Pillsafe Smart Dispenser and App

Team 10
Yanxi Zhu <yanxiz2>, Boyuan Xie <boyuanx2> and Zijin Song <zijins2>
ECE 445 Design Document
TA : Sophie Liu
Fall 2020

Table of Contents

1. Introduction

1.1 Objective

Many people are struggling with drug abuse in the world. This problem has aroused great concern of the government. According to the National Center for Drug Abuse Statistics (NCDAS), 700,000 drug-overdose deaths have happened in the United States since 2000 [1]. Statistics show that $34.6bn has been used for the Federal budget for drug control in 2020 [1]. More often, the overuse occurs in daily life, because it is difficult for doctors and guardians to supervise patients to use the correct number of pills. Although there are several commercial items related to drug use on the market now, most of them are easy to get smashed and cannot prevent people from drug-abusing by just smashing the bottle [2]. Most of the existing smart pill boxes are not for drug abusers. Drug abusers can not control how many pills they take each time and when to take the pill. So overusing happens normally for drug abusers. Fortunately, drug abusing is treatable. Thus, we believe designing and building a device for preventing drug abuse is meaningful and useful.

Inspired by the Health Maker Lab project 5, our plan is to design a smart device to dispense pills. The Pillsafe smart dispenser has the following functions. Firstly, it has a dispenser system which can count exactly the number of specific pills needed for the patient each time to avoid overuse using a photoelectric sensor. Secondly, it has a security system to avoid any tampering with the cap or the bottle. Last but not least, it has a notification system which involves both the hardware and software. On the hardware side, it has a LCD to display instructions and an LED to notify the user when the user cannot take anymore. On the software side, it gives guardians/doctors the right to supervise the patient. The app will notify the doctor or the guardian when the patient uses the drug and if the user has any misuse performances. It also includes an algorithm to study the patients' drug taking habits and provide information for the treatment.

1.2 Background

"The reality is that drug dealing can happen anywhere by anyone because prescription drugs are so easily accessible and their abuse is on the rise. According to the Center for Behavioral Health statistics and quality, an estimated 18 million people, 6% of whom are 12 and older have been reported to misuse prescription medications at least twice within the past year [3]." A 2017 National survey on drug use and health estimates that 2 million Americans have misused prescription painkillers for the first time in the past year belching in an increase in addiction and at worst death by overdose [3]. There are not many products in the industry that can prevent people from drug abusing. For example, Johns Hopkins University mechanical engineering students developed a similar product called "tamper-proof bottle", but it is being proved that they can be easily smashed and cannot prevent people from drug abusing [2]. A shortage of an effective device for drug abusers motivates us to focus on our project. Our device PillSafe Smart Dispenser could prevent people from drug abuse.

1.3 Physical Design

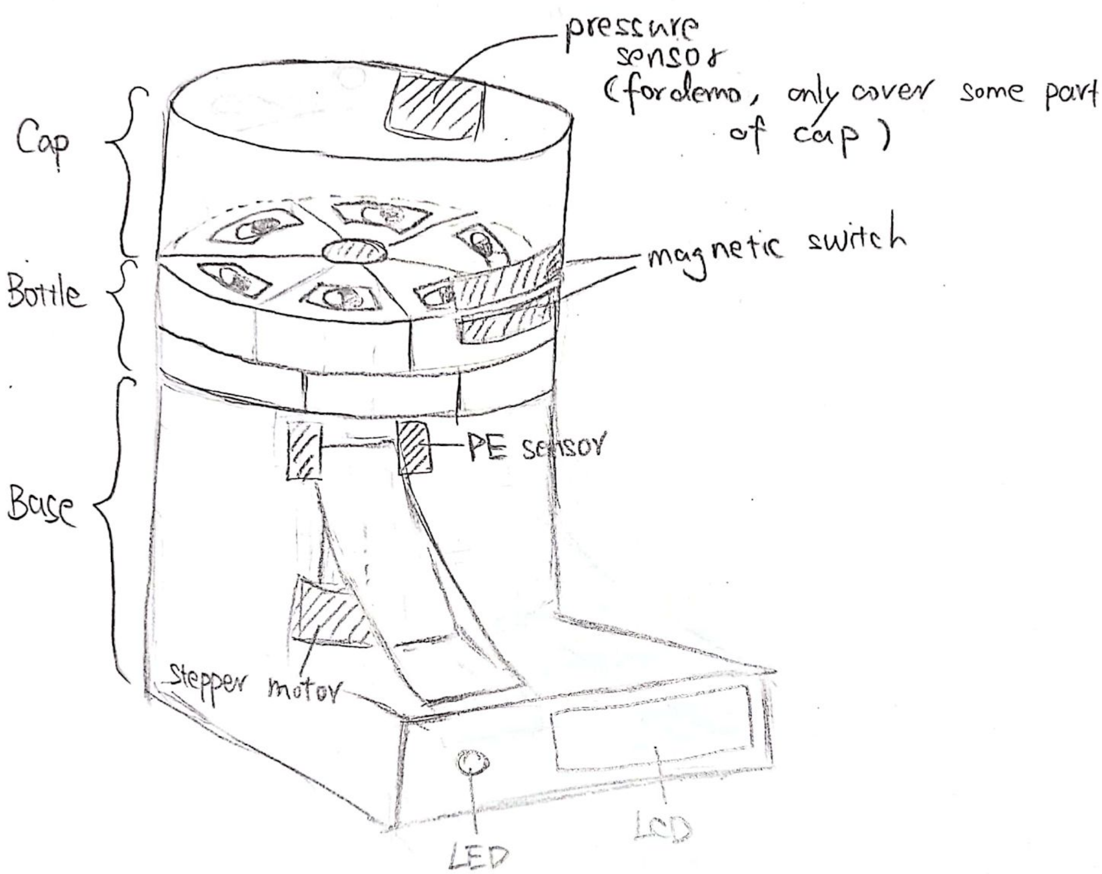

Figure 1: Physical Design of Pillsafe Smart Dispenser

Figure 1 is an abstract sketch of our physical design. The entire dispenser has three parts, from top to bottom: the cap, the bottle and the base. The cap is designed to protect the top of the bottle and prevent users from opening the bottle directly. The cap has a magnetic switch which is attached to the bottle. Once the cap is pulling up, the magnetic switch will be separated and alarms will be triggered. The cap has pressure sensors on its side walls so any actions done to the cap will be detected. The second part is the bottle. The bottle will contain the pill container and two openings. Besides the top opening, an opening on the bottom will be used to dispense pills. To only dispense one pill each time, we use the structure of cascade as shown on the left of Figure 2. The real mechanism is shown on the right of Figure 2. The cascading system can store a high density of pills and easily dispense one pill each time. The third part is the base. The base contains the stepper motor to rotate the cascading system. It also has a plate to hold pills dispensed from the bottle. In front of the plate, it has the LED and LCD to display information.

