the Wi-Fi Module which signals it to dispense pills or unlock the lock for refilling the pills. The control for properly dispensing pills by reading in input from the IR sensor and weight sensor as well as outputting to the motors, LED/buzzer, and lock are all implemented here as well.

1.5.2 Sensor Module

The sensor module consists of IR sensors and emitters or accurately counting the number of pills dispensed. This module also consists of a loadcell and loadcell amplifier for weighing the pills after they have been dispensed so the caretaker will know whether or not the pills have been taken.

1.5.3 Dispensing Module

The dispensing module consists of the motors which drive the disks to dispense pills and motor driver boards. The motor driver boards take input from the microcontroller and outputs to the motor. This ensures that the motors are only turned on when the program specifies they should turn. The microcontroller also specifies the motors' speeds and the number of steps to take.

1.5.4 Power Module

The power module provides the power necessary to run the pill dispenser. We first use an AC to DC converter to take power from a wall outlet to 12V DC. We then step down the 12V to 5V, then once more from 5V to 3.3V. This provides us with the 12V, 5V and 3.3V needed for different components.

1.5.5 Indication Module

The indication module consists of a buzzer and an LED which go off three times once the pills have been dispensed, and once more for a longer duration after the pills have been removed.

1.5.6 Safety Module

This module consists of a 12V solenoid lock which can be unlocked using the app for adding in more pills, but which remains locked even when removed from power. This component was added as an additional safety factor.

1.5.7 Software Module

The software module consists of a web server that is the interface for the caregiver to input pill specifications and dispensing times. The web server communicates with the ESP8266-01S WiFi IC to decide when to dispense pills. The WiFi IC then sends information to the microcontroller through UART serial communication. The WiFi IC receives updates about the pill dispensing from the microcontroller to be able to update the web server. This not only allows for the caregiver to input specifications for pill dispensing, but also to receive a notification if the medication has not been taken after a specified period of time.

2 Design

In designing a pill dispenser, the focus on a complete and reliable dispensing module was firmly centered in our process. From the very inception of the project, there has been a strong emphasis on how our pill dispenser will be able to not only dispense the correct dosage for a mix of pills, but also be controlled tightly by logic. Past iterations of the project in the course have attempted different mechanical designs by the machine shop such as a gumball drop mechanism; however, past attempts have been unsuccessful in controlling pill dispensing for different sized and shaped pills. This mechanism also is not ideal in that the pills are likely to get jammed as gravity pulls them down and they are funneled to be dispensed one by one. To create an effective product, we looked to alternative solutions of our own.

Referring to Figure 2, our final design of the dispensing module may be seen. Based off of automatic pill counters seen on the market, this dispenser module utilizes a revolving plate to move pills to the outside walls of the plate and separate the pills into a singular stream after passing through a channel. The pills are then detected by infrared sensors to keep track of the number of pills that have dropped down. This count is used to decide when to stop motor revolution. The end product uses a plate for each pill variety, to prevent cross contamination and allow for better dispensing.

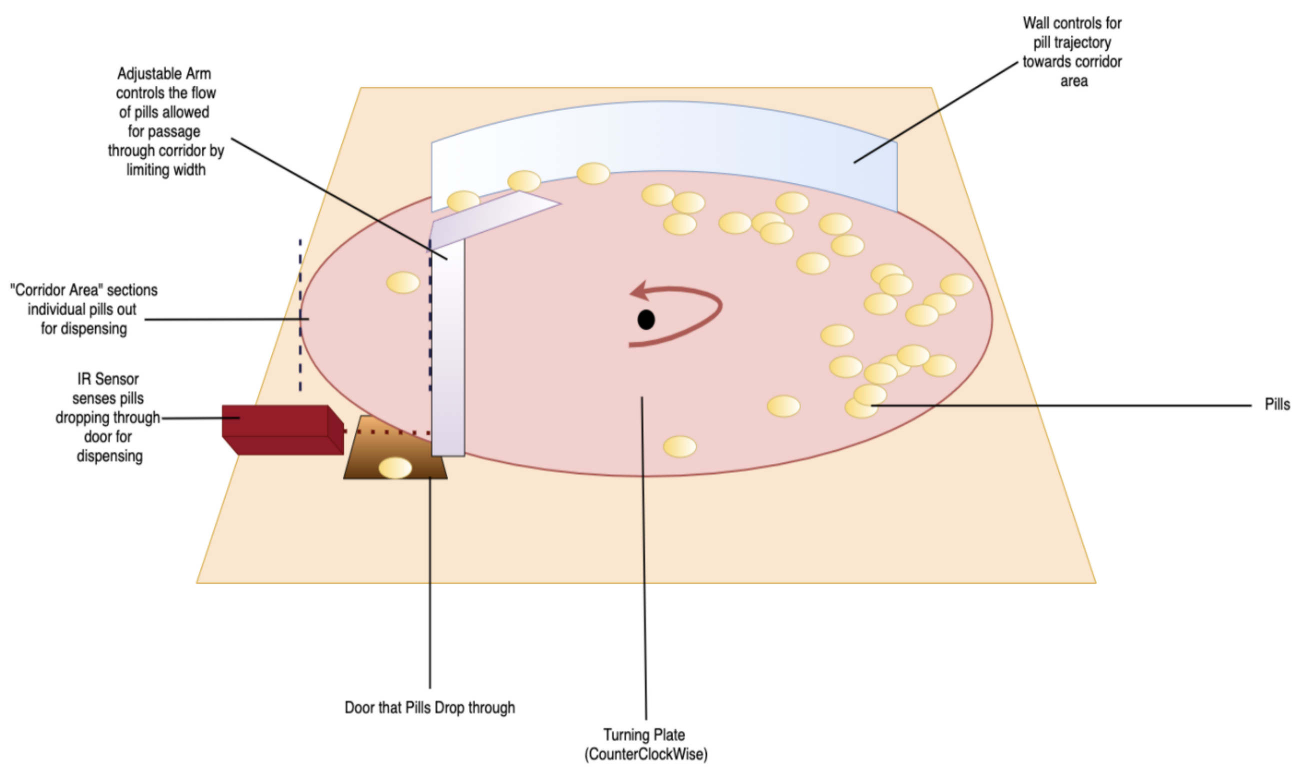

Figure 2: Mechanical Design of Dispensing Module

Figure 3: Physical Prototype of Pill Dispenser

In Figure 3, on the left, we can see how on the physical prototype consists of two dispensing modules, where the disks spin in opposite directions to funnel the pills inward toward the funnel to be dispensed in the cup below. The image on the right of Figure 3 displays the back of our dispenser where we have a lock for safety reasons. The lock can be unlocked by the caretaker through the web server connection for refilling pills and will remain locked even when disconnected from power.
With a finalized mechanical design, we then added functionality to control the dispensing process and its automation. This includes software and hardware design that aims for a more feature rich pill dispenser. All component are integrated for a Pill Pal that addresses ease of use, safety, and timeliness.

2.1 Power Module

The power module consists of a 12V AC to DC converter (HM-01831), a 12V to 5V linear voltage regulator (LM7805S/TR), and a 5V to 3V linear voltage regulator (ADP123AUJZ-R7) linear voltage regulator. For 12V, GND, 5V and 3.3V we have created various breakout sockets for them so we can use them for various components. The schematic of the power circuit is displayed in Figure 4.