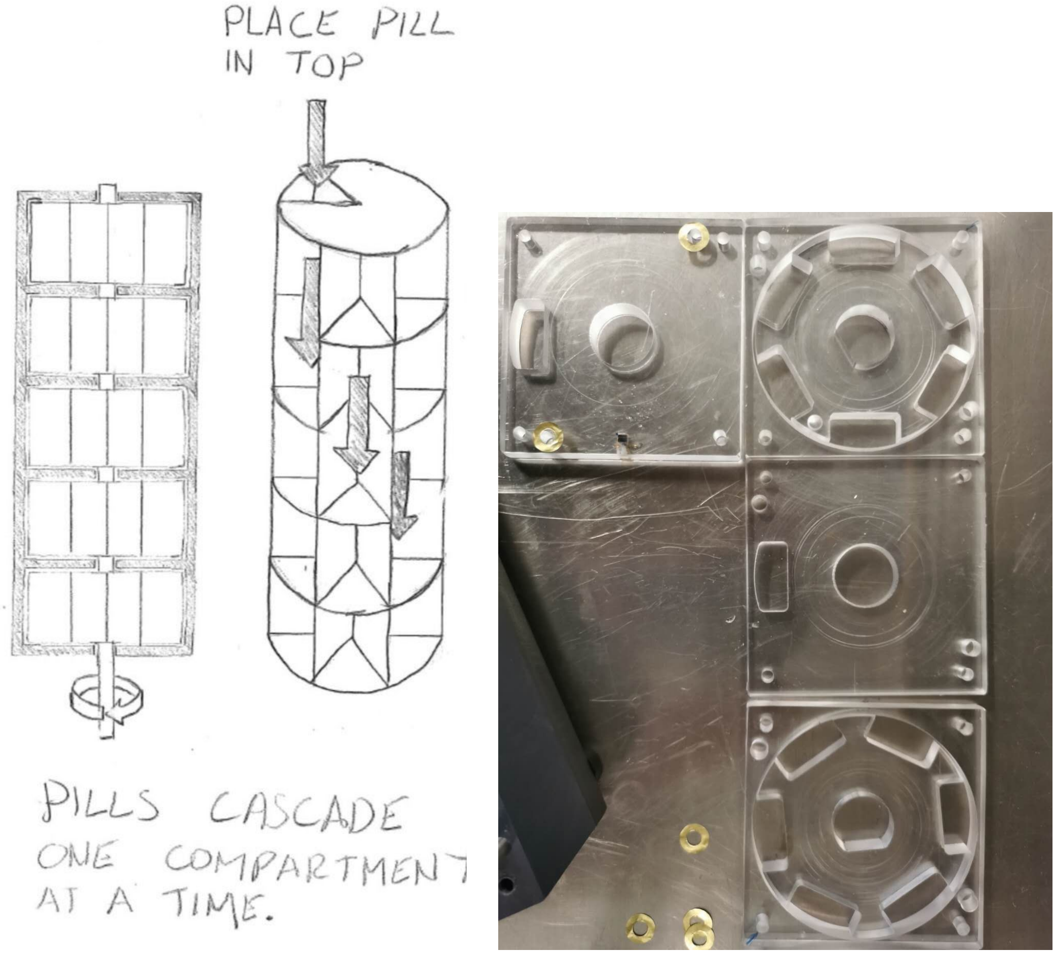

Figure 2: Cascade dispenser model[4] and real products

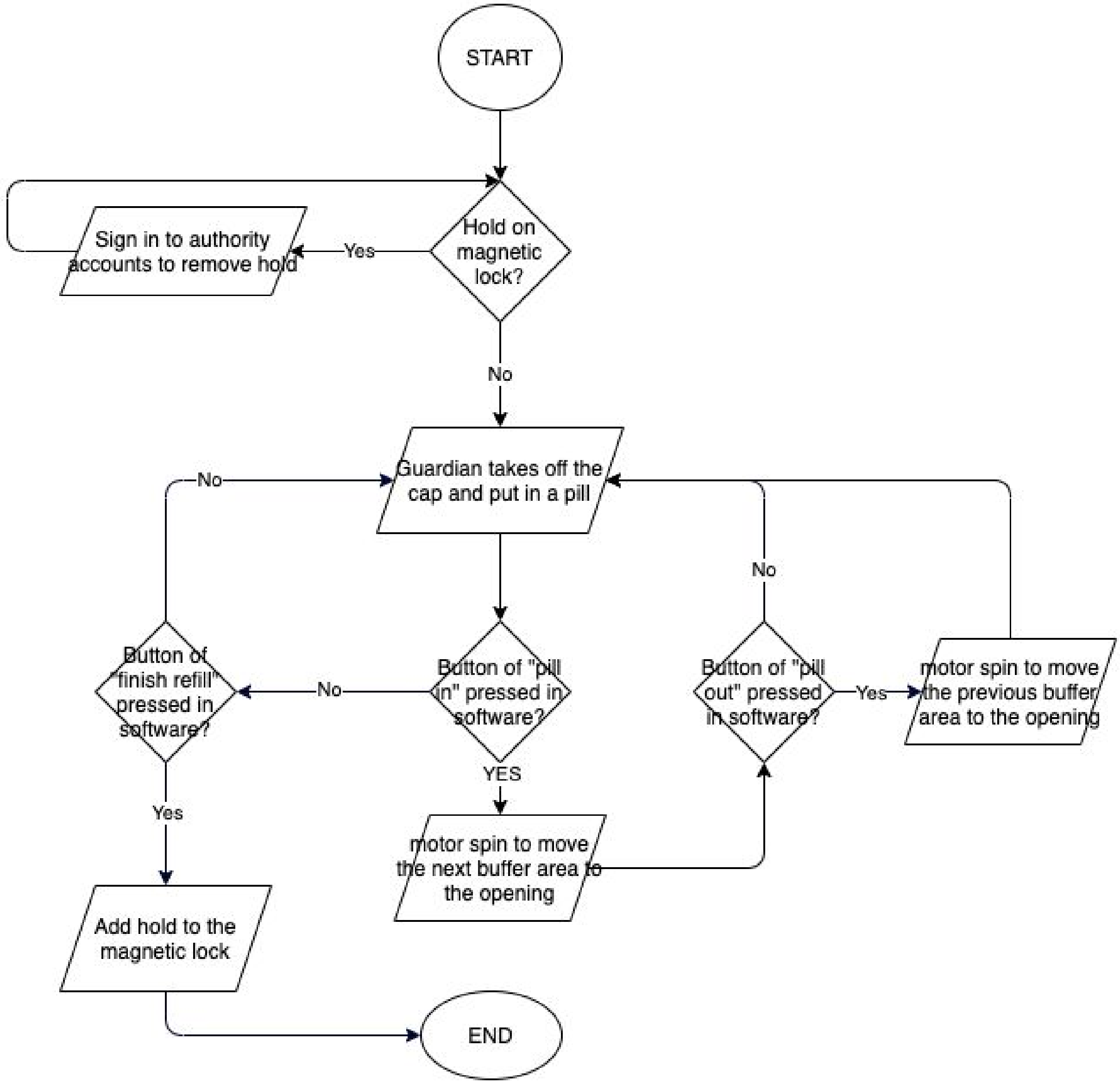

Figure 3: Flowchart of Refilling Pills Process

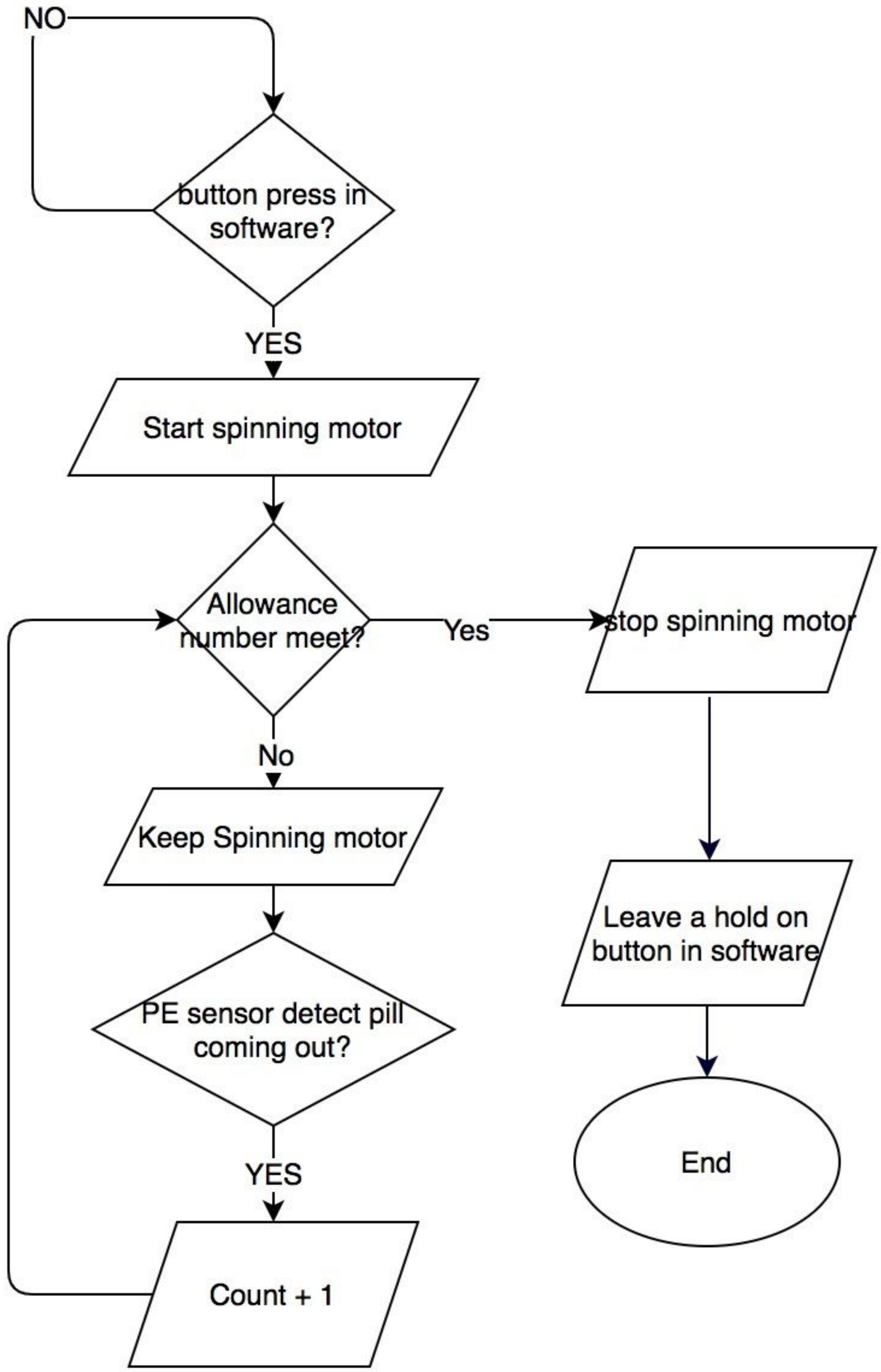

Figure 4: Flowchart of Dispensing Pills Process

1.4 High-Level Requirements List

- The PillSafe bottle can add and dispense different numbers and types of pills. The dispensing success rate should be 90%. In each successful dispensing, only one pill should be dispensed.
- The PillSafe Cap can protect the opening on the bottle. It should be able to detect violations, such as trying to pull up the cap or breaking the cap. The design should be able to send out notifications when violations are detected. Notifications on users' software apps will appear in 3 s, with at most 5 s lagging.
- The Pillsafe app should allow guardians/doctors to set up the allowance number of dispensing. The app should also be able to analyze the user drug using habits, such as the time between two times of dispensing.

2. Design and Requirements

2.1 Block Diagram

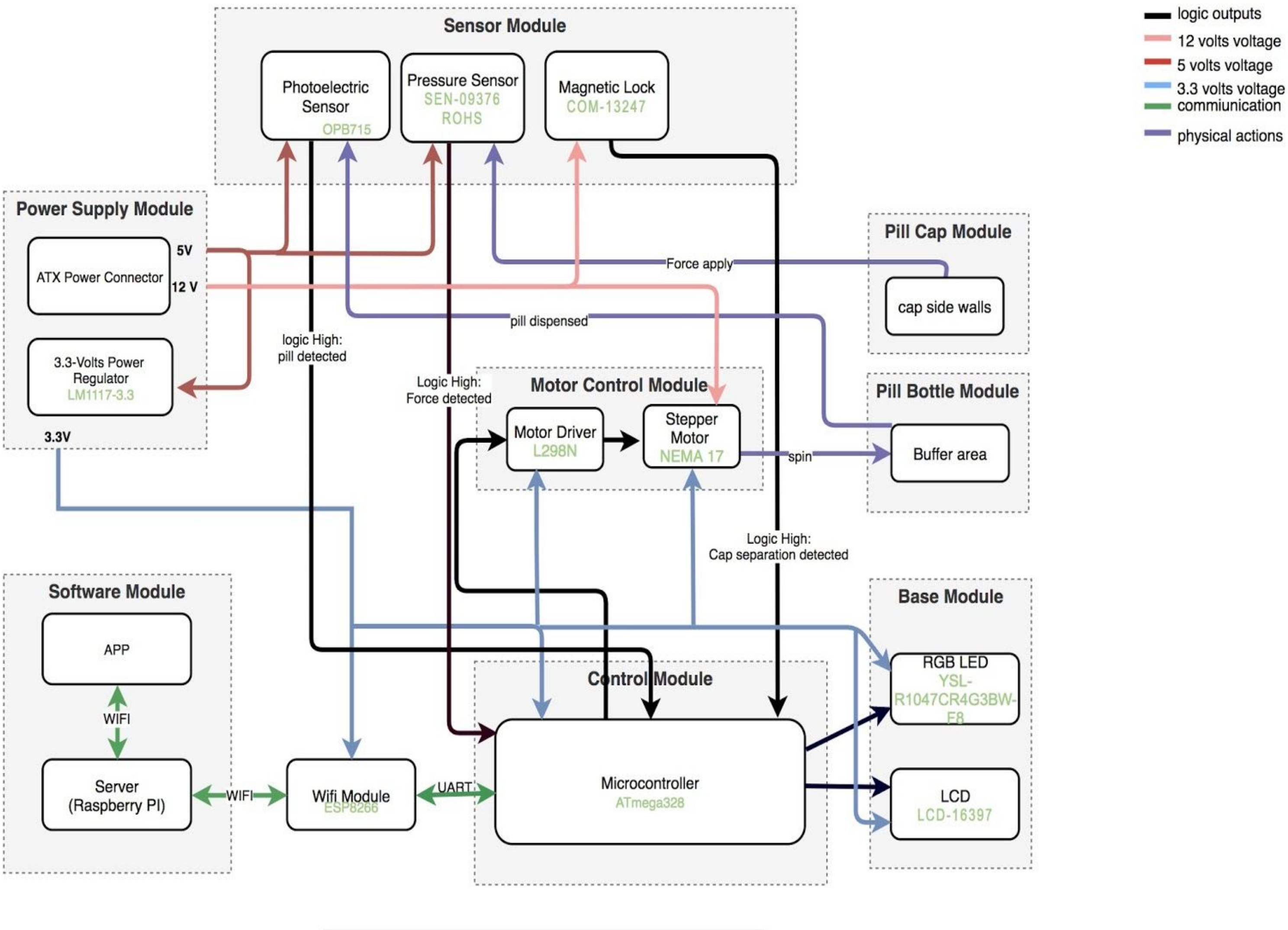

Figure 5: Block Diagram of Our Project

2.2 Functional Overview & Block Requirements

2.2.1 Power Supply Module

- ATX power connector

The bottle is designed to use a wall mount power supply. The reasons for using continuous power supply is to maintain the functions of all system components, especially the pressure sensors to protect the bottle from breaking. Since we have three voltage input requirements for all modules, the power supply module contains a power connector and a voltage regulator. The power connector will convert the wall mount 120V AC to 12V and 5V DC. The 12V should be connected to the motor, the 5V should be connected to the voltage regulator and sensors

Requirement	Verification
1. Power supply provides 12 ± 0.5 Volts in the 12V output pin. 2. Power supply provides 5± 0.25 Volts in the 5V output pin 3. The maximum current should be 2A when the 12V output pin connects to the motor module	Step 1. Connecting the adapter to a resistive load on the breadboard, plug the adapter onto the wall. Step 2. Attach oscilloscope across load Step 3. Probe the output voltage and the current for the load. Ensuring that the Vout is 4.75 to 5.25 volts for 5V output pin, 11.5 to 12.5 volts for 12V output pin. Step 4. After soldering the power connector onto the PCB, probe the voltage input at the voltage regulator, it should be between 4.75 to 5.25 volts

- Voltage Regulator

The voltage regulator is LD1117-3.3. This regulator will provide 3.3 Volts output for a 5V input. This version is a fixed output voltage regulator, nominal voltage is 3.3 Volts, which is the standard voltage for the Wifi module, LED, LCD and the microcontroller.

Requirements	**Verifications**
1. Supply Vout at 3.3 ± 0.1 V 2. The maximum output current should be 800mA.	Step 1. Attach 33 k resistor as load Step 2. Attach oscilloscope across load Step 3. Supply regulator with 5V DC Step 4. Ensure output voltage remains 3.2 V and 3.4 V Step 5. Measure all the current using oscilloscope, ensuring the error within 5%

2.2.2 Pill Cap module

- Pressure sensor

Pressure sensor is used to detect any attempts that will destroy the cap. We use plastic as the cap material. The cap will be 3D-printed with the material of Acrylonitrile butadiene styrene (ABS), this material is tough and hard to break by hand. We will set the alarming value high so that it will not send out an alarm due to normal actions. The pressure sensor will work as an Force sensitive resistor, together with a 3.3k ohm resistor, a simple voltage divider circuit can be built and connect to the analog input pins on the microcontroller. In this case, it is connected to pin PC0.

Requirement	**Verifications**
1. The resistance should be at least 100k ohm when there is no force applied. 2. The resistance should be at most 0.5k ohm when above 100N applied on it 3. When the pressure sensor is left untouched, the voltage received at the microcontroller should be close to 0V 4. When the pressure sensor is hardly pressed, the voltage received at the microcontroller should be close to 5V	Step 1. Insert the pressure sensor into the wall Step 2. Force the wall by hand and measure the resistance of the FSR by ohmmeter Step 3. Check that the resistance value in the previous step will not trigger the alarm system Step 4. Force the wall by using scissor and measure the resistance of the FSR by ohmmeter Step 5. Check the resistance value in the previous step that it is small enough to trigger the alarm system. Step 6. Repeat step 2 to 5 and measure the Voltage at pin PC0 on the microcontroller.

- Magnetic lock

This is the Magnetic Switch/Lock Set, a small reed switch assembly specifically designed to alert for aperture opens between the cap and the bottle. One half of the assembly is set on the cap and the other attached to the bottle. When the switch set is separated from each other the contact is broken and triggers an alarm.

When each magnet peace comes within 20mm of each other they complete the circuit with the internal reed switches. Each Magnetic Switch will be powered with 12V voltage input from the power connector directly.

Requirements	Verifications
1. The state of open/close of the switch should be sent to the microcontroller. 2. The state to trigger an alarm should be controlled by the software. Only when there is a hold in the software, separation will trigger an alarm. When no hold is applied, in other words, during the refilling process, no alarm will be triggered.	Step 1. Connect the magnetic switch and the microcontroller on a breadboard. Step 2. Open/close the switch, a Voltage should be measured at the analog input pins on the microcontroller. Step 3. Repeat step 2 with the hold on in the software. Step 4. An output at the RX pin on the microcontroller should be detected. Step 5. Repeat step 2 with hold off in the software Step 6. A different output at the RX pin on the microcontroller should be detected.

2.2.3 Pill Bottle module

- Opening

The opening should be the only way to put in pills. This open should be the same size as the buffer area, which is 32mm*13mm with a 5% variance. It should be able to fit the normal sizes of pills on the market.

Requirements	Verifications
1. The pill will fall into the buffer area through the opening.	Step 1. All the mechanical functions can process well. Step 2. Prepare 5 common types of pill sizes on the market. Choose capsule size 000,00,0,1,3 Step 3. Verify that each type of pill can pass through the opening

- Photoelectric Sensor

The photoelectric sensor is used to count the pills dispensed. It is a retro reflective photoelectric sensor with a collector and a receiver as a whole. It should be able to detect the pill successfully when the pill falls off from the bottle's bottom opening. It will then notify the microcontroller to add the count once and check with the allowance number. that the upper door needs to be closed and the bottom door can open.

Requirements	Verifications
1. The photoelectric sensor must be able to detect items passing through within 5 mm +/- 1 mm. 2. The photoelectric sensor must be able to provide a logic HIGH output to the microcontroller.	Step 1. Using the oscilloscope to measure the voltage output of the sensor when placing capsules at a distance of 5mm Step 2. Record the time frame of logical 1 output from the PE sensor Step 3. Repeat step 1&2 for all capsule sizes 000,00,0,1,3.

2.2.4 Base module

- Stepper Motor

Motor is used to control the rotation of the bottle. This motor should have a low voltage requirement so that its speed can be slow. The motor should also be able to spin bidirectional so that if the guardian puts in anything wrong, they can rotate the buffer area back to the opening to pull out through the opening. We use a NEMA 17 stepper motor here.

- Motor Driver

To provide a peak current of 1.68A for NEMA 17, We use a L298N motor driver here to control the stepper motor. L298N can hold up to 2A as peak current. It also has 2 phases and four wires, which is able to power up NEMA 17.