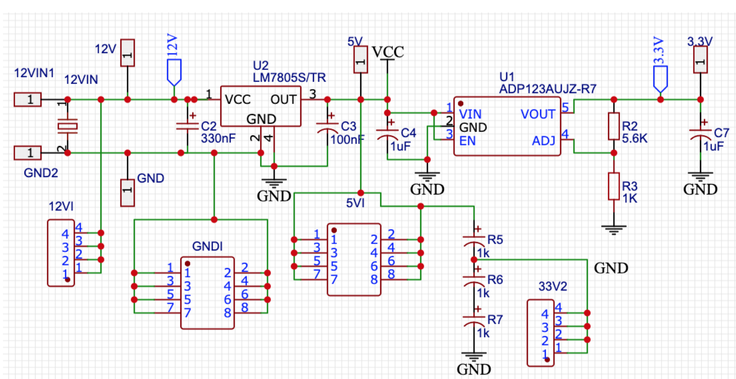

Figure 4: Voltage Regulation Circuit

After obtaining power from the wall socket, we convert the AC power to a steady 12V DC to use in our circuit. The 12V is used for powering the solenoid lock as well as the two stepper motors. We then step-down the 12V to 5V. The 5V acts as our primary voltage supply as it powers our Atmega328-PU microcontroller, as well as various peripherals. The IR emitters/receivers, the relay for the solenoid lock, the load cell, and piezo buzzer and LED are all controlled with 5V inputs. We then again step down the voltage once more because we have a low powered WiFi IC to drive which requires 3.3V. Because the WiFi IC can be damaged if the voltage varies too much from 3.3V, we obtained two sources of 3.3V. The first way we get 3.3V is feeding the 5V into a linear voltage regulator, while the other way just consists of stepping down the voltage using a few resisters to create a voltage divider circuit.

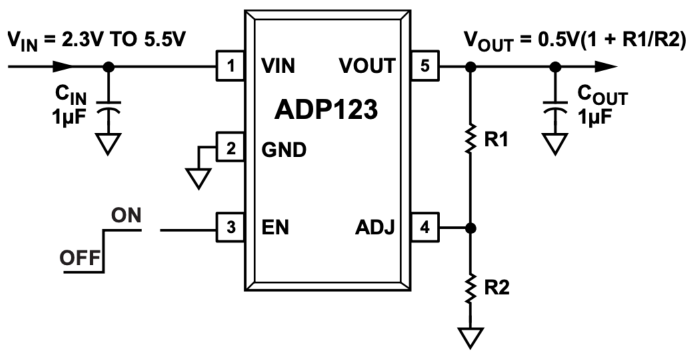

Figure 5: ADP123 with Adjustable Voltage

$$V_{out} = 0.5V(1 + \frac{R1}{R2}) \tag{1}$$

For converting the 5V into 3.3V with a linear regulator, we used ADP123 with adjustable voltage as seen in Figure 5. Using Equation 1, We selected values for R1 and R2 so that we get a V_{out} of 3.3V when V_{in} is 5V. We decided on a value of 5.6KΩ for R1 and 1KΩ for R2.

2.2 Dispensing Module

Our dispensing module consists of the two NEMA-17 stepper motors which turn the disks to dispense, and their L298N motor driver boards. We selected stepper motors over cheaper DC motors as they allow for precise speed control and precise positioning. The motor we selected has 200 steps in each 360 degree rotation and we can easily run the motors in the opposite direction by adding in a minus sign in front of the steps. This allows us to turn the motor as little as one step (1.8°) at a time. The stepper motors we chose are also better suited for higher torque which was advantageous as the disks attached to them ended up being heavier than we anticipated. Additionally, we found that the stepper motors we chose are used in a wide variety of projects and can be easily controlled with simple code in the Arduino Development Environment that we used[7]. We originally planned to drive the motors with separate L298N motor driver chips, but through testing we found that these chips required additional external capacitors and diodes to work properly. Rather than adding all these additional components, we opted for an integrated circuit board that has the L298N motor drivers included with them. These boards were also more reliable. Since our microcontroller doesn't have the power to drive the motors directly, we use the motor driver board which takes in 12V, ground, and four inputs from the microcontroller. We use a total of 8 outputs from the microcontroller, 4 for each motor, to control the movements of each of the motors which drive the disks for dispensing.

2.3 Sensor Module

The sensor module is important in ensuring that the correct number of pills are dispensed and in checking when the pills have been taken from the dispenser. In order to keep a count of the number of pills dispensed, we decided to use an IR sensor and emitter pair. We placed the sensor and emitter such that they faced each other to create a small beam near the location where the pills would drop from the wooden disc into the funnel. This was so that when the pill got close to dropping, it would break the IR beam and the sensor would send an output of 0.0V to the microcontroller. Thus, we can keep track of the number of pills dropped based on the output of 0.0V from the IR sensor. The load cell and load cell amplifier sat below the cup that collected the pills. We effectively used the load cell to detect a difference in weight. Once the pills were dispensed, we would take a reading and then compare it every 500ms to see when the weight changes.

2.3.1 IR Breakbeam Sensor and Emitter

In order to choose the IR sensor and emitter that would work best for our project, we had to test a few options. We found that the Adafruit 2167 breakbeam IR sensor worked best. We also found that we needed to place the IR sensor and emitters such that there was 25cm or less between the two bulbs. We ended up using a distance of about 2cm which worked well with common pill sizes. The output from the IR sensor was connected to digital pins 10 and 11 on the ATmega328-PU. This was because the output from these IR sensors is digital. The input to both the IR sensor and IR emitter was 5V and ground that we got from the power module.

2.3.2 Load Cell and Wheatstone Bridge

The Load Cell in important as it senses whether or not the pills have been taken by the patient after being dispensed. In Figure 6 we show how the load cell is placed where the pills are dispensed. A removable cup sits in that compartment to catch the pills as they are being dispensed.

Figure 6: Load Cell

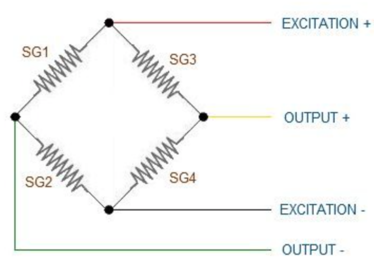

Figure 7: Load Cell Wiring

Initially we only purchasd the CZL639HD load cell, but we found that the weight change was not large enough, so we couldn't reliably read the weight without an amplifier. To solve this problem, we ordered an HX711 load cell amplifier. This made a significant improvement in the output. Although we were not able to get precision where we could get the exact weight of each pill, we were able to recognize the change of weight accurately enough to represent when the pills have been taken and when they have not been taken. The inputs to the Wheatstone bridge were the four outputs from the load cell. The bridge gets 5V and ground from the linear voltage regulator and then has two outputs to the ATmega328-PU. The weight is taken as a difference of the two outputs that are sent to the microcontroller. Figure 7 displays how the load cell is wired. Excitation+ and Output+ are used to find the change in weight where Excitation+ should be greater than Output+ when there is a weight applied to the load cell. Excitation- and Output- are just connected to a common ground to ensure that the weight readings show an accurate difference in voltage.

2.4 Control Module

For our microcontroller, we chose to use an ATmega328-PU. This was because it gave us the ability to get both inputs and give outputs to a lot of components. Additionally, we were able to find many resources on bootloading the ATmega328 with Arduino IDE to allow for quick code uploading and testing. We needed the microcontroller to have RX and TX pins to allow us to upload code and to connect to the ESP8266-01S WiFi IC.