Requirements	Verifications
3. The motor should be able to rotate 60 degrees in 32-34 steps. 4. The motor driver should be able to keep the motor rotating at a proper speed so that the pill can safely land to the next level of the bottle.	Step 4. Connect the motor with the motor driver in a test circuit. Step 5. Give a PWM input signal to the motor driver. Step 6. Test the motor with different step numbers. Measure the rotating angle. Step 7. Test the motor with different duty cycles and choose the most appropriate timeframe that can allow pills to pass safely. Step 8. Repeat step 2 and 3 with the software command of rotating backwards.

- Pill holder

This should be a cuboid space that can hold all the pills stored in the pill bottle.

- LED

LED lights are used to notify the user the situation of dispensing. LED should have three modes: red, yellow and green. Green will light up when the pill dispensing process starts. Yellow will light up when the pill is dispensed to notify the user to take the pill. Red will light up when the entire dispensing process is finished or detecting violations. The LED will be a common cathode RGB LED, and the color legs are connected to the digital output pins on the controller.

Requirements	Verifications
1. The common Cathode RGB LED can light up three colors with 3.3 voltage inputs for each individual leg. 2. Green light should light up when the motor spins and the photoelectric sensor detects nothing 3. Yellow should light up when the photoelectric sensor provides a logic HIGH 4. Red should light up when the process is finished or detecting violations from pressure sensors and magnetic locks.	On the breadboard, connect the cathode to ground and RGB pins to a 3.3 volts DC voltage supply. Verify that the LED can light up correctly.Using the Arduino Development Board to do the testing with the driver code Step 1. In the driver codes, set the steps of motor to 10 Step 2. Upload the driver codes and power up the Arduino Board with 3.3V at VCC. Step 3. Verify if the green light is on. Step 4. Probe the other two LED lights to see if the output is LOW Step 5. Manually set the output of the PE sensor to HIGH. Check that the yellow light is on. Repeat step 4 for the safety check. Step 6. Manually set the analog input from PC0 on Atmega328 to a high voltage, and verify if the red light is on. Repeat step 4 again. Step 7. Manually set the magnetic switch indicator to Open in the driver code. Verify that the red light is on. Repeat step 4 again.

- LCD Screen

The LCD screen is used to display the instructions . It will print words such as “dispensing start” “ 1 pill dispensed, please take the pill” “dispensing finished” “Stay away from the bottle!” “Put Cap Back” “ Please refill” “Wifi Connected” and etc based on situations and contents. We choose sparkfun electronics LCD-16397. This LCD has a size of 8.5 cm*3.5 cm, which is fittable on the base’s front edge. It is also a 16*2 LCD, so we can display enough information on it.

Requirements	Verifications
1. It should display different sentences based on the states in the entire process 2. The display area should be clear to see 3. The time delay of updating should be less than 5ms.	Step 1. Using the Atmega328p to test the LCD system. Step 2. Uploading the driver's code to the microcontroller and manually set the stage to "start" Step 3. Probe the output pin and verify that the output is HIGH Step 4. Verify that the sentence shown on the screen is correct and clear Step 5. Change the state to "finish" in the drivers' code, repeat step 3 and 4 Step 6. Change the state to "violation" in the drivers' code, repeat step 3 and 4 Step 7. Change the counting number and then change the state to"pill dispensed" in the driver's code, repeat step 3 and 4.

2.2.4 Control Module

This module contains the microcontroller. The microcontroller gets data inputs from the sensors and the software. It outputs data to the display module and the wifi module. It should be able to transfer the analog input from the photoelectric sensor and the pressure sensor to digital states and send it to the display module. It should also transfer data between the control unit and the software unit. The microcontroller should have little power consumption.

- ATmega328p

  pin connections:
  - VCC is connecting to the 3.3V output of the voltage regulator
  - Pressure sensor on the cap connects to the analog input pin PC0
  - Magnetic switch on the cap connects to the analog input pin PC1
  - Photoelectric sensor on the bottle connects to the analog input pin PC2
  - The L298N's four inputs are connected to the digital pins PB0,1,2,3

- LED Green is connected to digital PWM output pin PD2; LED Yellow is connected to digital PWM output pin PD3; LED red is connected to digital PWM output pin PD4
- LCD’s RX pin is connected to digital output pin PD7
- Wifi module TX is connected to digital pin RX, PD0
- Wifi module RX is connected to digital pin TX, PD1

Requirements	**Verifications**
1. The microcontroller should communicate over a standard I2C bus to all the sensors 2. The microcontroller should be able to communicate over UART to the wifi module 3. The microcontroller should be able to control the LCD display	Connections: All the connections are verified using the ATmega and driver codes are tested on the board as well. Communications: This will be verified in the wifi module. Outputs: Step 1. Power the board with 3.3 Volts to VCC Step 2. Write the driver code for the LCD display. Step 3. Connect the microcontroller to all the modules Step 4. Repeat the verification steps for the LCD module, this time do not manually change the driver code.

- Schematic and PCB

  Schematic and PCB for the microcontroller connections. All modules instead of the power supply are connected to the microcontroller through connectors.

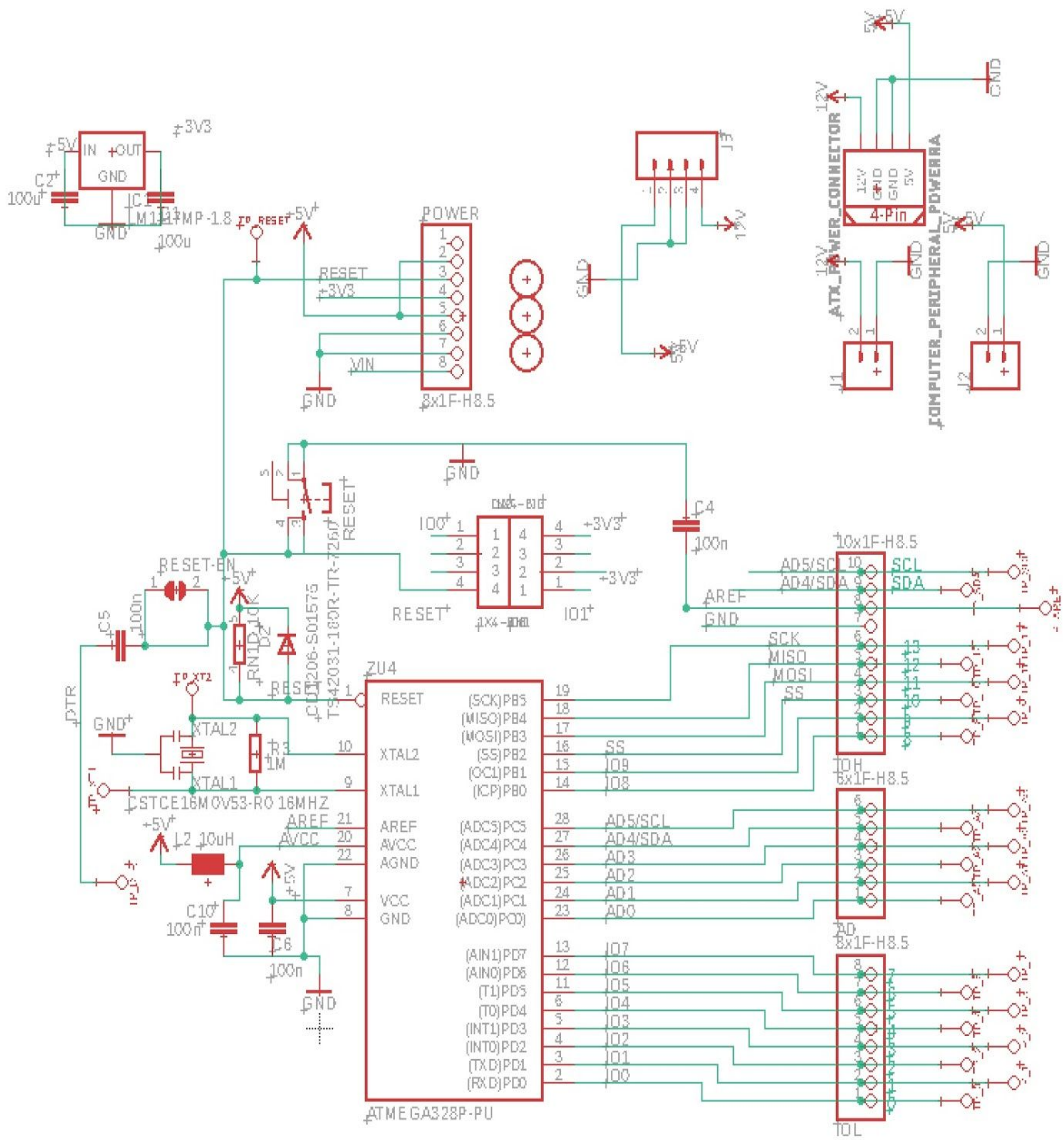

Figure 6: Schematic diagram of Microcontroller

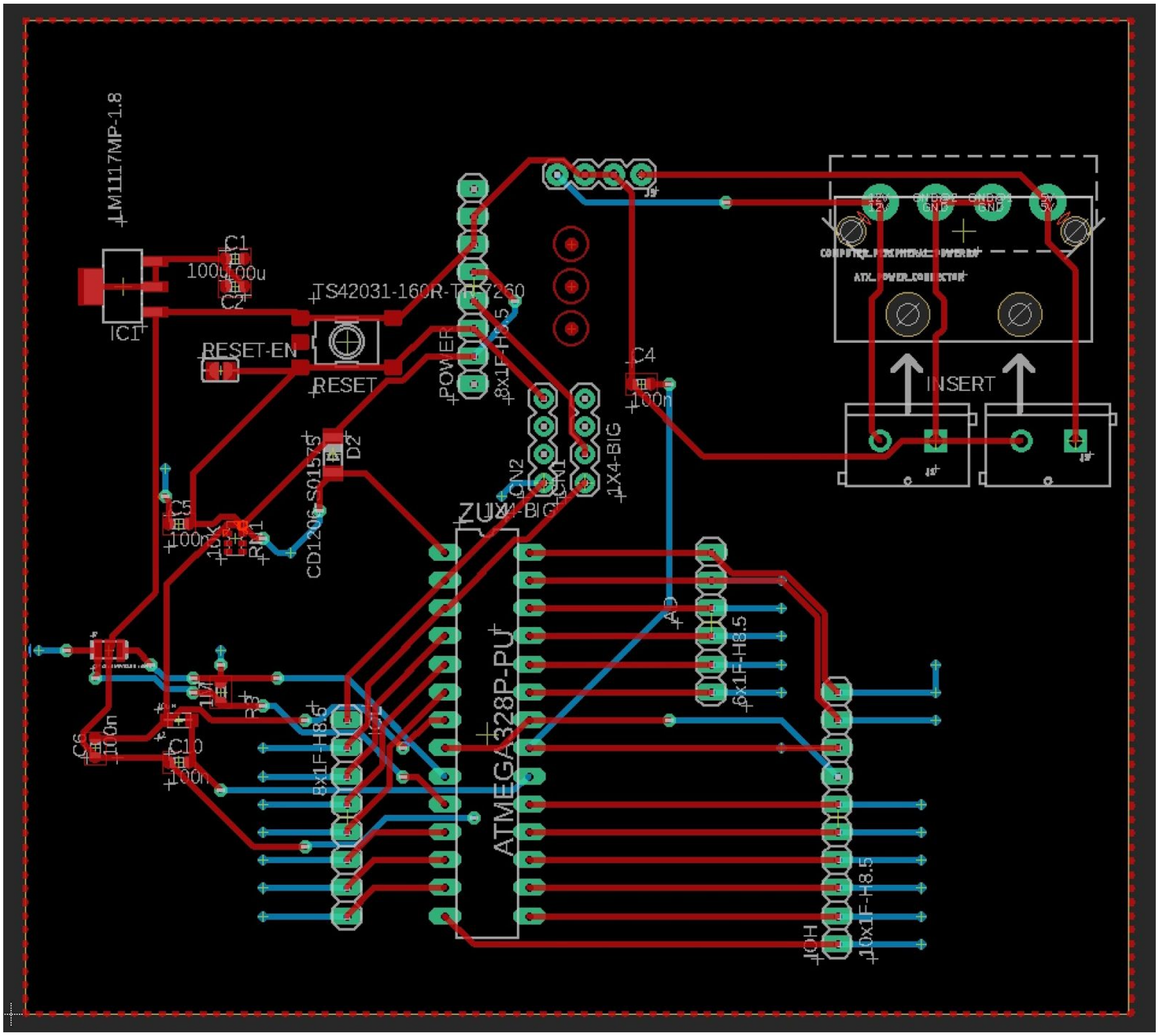

Figure 7: PCB design of Microcontroller

2.2.6 Software Module

The application will have an interface with the display module and the control unit. It will be able to use the data from the microcontroller to detect violations. It will also be able to dispense the pill when the user press the button in the software. The app will have two set of user accounts. One for the patient and the other for the guardian and the doctor. For both accounts, they can receive notification of violations and see the data of drug taking. For authority accounts, which can only be logged in by guardians and doctors, they have the right to set up the allowance number of pills taken. Their account also has the right to remove hold on the magnetic switch and do the refill process. It must be able to send the alarm to the doctor/provider when the patient is misusing the drugs within 3 seconds.

Requirements	Verifications
1. The software can detect the power on/off stage of the dispenser 2. After detecting that power is on, send instructions on connecting to WiFi 3. Send alarm to users/doctors/guardians when any misusing happen, include: opening the cap, breaking the cap and cutting off power supply 4. All interfaces on the software should work properly. 5. The communication time between the software and the microcontroller should be at most 5 seconds.	Step 1. Connect the wifi module to an arduino uno test board. Step 2. Connect the wifi. Step 3. Open the software and create the user account. Step 4. Add authority accounts in the software, for example, doctors or guardians. Step 5. In the authority accounts, set the pill allowed for the patient. Step 6. Run the drivers' codes to verify the allowance number is inputted when the dispensing button is pressed. Step 7. Verify that the initial stage for the magnetic lock driver code is "hold on" Step 8. Change stages of dispensing to "finish", the software should receive notifications Step 9. Change pressure sensor outputs to HIGH, the software should receive notifications.

2.2.7 Wifi Module

The microcontroller in the PillSafe Cap uses Wifi protocol to transfer the data between the control unit and the software module. It must be able to allow the data from the microcontroller to be sent via UART and be able to be accessed on a Wifi network. The connection should be stable within 10 meters.