2.5 WiFi and Software Modules

As we wanted the caregiver to be able to remotely check the status of the pills, input pill configurations, and know when to refill the pills in the box, we decided that we wanted this to be done through a WiFi connection. We chose the ESP8266-01S to connect to WiFi and communicate to a web server. We used web server to take input about the type of pill, number of pills to dispense on each day of the week, and at what time to dispense the pill (in 24 hour time). We updated the web server with the count of pills in the box based on how many are dispensed. We also updated the status of the pill in terms of taken, not taken, or an error with dispensing. Finally, we allowed the caregiver to be able to unlock the box through the use of buttons on the web server. We utilized UDP protocols to implement sending information only when the current hour and minute matched the pill dispensal time on that day. In order to send the information, we stored the information in a file on the ESP8266-01S on chip flash file system (called SPIFFS). We used a char (8-bits) where the first or most significant bit was 0 if the lock should stay locked, 1 if it should be unlocked. The next three bits were the number of pills to dispense from disc 1 in the box, the following three bits were the number of pills to dispense from disc 2 in the box, and the last bit (least significant bit) did not have any meaningful value. This information in the file was sent through UART to the ATmega328-PU if the time conditions were met. The TX pin on the ESP8266-01S was how we sent the information to the microcontroller. The RX pin was used to receive information from the microcontroller. The information received by the WiFi IC was a char where the most significant four bits were changed based on whether the pill was taken, not taken, or there was an error in pill dispensing. A benefit to the ATmega328 was that we could program which pins would serve as RX and TX to the ESP8266-01S. This way, we could still program the microcontroller while keeping the UART serial connection between the WiFi IC and microcontroller. The microcontroller would receive information about how many pills to dispense for each disc and whether the lock should stay locked or be unlocked. Then it was able decode the char and start the dispensing process.

2.6 Safety Module

Given the issue that pills could be dangerous if the user overdoses, we wanted to implement a safety feature with our pill dispenser. This way, only the caregiver can access the pills that are not dispensed at the correct time and can access the inside of the box if there are any issues. When thinking about the dangers of overdose, we decided that the box should naturally remain locked even when removed from power. This was the reason we chose the 1512 EM solenoid lock or a "fail-secure" lock. Thus it would naturally remain locked when pulled from power. The only time the lock retracts is when 9 - 12V are applied. We were able to configure the lock to open or close by getting output from the microcontroller as input to our 5V relay and powering our relay with 5V from the linear voltage regulator. We also connected 12V as input to the relay to allow it to output to the solenoid lock when the output signal from the microcontroller was high. The positive output of the lock was connected to the NO (normally open) pin on the relay to ensure the lock only opened when 12V was applied. The solenoid lock was also connected to the common ground.

2.7 Indication Module

After the pills have been dispensed we need to notify the user to take their pills. While there is a pill status update on the web server, we wanted a notification on the physical dispenser for the patient. We came up with having both a piezo buzzer and green LED alert to allow those who have poor sight or are hard of hearing to still receive some sort of notification. The buzzer and LED went off the times after the pills have been dispensed and then once again after the pills were retrieved by the user. In Figure 8 we see how the buzzer on the left and LED on the right are situated right by where the pills are dispensed in the cup at the front of our pill dispenser. The input from the LED and buzzer come from a digital output pin from the ATmega328-PU. The anode pins of both buzzer and LED are connected to ground. We also used a 330Ω resistor from the output of the microcontroller to the cathode pin of the LED, to control the current going to the LED.

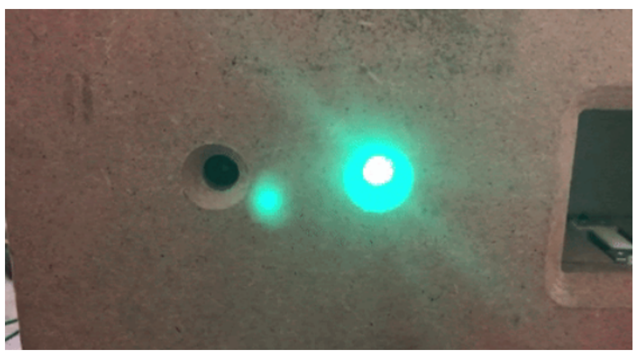

Figure 8: Indication Module

3 Design Verification

For all components of our pill dispenser, individual tests must be performed to determine the behavior and anticipate the interaction a piece of hardware may have with other components in the project. This allows us to make design adjustments, optimizing for a more complete end product.

3.1 Power Module

To test the power module fulfilled our requirements, we first plugged in the 12V AC/DC converter into a wall outlet and used a multimeter to test the voltage drop over the positive and negative leads. We got a reading of 12.08V which was within our requirement range listed in the table below. We also tested the current using the corresponding mode and got a reading of 2.5A. We later found this was too high of a current for our components. This is why our requirement for current is not viable. We then soldered our LM7805S/TR and ADP123AUJZ-R7 linear voltage regulators onto our PCB along with accompanying resistors and capacitors. We then put the positive wire to the 12V through-hole on the PCB and the negative wire to the GND through-hole to test the linear voltage regulators. We used the multimeter to test the supposedly 5V output from the LM7805S/TR voltage regulator and got 5.05V as the output which was within the range we required. Finally we tested what was meant to be 3.3V output from ADP123AUJZ-R7 voltage regulator and the resistors. We were able to read an output of 3.34V which was withing the required range we specified. The values we found are displayed in Figure 9.

Required	Actual Value
11.5V $\pm 0.7V$	12.08V
5.0V $\pm 0.5V$	5.05V
3.3V $\pm 0.6V$	3.34V

Figure 9: Table of Required and Measured Voltage Values

3.2 Microcontroller

While testing the ATmega328-PU with the motors and IR sensors, we found that the IR sensors output as digital LOW or HIGH and therefore would need to be connected to digital output pins. However, we were still able to test that at least four analog pins could work as output because we used four analog pins to output to the motor. We connected 5V to the ATmega328-PU and were able to upload simple example code to make an LED blink, thereby testing that the microcontroller was correctly bootloaded and could get new code and run it. In order to test the input we were getting from the IR sensors, we connected their output pin to digital input pin 10 and 11. When the IR beam was not blocked, the code we had would print to serial saying the beam was "Unbroken". Once the beam was broken, we had the code print "Broken" to the serial. This tested that we were able to successfully receive a HIGH or 1 output when no pill would pass through the beam, and a LOW or 0 output if a pill passed through the beam.

3.3 Motors

In testing the motors, the most important requirements to experiment with were the speed and rotations of the motors. Testing with the NEMA-17 step motors start with verifying the operating voltage. Then, we

place a disc on the motor and scrub step increments with loads on the disc to determine the smoothness of rotations with different loads on the disc.
As seen in Figure 10, the percentage of successful trials peak at four steps of the stepper motor, and decrease with less or more steps. Successful trials are defined as a dose of the correct amount of pills dispensed for both motors, one after the other together as a single trial. At four steps, the accuracy for 15 trials of two pill types was evaluated at 85%.

Figure 10: Chart of Successful Pill Dispensing by Motor Steps

3.4 IR sensors

To test the IR sensors, we used a multimeter to check the output from the output wire. We supplied 5V and ground to both the IR LED and the sensor and had them lined up to face each other. Then we tested the voltage drop over the power and ground wires on both the LED and sensor to find out if it is receiving 5V. We were able to confirm that the LED was receiving 5V and was therefore on. Also the IR sensor was receiving 5V. We tested the output voltage from the IR sensor and found it to be 0.7V when there was nothing blocking the beam between the IR LED and sensor. When we placed an object of about 10mm to block the beam, we tested the output voltage from the sensor to be 0V. This was quite accurate as we moved the object a few times away from the beam and then crossing the beam. Each time, the output was 0V when the object was blocking the beam. In Figure 11, we have shown the high and low voltage output for each IR sensor we tested.

IR Sensor	Low (Volts)	High (Volts)
TSL237S-LF	-	No output in DC and AC
TSOP34438	-	0.7V
SEN-0042	3.4V	3.7V
Adafruit 3mm Break Beam	0.0V (output as digital 0)	0.7V (output as digital 1)

Figure 11: Table of High and Low Voltage Outputs from IR Sensors

We then tested different sized pills starting with one that was 6mm till those that were 16mm. The smaller pills were harder to detect with the sensor as they had to be lined up exactly between the LED and sensor bulbs. However, we were able to successfully detect the 6mm, 9mm, 13mm, and 16mm pills when they were placed to break the beam. The output of the IR sensor gave 0V reliably when the different sized pills broke the IR beam. This proved that we could break the beam with a range of different sized pills although the 6mm and 9mm pills were very difficult to place.

To test the distance range which the IR sensor and LED are still operable, we started with them being 1cm (10mm apart) and were able to successfully test when the beam was broken with a pill 16mm long. The output was 0V. We then moved the LED 10mm away at a time and retested that the 16mm pill was able to break the beam successfully each time. We tested 10mm, 20mm, 30mm, 40mm, 50mm, 60mm, 70mm, 80mm, 90mm, and 100mm. Each time, the pill was successful in breaking the beam and the output from the sensor would read 0V on the multimeter. We then incorporated the IR sensors and LEDs with the microcontroller and the motors in the box. We tested detection of a red 13mm long pill and a blue 16mm long pill. With the trials completed, we noticed that the larger blue pill was generally detected more accurately. This is shown in Figure 12, where the blue pills tend to overlap and have a higher value than the red pills.

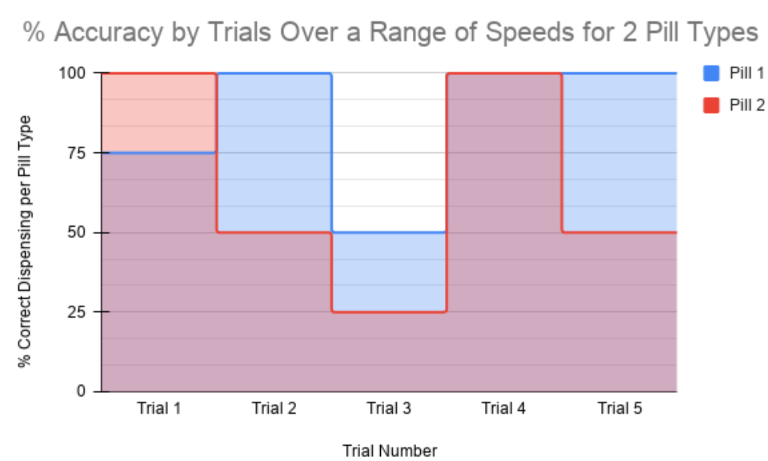

Figure 12: Chart of Accuracy for IR Pill Detection by Different Pills

The output is based on what the microcontroller received as input from the output line of the IR sensor. As the IR sensor outputs a digital 1 or 0, we were able to read it as either the beam was broken or it was not broken (0 being broken, 1 being not broken). With all the testing, we were able to successfully complete our verification and fulfill the requirements we set in the design document.

3.5 Load Cell and Wheatstone Bridge

The load cell is used to detect the presence of pills in the pill retrieval area, or its lack thereof. For a sensitive enough load cell, a CZL639HD is used, with a granularity of 0-100g. This granularity is important, since most pills' weights lie in a range from 200mg to 500mg. This is coupled with a HX711 wheatstone bridge to amplify and fine-tune voltage readings from the load cell. Tests for the load cell involve verifying the operating voltage of the load cell with a multimeter, then adding pills of varying sizes one by one to read the changes in weight. This is done until the readings are different enough to have a perceptible change from its tared weight. The cup is then removed from the top of the load cell and the change may be mateched against a threshold value and verified by printing raw voltage readings to hardware serial.

3.6 EM Lock

In order for the lock to be "fail-secure" we had to ensure that it remains locked when not connected to power. We tested this after the lock was mounted in the box by trying to open the top of the box without connecting the PCB to power. We could not open the plexiglass lid which shows that when 0V applied, the lock stays in the locked position. Once we connected the EM solenoid lock to the 5V relay, we tested when IN1, the output from the microcontroller, was set to low and when it was set to high. When IN1 was set to low, the lock remained in the locked position, which is what we required. When IN1 was set to high, the lock retracted and was in the unlocked position. This means that once we applied 12V, the solenoid lock was able to unlock, as we required.

3.7 Software and WiFi Module

3.7.1 WiFi IC

The WiFi IC chosen needs to support WiFi connectivity to receive and send HTTP requests to some server for data interaction. With these needs in mind, the chosen WiFi IC is an ESP8266-01.
The ESP8266-01 WiFi connection to the microcontroller was tested for by sending packets of bytes between the two components. The packets are read and printed to hardware serial as ASCII characters to verify successful transmission. The time is kept from the beginning of the transmission to the end and verified that the data transmission is less than 10% of the 5-minute timeout threshold of 30 seconds.

3.7.2 Server

The ESP8266-01 also serves as the host server in our project. The server establishes a connection between the on-board host with some client, a laptop or mobile device, to send and receive data. Tests performed to verify the data transmission from microcontroller to server were done by first creating a web server, then testing with a simple circuit to turn on a LED. Controlling the on and off state of the LED proved that the data transmission for the server is complete.

3.7.3 Application Interface

The application interface is the web server itself. Testing the interface consists of inputting in user data into the web server and first displaying it back on to the web server. This test confirms that the web server is reading and writing user input. Then, a schedule is used to send the user input to the microcontroller. The data sent is printed using the serial output of the microcontroller and the ASCII value is verified. This proves that the interface is able to use the web server to communicate with the microcontroller using user input.

4 Cost

To make our product more accessible than other pill dispensers on the market, we aimed to keep our costs low. Table 1 outlines the cost of materials while Table 2 outlines the cost of Labor. Because of time constraints, we did pay a bit extra for certain parts to get them delivered more quickly. The overall cost of this project was $\$125.25 + \$52,500 = \$52,625.25$.

4.1 Parts

Table 1: Parts Costs

Part	Manufacturer	Quantity	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
EM Locking Mechanism	Adafruit Industries LLC	1	14.95	14.95	14.95
Load Cell (0-100g)	Phidgets Inc.	1	7.00	7.00	7.00
Stepper Motor (Nema 17)	Twotrees	2	35.99	35.99	35.99
L298N Motor Driver Board	Qunqi	2	8.89	8.89	8.89
WiFi IC (ESP8266)	MakerFocus	4	12.99	12.99	12.99
HX711 Load Cell Amplifier	Sparkfun	1	9.95	8.46	9.95
5V Relay	Youngneer	1	1.50	11.99	1.50
IR Breakbeam Sensor	Adafruit	2	13.36	13.36	13.36
HM-01831 12V DC Converter	Binzet	1	15.49	15.49	15.49
LM7805S Linear Regulator	HGSEMI	1	0.23	0.16	0.23
ADP123 Linear Regulator	Analog Devices	1	1.14	0.79	1.14
ESP8266-01S	MakerFocus	1	3.25	3.25	3.25
Buzzer	INGHAi	1	0.66	0.51	0.51
Total					**125.25**

4.2 Labor

Table 2: Labor Costs

Name	Hourly Rate ($/hour)	Hours	Total ($) = $HourlyRate * Hours * 2.5$
Pallavi Narayanan	35	200	17,500
Jerry Chang	35	200	17,500
Deonna Flanagan	35	200	17,500
Total			**52,500**

5 Conclusion

In the past few weeks, we have had the exceptional privilege of designing and creating a deliverable project from both a technical and professional point of view. Throughout every step of the process of mockups, ordering, soldering, testing, coding, etc. we have been intimately involved with choices that have brought an idea to fruition. Although not all components were completely integrated, we clearly see the next steps that would have to be taken for a polished project. For the work and goals that were met, we have had the challenges of performing tests and drawing our own conclusions to navigate through the design process. With more time, we firmly believe that Pill Pal may be presented as more than a proof of concept, but as a final product, ready for the market.