Requirements	Verifications
1. The wifi module should be able to connect with the microcontroller under a standard UART bus 2. The wifi module should be able to connect to phone via Wifi IEEE 802.11b/g/n and exchange the data with server 3. The wifi module should be able to connect to a phone for a range of at least 10 meters without blocking 4. The wifi module should be able to send data from the server set numbers in the controller's driver code 5. The wifi module should be able to upload data from the microcontroller to the server	Wifi connections: Step 1. connect the wifi module to the microcontroller via RX/TX pins Step 2. Use the Arduino Development Board to test driver codes' functionality, which includes sending data from server to microcontroller and receiving data backwards. Step 3. Using a Wifi hotspot and try to connect the wifi module to this Wifi Step 4. Test through HTTP protocol Step 5. Repeat step 1-4 and adding distances until the module fails to connect

Module Name	**High Level Requirement**	**Points**
Power Module	● This module should supply enough steady power to the circuit. ● This module should supply 11.5 to 12.5 Volts for the motor and magnetic lock, 4.75-5.25 Volts for sensor modules, and 3.2-3,4 Volts for all other modules.	**5**
Control Module	● This module should communicate with the server through the wifi module under operation. ● This module should gather analog inputs from sensor modules and create correct digital outputs to display modules. ● This module should create correct outputs to control the motors.	**7**
Motor Module	● This module should use motors to control the rotation of the Cascading dispensing system	**10**
Display Module	● This module should display different color LED lights for different stages to notify users. ● This module should display the wording instructions on the LCD screen.	**5**
Sensor Module	● This module should do the correct counting of pills dispensed using the photoelectric sensor. ● The pressure sensors should be able to detect unreasonable actions on the cap. ● The magnetic lock should be NO and correctly indicate separation actions.	**10**
Wifi Module	● This module should create communication between the hardware and the software through the server.	**3**
Software Module	● This module should be able to send default settings to the controller. ● This module should be able to send notifications while detecting faults from users. ● This module should be able to receive notifications for different stages.	**10**

	• This module should be able to let guardians/doctors do the refill. • This module should contain different account categories for users and guardians/doctors. • This module should be able to do some data analyzing of the user's drug taking habits.	
Total		**50**

Table 1: Point assignment

2.3 Project Differences Analysis

The current similar product is the "tamper-proof bottles" designed by Johns Hopkins University mechanical engineering students. [2] The primary function of their product is "lock." They tried to lock the pill box and only dispense the pre-set number of pills. Their ideas include:

- Med-O-Wheel SECURE: this locked pill box dispenses the prescribed dose of medication at a pre-set times
- Timer Cap: this cap is equipped with a built-in-timer to count the time of using drugs.
- Safer Lock: The pill bottle includes a four-digit combination lock in the cap.

Even though their ideas are good, they do not provide solutions to the issue. Users can easily destroy the box and the doctors/guardians have no way to know. Also, this product is not able to improve the drug abusers' performances.

Different from their ideas, we give higher priority to the doctors/guardians. The performance of the user of our pillsafe bottle is presented to the authority accounts for supervision. Especially when the pressure sensors and magnetic switch work, the authorities will know right away and stop the false actions.

More importantly, To help the user improve their resistance to overusing pills, we purposely prevent them from taking out pills freely. The cap is sealed when the magnetic switch is in the"hold on" state. The patient has no way to open the cap. It can only be opened by guardians/doctors to refill the pills. Patients can only get access to pills through the opening in the base area where pills are dispensed from the bottle area one at a time. To further prevent overuse of drugs, our product is able to count the pills. Along with the algorithm to analyze the using habit, we can provide help to the treatment. In the short run we contain the patient from overusing the pills, and we train them with good using habits in the long run.

Last but not least, our product can be applied to different sizes and kinds of pills since we design the holding place for pills in the bottle area suitable for most pills in the current market.

2.4 Tolerance Analysis

2.4.1 Photoelectric Sensor

One important factor to success of our project is that we can count the exact number pills being dispensed with high accuracy (need to be close to 100%) in real time. We need to make sure that our photoelectric sensor can give the correct result. In order to provide high accuracy, we want to determine the best positioning of our photoelectric sensor. We find our photoelectric sensor has two ways of receiving lights: from front or from a side. Below are the pictures of their trigger points vs displacement graphs.

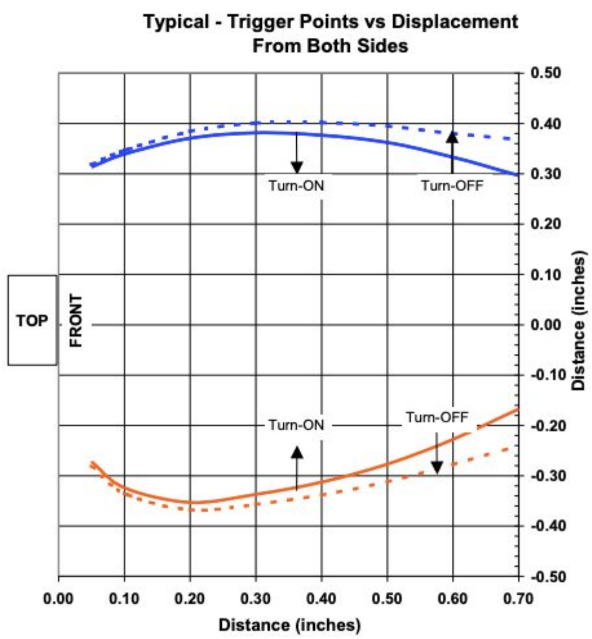

Figure 8: Plot of Trigger Point vs Displacement of PE Sensor when using receiver in the front

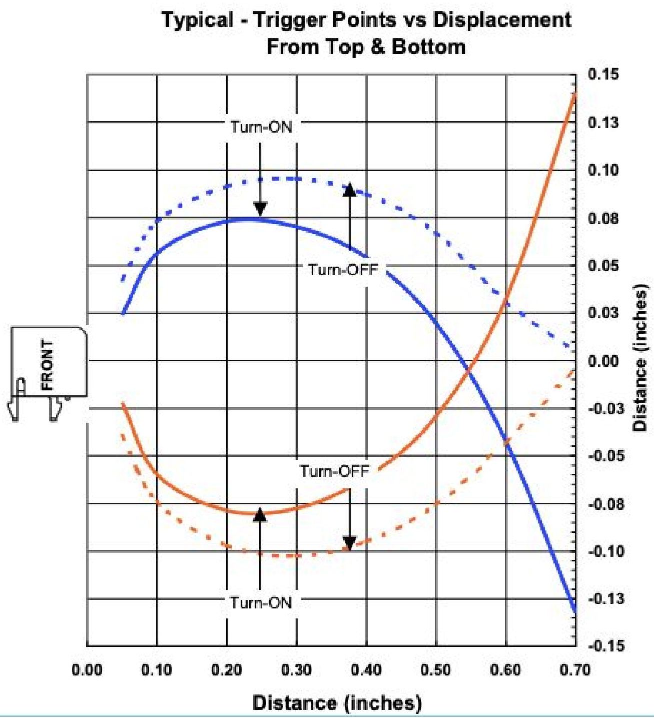

Figure 9: Plot of Trigger Point vs Displacement of PE Sensor when using receiver in the side

For both graphs, the x-axis and y-axis are the distance between the item and the sensor in inches. In our cases, one axis will be fixed, because it is based on the distance from the sensor to the bottle's bottom opening. In our design, the fixed axis is the x-axis and the value is about 0.4 inches.