5.1 Accomplishments

We have successfully tested all individual components of the pill dispenser and have integrated most of the parts by the end of the construction period. The actual act of dispensing pills is around 75% successful in accuracy of dosage. This figure includes the integrated system of motors, IR sensors, load cell, LED, and buzzer tested with a variety of pill types. Given that this is a first prototype with limited time and budget, we believe that this successful integration of these hardware parts with control logic is a victory in itself.
Besides the accuracy rate, we have also previously accomplished communication between the WiFi IC and Microcontroller to send data in between two points. The Web Server also successfully polls for precise time in a geographic region for schedule checks. The Server also stores and reads user input to make logical decisions based on input.
These component interact together to produce the aforementioned accomplishments, all while soldered on a PCB we designed.

5.2 Uncertainties

Uncertainties in our project may be seen in the control of pill dispensing. With an accuracy rate of 75%, this leaves another 25% of error to dissect. These errors are not from a lack of data transmission, but due to mechanical and sensor limitations that are unavoidable. The sensors leave a bit to be desired with its precision in readings of pills. Additionally, the location of the sensors were decided by physical limitations of the design and we could not place them such that the bulbs of the LED and sensor were low enough for shorter pills to be detected. This meant that we had a limited size of pill that could be detected thus we went with taller pills. The motors are not always consistent in its rotations and steps may be somewhat skewed from residual noise in previous rotations, accounting for odd twitching behavior seen at times.
Other uncertainties exist with the UART serial communication between the ESP8266-01 and the ATMega328-PU. The ESP8266-01 is designed to write to hardware serial when reconnected to power. This proves to be an issue when the pill dispenser is disconnected then reconnected to power, as it then sends startup noise to the Microcontroller. The measurable noise each time is inconsistent, making it difficult to pinpoint at times. However, it is observed that further examinations of the output may be done to filter out the startup noise.

5.3 Ethical considerations

There are a couple of potential safety hazards with our project. This is mainly regarding the control of a microcontroller, sensors, motors, EM Lock, and a WiFi module with medicine. This is because there could be the issue that the microcontroller stops working or sends wrong information about the amount of pills to dispense. If it is too many pills, an overdose can cause fatal or potentially lethal side effects such as internal bleeding, bruising, poisoning, etc. If the sensors and modules overheat, this could negatively affect the medication as pills become ineffective over $86°F$ [4].

Pill storage based on FDA regulations requires the pills to be stored at adequate lighting, ventilation, temperature, sanitation, humidity, space, equipment, and security conditions [6]. The recommended temperature for most pills is $59°F$ to $86°F$. [7] The regulated relative humidity level for most pills is 60% or lower with good ventilation in the storage location and an odor-free environment. [8]

Another large safety concern is if the outlet adapter does not work properly, it could allow too much amperage and cause a fire or short circuit the system. To safeguard against this, we will thoroughly test the adapter to ensure it works as expected and double check the connections of the subsystems and modules to ensure they are not receiving more current than they should be. This will lessen the likelihood of a fire. To mitigate the possibility of a short circuit, we will have to double check how we connect power and ground to all of our sensors and modules.

We develop this project in firm belief of IEEE Code of Ethics 1.1 "to hold paramount the safety, health, and welfare of the public"[1] at the heart of our project. This solution proposes to accept the growing elderly population and help assuage the inevitable need for more precise caregiver control. Thus, we also consider safety lock features in case of potential abuse of our product by addicts.

We also seek "to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies"[1] by giving control of the drug distribution to medical caregivers instead of engineers. This technology is being developed for health fields in hopes of improving coverage with accessible innovation. Should there be mechanical failures in the control of pill drops, we will have an alert for the caretaker and the physical alert system, the LED and Buzzer, should not signify to the user that pills should be taken. This will control for possible incongruities in dosage, for the caretaker's own discretion.

Furthermore, we earnestly acknowledge the use of prior developed technologies in our modeling of the dispensing system as 1.5 states that we must "be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others."[1] The aforementioned mechanical complexity we model and alter our dispensing system after seeks to make for a safer and more accurate system for patients.

5.4 Future work

For future work considered, we are looking into three main goals.
First, with a larger budget and more time, we would like to spend more time testing other options for more sensitive sensors with greater precision. This would allow for a tighter control on dispensing such that pills

of varying sizes and shapes may also be dispensed. Raising the accuracy rate in dispensing is the greateset concern for future development, should this product be released to the market.

Secondly, we would like to develop a more robust web application to host different profiles, such that multiple pill dispenser units may be made for several patient-caretaker pairs. The individual configurations are likely to be stored on systems in the box so that issues with WiFi connectivity do not interfere with the act of pill dispensing, and will only affect minor status updates to the interface. The more dangerous alternative would be a lack of pills that could cause shock to the users. A better interface and remote mobile application could allow for subscription type services and further refined work with pill dispensers.

Finally, more pill dispensing units should be added to each pill dispenser. As it currently stands, we are operating with two different types of pills, on two different dispensing plates. Ideally, we would want to dispense more than two types of pills for diverse options for users. Given that we have successfully integrated two dispensing units in this prototype, we should be able to easily scale up by simply adding more dispensing units and forming appropriate connections in hardware and software.

References

[1] IEEE, "IEEE Code of Ethics," Online, IEEE Std., 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html

[2] U. S. C. Bureau, "By 2030, All Baby Boomers Will Be Age 65 or Older," Online, The United States Census Bureau, 2020. [Online]. Available: https://www.census.gov/library/stories/2019/12/by-2030-all-baby-boomers-will-be-age-65-or-older.html

[3] "Experience a truly easy, versatile counter: Rx Count Corporation," Online, Rx Count, 2021. [Online]. Available: https://rxcount.com/?gclid=Cj0KCQiAgomBBhDXARIsAFNyUqNoHzUp7Z8ox-7JUDuvFTt4N_WOWLr8zn4_upiGtQFmgwp7R--lq-IaAuruEALw_wcB

[4] M. Shomon, "Variations in Temperature May Be Hazardous to Your Drugs," Online, Verywell Health, 2021. [Online]. Available: https://www.verywellhealth.com/how-temperature-can-affect-medication-stability-3233264

[5] B. C. Hancock, "An investigation into the kinetic (sliding) friction of some tablets and capsules," Online, International journal of pharmaceutics, 2010. [Online]. Available: https://www.semanticscholar.org/paper/An-investigation-into-the-kinetic-(sliding)-of-some-Hancock-Mojica/4e65713911b9ef916cd8dcc5015f20e3992d6d5b

[6] Food and D. Administration, "CFR - Code of Federal Regulations Title 21." Online, International Journal of Pharmaceutics, 2020. [Online]. Available: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/cfrsearch.cfm?fr=205.50

[7] "Choosing the Right Motor for Your Project-DC vs Stepper vs Servo Motors," Online, Seedstudio, 2019. [Online]. Available: https://www.seeedstudio.com/blog/2019/04/01/choosing-the-right-motor-for-your-project-dc-vs-stepper-vs-servo-motors/#:~:text=Stepper

Appendix A Requirement and Verification Tables

Requirement	Verification	Verification Status (Y/N)
1. Must output 10.5 ± 1.5 VDC and 2.95 ± 0.05 A at an error range of 0 - 25%	1. Test voltage rating a. Use a multimeter to check voltage drop of power supply over a 1KΩ resistor to test that 10.5 ± 1.5 VDC and 2.95 ± 0.05 A is outputted using Ohm's law.	**Y***

*Not able to incorporate into PCB design, due to issues with tracks. However, the adapter worked with linear voltage regulators.

Figure 13: AC/DC Adapter Requirement and Verification Table

Requirement	Verification	Verification Status (Y/N)
1. Microcontroller should take at least two analog inputs from the sensors	1. Test connections using driver codes a. Connect 3.3V to VCC to power the board b. Upload the driver code	**Y**
2. A digital output HIGH should be produced when the sensor provides a 1 and a digital output LOW should be produced when the sensor provides a 0	2. Test digital output produced a. Set PE sensor output to 0 and check if the corresponding output pin voltage is low (approx. 0V) b. Set PE sensor output to 1 and check if the corresponding output pin voltage is high (approx. 3.3V)	**Y**
3. Must sink or source 10mA on each of the two GPIOs at 3.3V +/- 5%	3. Check that the amperage can reach 10mA and expected values should be within +/- 5% of expected value	**Y**

Figure 14: Microcontroller Requirement and Verification Table

Requirements	Verifications	Verification Status (Y/N)
1. Must operate at 5.0 ± 0.2 VDC	1. Test voltage rating a. Use a multimeter to check voltage drop over the step motor. Ensure the drop is 5.0 ± 0.2 VDC.	**Y**
2. Must rotate 360° for a minimum of 0.07 ± 0.01 sec/60°	2. Rotation Test a. Place a marking on the base of the motor and attach a piece of colored tape to the rotating piece where the starting location of the tape coincides with the marking. Starting from 3.5V to 6V, test in 0.5 volt increments to count the number of times the tape passes the marking in a minute. b. Weigh a piece of plastic or a CD such that it is 0.58 ± 0.05 oz. Attach the item to the motor and connect the motor to the power supply to see if it can still work with the additional weight.	**Y**
3. Must be able to turn 360° for a minimum of 0.07 ± 0.01 sec/60° with a load of 0.58 ± 0.05 oz attached.	3. Rotation with Load Test a. Attach a disc of maximum diameter 140mm to the motor. Place a marking on the disc and a marking exactly below on the base of the motor. Check that the disc's marking passes the marking on the base in 0.42 ± 0.06 seconds. b. Place an item on top that is 0.58 ± 0.05 oz. Do the above test while the object is on top of the disc.	**Y**

Figure 15: Motors Requirement and Verification Table

Requirement	Verification	Verification Status (Y/N)
1. Must operate at 4.1 VDC ± 1.4VDC	1. Test voltage rating a. Use a multimeter to check voltage drop over sensor when in circuit is 4.1 ± 1.4 VDC	**Y**
2. Must be able to detect a pill when it crosses a beam of maximum distance 100mm, minimum distance 12.5mm, range of 10° above or below the sensor	2. Set up the IR LED on one side and the IR sensor on the other side at 12.5mm a. Use a small object (10mm or less) and guide it through IR pair's beam b. Check the voltage change through multimeter, change should be 0.3V or greater c. Set up the IR LED on one side and the IR sensor on the other side at 100mm d. Use the same small object (10mm or less) and guide it through the IR pair's beam e. Check the voltage change through multimeter, change should be 0.3V or greater	**Y**
3. Must be able to detect different sizes of pills minimum of 6mm to maximum of 12mm	3. Use a small 6mm pill within the range of detection and use a multimeter to check output voltage from the IR sensor. Check to see if voltage changes by 0.3V or greater. f. Use a medium 9mm pill within the range of detection and and use a multimeter to check output voltage from the IR sensor. Check to see if voltage changes by 0.3V or greater. g. Use a large 12mm pill within the range of detection and use a multimeter to check output voltage from the IR sensor. Check to see if voltage changes by 0.3V or greater.	**Y**

Figure 16: IR Sensors Requirement and Verification Table

Requirement	Verification	Verification Status (Y/N)
1. Must operate at 5 ± 0.25 VDC	1. Test voltage rating a. Use a multimeter to check voltage drop over sensor when in circuit is 5.0 ± 0.25 VDC	**Y**
2. Need to be accurate in deciding when pills are in the cup and when they are not.	2. Load in cup test a. Place a 1oz plastic cup on top of the weight sensor and use a multimeter to check the voltage drop. Add 6mm, 9mm, or 12mm, or all three pills and check the voltage drop over the weight to see if it has changed from the previous measurement. Then take away pills and cup to see if the 75% threshold is crossed. If so, this is a good mark for detecting pills or a lack thereof.	**Y**
3. Must be able to differentiate between when the cup has no pills and when it does have pills (through difference of weight measurement voltages). A tolerance threshold of up to 75% a dosage weight will be allowed, such that we say pills have been retrieved once a 25% → 24% threshold change in voltage is crossed.	3. Threshold test a. Place a 1oz plastic cup on top of the weight sensor and use a multimeter to check the voltage drop. Add 6mm, 9mm, or 12mm, or all three pills and check the voltage drop over the weight to see if it has changed from the previous measurement. Then take away pills and cup to see if the 75% threshold is crossed. If so, this is a good mark for detecting pills or a lack thereof. b. Try the above test with different numbers of 6mm pills, 9mm pills, and 12mm pills. Repeat until dose detections are perceptible.	**Y**

Figure 17: Load Cell Requirement and Verification Table

Requirement	Verification	Verification Status (Y/N)
1. Must remain locked in its natural state when under 9 VDC is applied.	1. Test whether fail-secure a. Use a multimeter to probe the voltage entering the EM lock at increments of 0.5 volts, starting from 0 VDC. The lock should remain locked.	**Y**
2. Must unlock when 10.5 VDC ± 1.5 VDC is applied to lock.	2. Test unlocking properties a. Use a multimeter to probe the voltage entering the EM lock at increments of 0.2 volts, starting from 8.8 VDC to 12 VDC. The solenoid should retract when 9 VDC is reached and remain unlocked until a value larger than 12 VDC is reached.	**Y**

Figure 18: EM Lock Requirement and Verification Table

Requirement	Verification	Verification Status (Y/N)
1. The WiFi IC must be able to communicate with microcontroller on pill dispensing at 10% of total timeout of pill dispensing process of 5 minutes.	1. Wifi to Microcontroller speed test a. Connect voltage to the microcontroller and wifi modules b. Upload a sketch using the Arduino to the ESP8266-01s c. Connect the wifi module to the microcontroller using TX/RX pins and send variable data through ESP8266-01S program. d. Check that the maximum timeout time when no data is sent is 30 seconds. Look at output logs and print statements to verify that ESP8266-01S program runs in 30 seconds.	**Y***

*Communication was successful, but broke before the demo. Video proof available.