For this x-value, as shown in Figure 8&9, we find out that using receivers in the front can detect the item much earlier than the other way. The front receiver can detect the item from 0.3 inches, while the side receiver can only detect the time from 0.08 inch. In our design, the earlier the sensor detects the pill, the safer it can transfer the data to the microcontroller. So we will choose the placement shown in Figure 8.

2.4.1 Stepper Motor and Its driver

Control of the stepper motor is essential for our project. Since we are using Atmega328P, luckily we can use the library for Arduino. Arduino IDE standard includes the *Stepper* library, which is able to control the stepper motor.

To demonstrate the motor, we need to load the *stepper_oneRevolution* sketch in the *Stepper* library.

Since we have six buffer areas in the cascading dispensing system, we want the motor's each revolution to be around 60 degrees, so that we can use the sketch mentioned above to achieve the dispensing function. From the datasheet of NEMA 17, the degree per step is 1.8 degrees. So we need to set the value as the following:

const int stepsPerRevolution = 34;

The step is set to 34 because the motor has at most a 1 degrees backlash. So 1.8*34-1 = 60.2 degrees, which is accurate enough for our design.

We also need to set the speed. The speed is preset to 60 RPM. In our design, this is too fast. We want the counting process to be accurate, so we need to slow down the speed. 3s per revolution is expected. This can be set as the following:

myStepper.setSpeed(20);

Finally, in our design, we give guardians the right to rotate the plate backwards so that they can take the pill out when they mistakenly do the refill process. This can be achieved by the following commands:

myStepper.step(stepsPerRevolution); // for clockwise
myStepper.step(-stepsPerRevolution); // for anti-clockwise

2.5 COVID-19 Contingency Planning

If this course goes all online, we will add more complex functionalities in the software part. We would add more data analysis functions and personal design to our App. People can react differently to the same drug based on their various absorptions to the medicine and their age variance. Based on the patient's information and his/her history of taking medicine, our app will be able to provide scientific data analysis and suggestions on his/her future drug use.

We purposely simplify the task on soldering, so if this course goes all online, we are still able to hook up everything on the breadboard and do the demo.

3. Cost and Schedule

Type	Manufacture	Part Number	Quantity	Unit Cost	Total Cost
Magnetic Switch	SparkFun Electronics	COM-13247	1	$3.50	$3.50
Microcontroller	Microchip	ATmega328	1	$2.08	$2.08
PE Sensor	TT Electronics	OPB715Z	1	$6.84	$6.84
LED	CHINA YOUNG SUN LED TECHNOLOGY CO., LTD.	YSL-R1047CR4G 3BW-F8	1	$2.05	$2.05
LCD	SparkFun Electronics	LCD-16397	1	$19.95	$19.95
Voltage Regulator	Texas Instruments	LM1117IMPX-3.3	1	$1.10	$1.10
Pressure Sensor	Interlink Electronics	FSR 406	1	$11.25	$11.25
Stepper Motor	OSM Technology Co.,Ltd.	NEMA 17	1	$40.05	$40.05*
Motor Driver	Qunqi	L298N	1	$2.47	$2.47
Wifi Module	Espressif Systems	ESP8266	1	$10.00	$10.00
Total cost	**$59.24**				

Table 3: Component Cost

* The stepper motor is provided by the machine shop for free

Week	Yanxi	Boyuan	Zijin
Week 1: 09/28/2020	Work on the design document. Work on tolerance analysis.	Work on the design document. Work on R&V table.	Work on the design document. Work on software design.
Week 2: 10/05/2020	PCB design Help with order parts	Schematic design and simulation Place order for parts	Writing code for the app and controller
Week 3: 10/12/2020	3D-printing the cap Test Mechanism	Solder part on PCB	Writing code for the app and controller
Week 4: 10/19/2020	Test sensors and displays	Test power units and circuits	Writing code for the app and controller
Week 5: 10/26/2020	Testing and make adjustment		
Week 6: 11/02/2020	Prepare for mock demo		
Week 7: 11/06/2020	Final integration and testing for both software and hardware		
Week 8: 11/16/2020	Prepare for final presentation and final report submission		

Table 4: Schedule

4. Ethics and Safety

It is important to consider possible ethical and safety issues involved with a technical project. Since our app is able to get first hand information about users' drug habits, we are responsible to protect our users' privacy. To protect our users' privacy, we designed a password protected login system and every user is required to sign a privacy agreement for other users when they sign up. Also, our project is designed for public health. Every feature in our project is designed to prevent drug misuse. These are implement of #1 of IEEE code of Ethics: "to hold paramount the safety, health, … , to protect the privacy of others, ... [7]"

Despite the facts talking above, we should also consider the user's individual/personal rights. Since this application will give the absolute control to the doctor/provider side, there might exists discrimination, financial or sexual orientations which violating #8 of IEEE code of Ethics [7]. Even though we are not able to control the behavior on the administrator side, we will give users/patients the right to stop the program under emergency situations.

We will guarantee our device is safe for our users. To accomplish that, we will pay close attention to points that can possibly cause safety issues to users later. Firstly, we need to be careful of the voltage we apply to the battery, since the battery could explode and cause danger if an overcharge happened. Secondly, the smart cap contains mechanical parts for the user interface module. We would make sure that these mechanical parts are safe to physical bodies. These are another implement of #1 of IEEE code of Ethics: “to hold paramount the safety, health, ... [7]”

References

[1] National Center for Drug Abuse Statistics, *NCDAS: Substance Abuse And Addiction Statistics*. 2020. [online] Available: NCDAS, https://drugabusestatistics.org/. [Accessed: Sep. 14, 2020].

[2] American Addiction Centers Editorial Staff, What's New in the World of Tamper-Proof Bottles? (Does it Matter?). 2020. [online] Available: DrugAbuse, https://drugabuse.com/whats-new-in-the-world-of-tamper-proof-bottles-does-it-matter. [Accessed: Oct. 1, 2020].

[3] A. Mizan, *Pillsafe*. 2020. [online] Available: HEALTH MAKERLAB, https://healthmakerlab.medicine.illinois.edu/video_submissions/5e135a286e3da/5e135a286e3da_input_1.mp4. [Accessed: Sep. 14, 2020].

[4] Hackaday.io. *Project | Pill Minder - Automatic Pill Dispenser | Hackaday.Io.* 2020. [online] Available: https://hackaday.io/project/4900/logs. [Accessed: Oct. 17, 2020].

[5] M. Silva, *Introduction to Microcontrollers - Interrupts*. Sep. 18, 2013. [online] Available: https://www.embeddedrelated.com/showarticle/469.php. [Accessed: Oct. 1, 2020].

[6] E. Wong, *What is the average latency of a WiFi network*. Aug. 5, 2019. [online] Available: Quora, https://www.quora.com/What-is-the-average-latency-of-a-WiFi-network. [Accessed: Oct. 1, 2020].

[7] IEEE, *IEEE Code Of Ethics*. June, 2020. [online] Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: Sep. 14, 2020].