Figure 19: WiFi IC Requirement and Verification Table

Requirement	Verification	Verification Status (Y/N)
1. Must be able to establish port connection for host-client transmission of data from microcontroller to application	1. Connectivity Test a. Create a web server using the Arduino IDE using given code b. Add in credentials and test the web server by entering the ESP IP address into a web browser	**Y**
2. Use POST and GET requests for transferring data, interface between microcontroller to application	2. HTTP updates test a. Using the created web server, connect LEDs with simple web server code and test that LEDs can turn off and on by command on the web browser through IP address.	**Y**
3. Must establish WiFi connection	3. WiFi connectivity test a. Using LED and web browser testing, if successful, then WiFi connection is established.	**Y**

Figure 20: Server Requirement and Verification Table

Requirement	Verification	Verification Status (Y/N)
1. Must be able to connect to the server by WiFi and access updates from the server.	1. Wifi connectivity test a. Echo statements between WiFi point and server ports. Send a statement between the host and client to confirm that the connection has been made.	**Y**
2. Must allow the caregiver to successfully unlock the machine for medication refill and be updated on the status of pill dosage and dispensing.	2. Unlocking test a. Verify that the application is connected to the server and WiFi IC first with Requirement 1. Then, use print statements to verify that the correct data is sent when selected by caregiver on User Interface. Use print statements when debugging on microcontroller to show that output voltage to control EM lock turns HIGH.	**Y**
3. Should allow the caregiver to input prescription details to change the mechanical environment such as dispensing the appropriate medication and locking and unlocking the dispenser to add in more medication.	3. User Input test a. Use print statements to verify user input is received correctly as raw strings then converted to correct types after processing and storing.	**Y**

Figure 21: Application Interface Requirement and Verification Table

Appendix B   Pill Dispensing Process

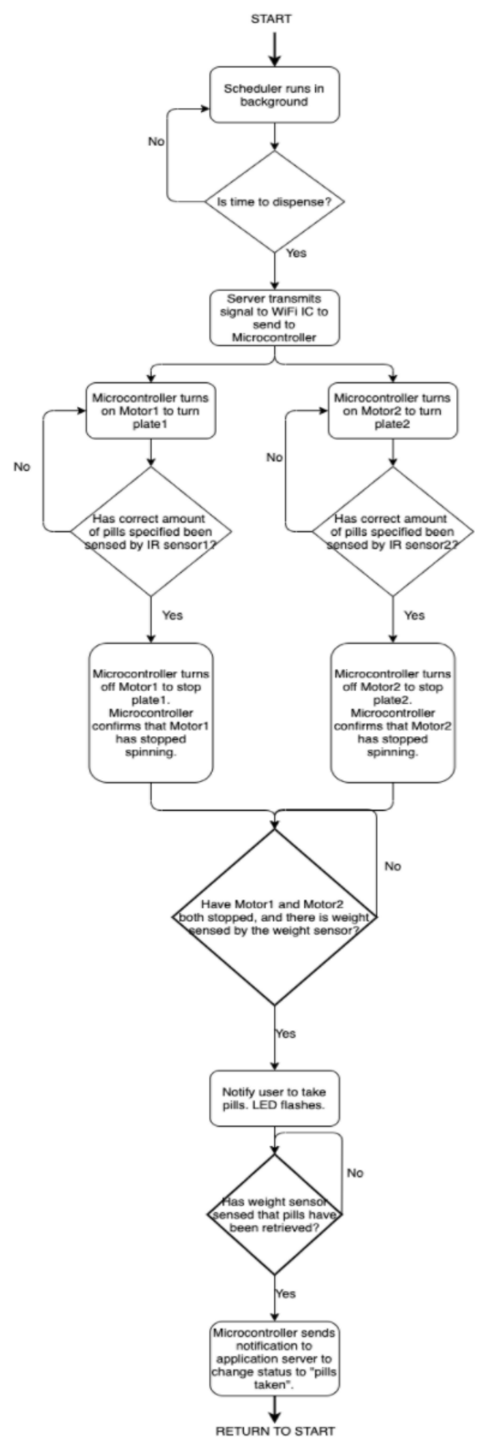

Figure 22: Pill Dispensing Flowchart

Appendix C Caretaker Interface Interaction

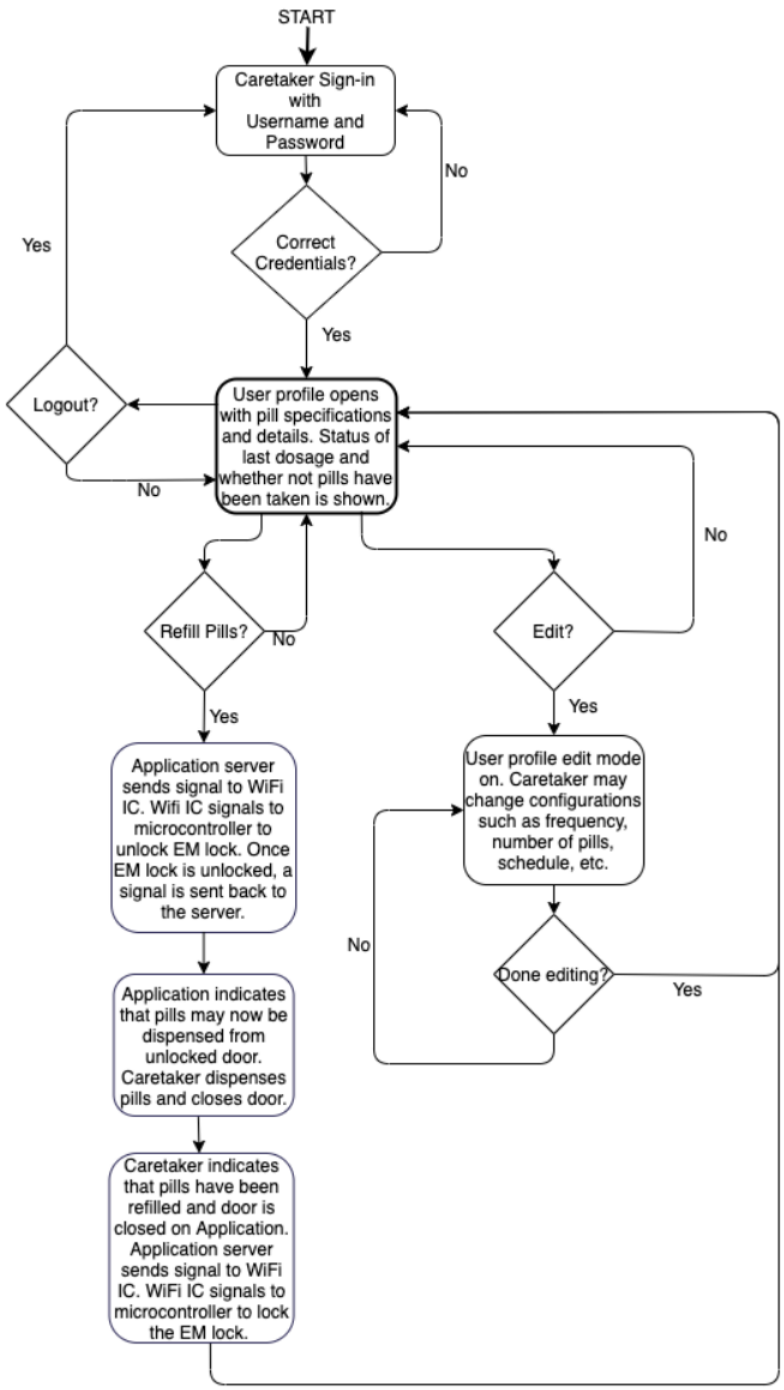

Figure 23: Caretaker Application User Interface Flowchart

Appendix D PCB Design and Layout

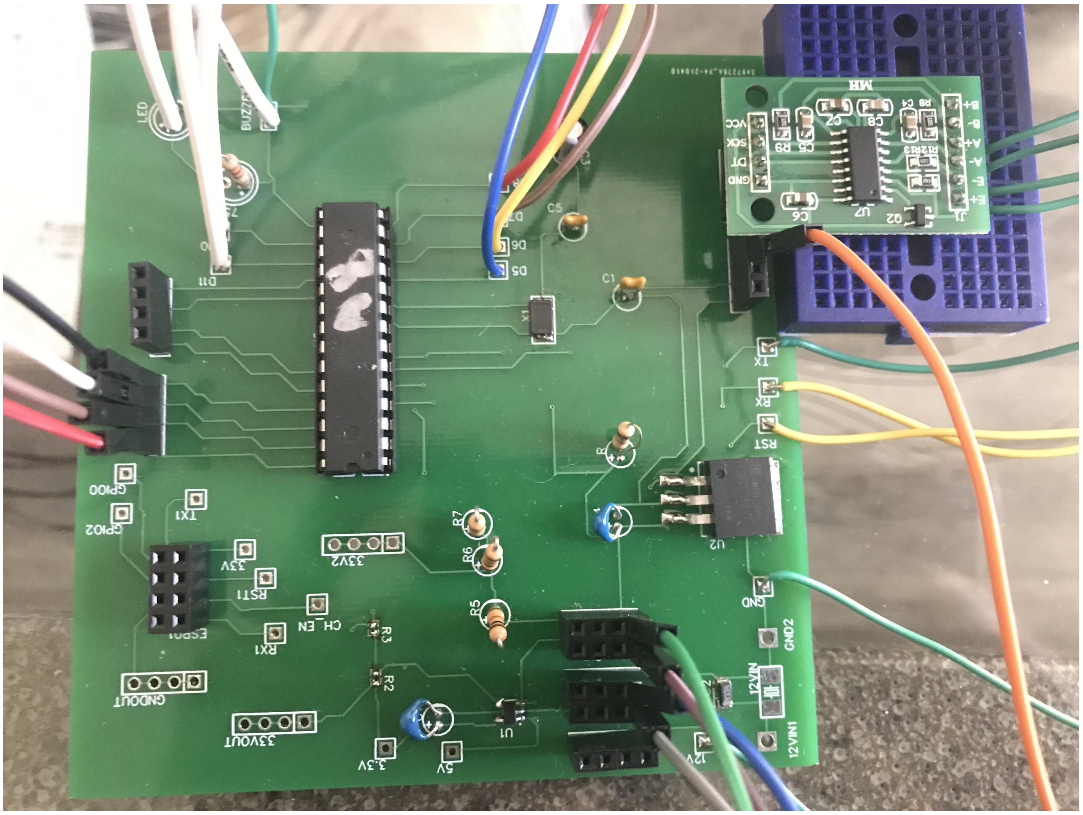

Figure 24: Printed Circuit Board (PCB)

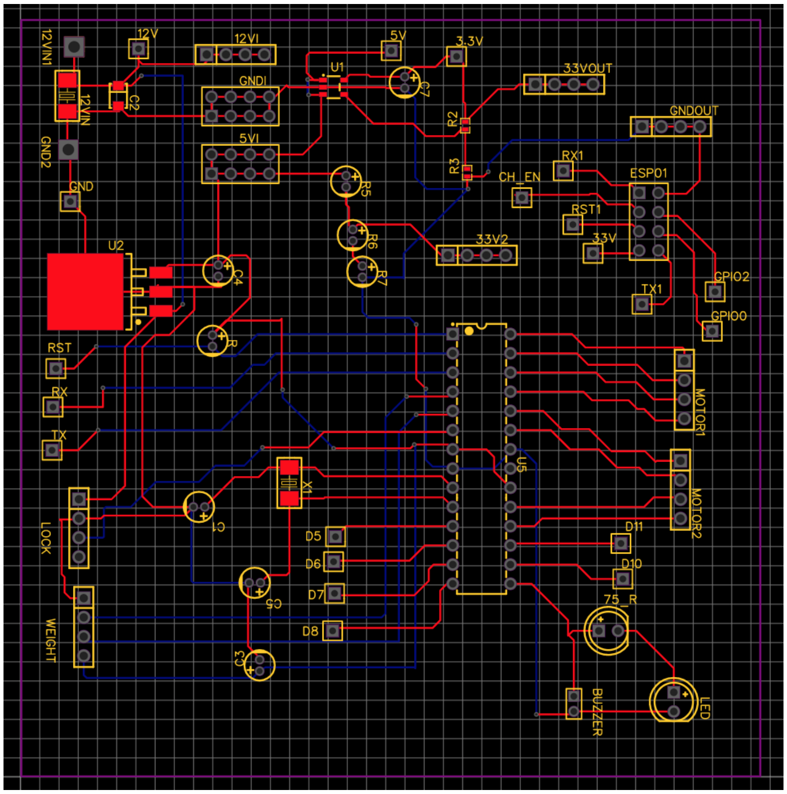

Figure 25: PCB Design Layout

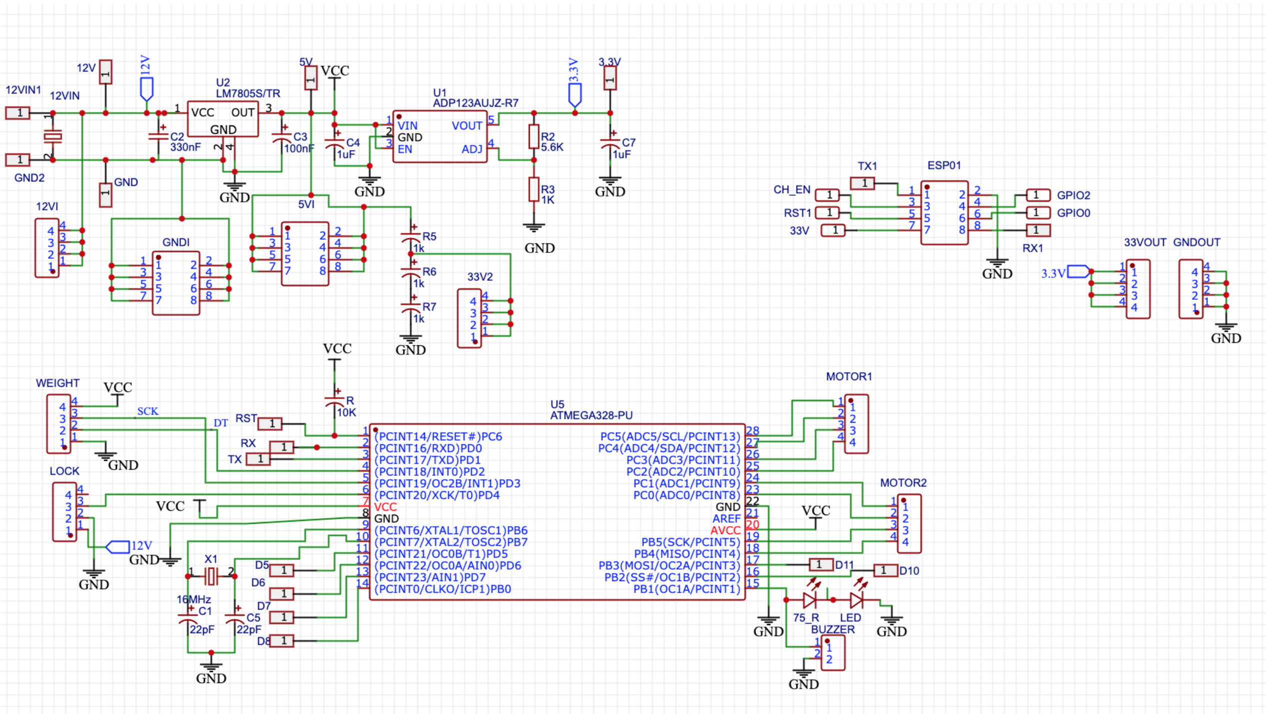

Figure 26: PCB Schematic