Med-I-Can

ECE 445 Design Document

Eric Shen (eshen5), Kyle Hand (kjhand2), and Alyssa Dias (amdias2)

Group 26

TA: Dongwei Shi

10/4/2018

1. Introduction

1.1 Objective

A common issue among the disabled and elderly communities, is the need for taking a wide variety of medications at least once a day. Some individuals rely on their caregivers to assist them in this task, and in situations where multiple people are depending on much fewer caregivers (i.e.: group homes), this becomes a tedious task, and can be detrimental if not completed correctly.

Our goal is to create a solution that will simplify the task for caregivers, allowing patients to take a step closer to independence, reduce the risk of incorrect administration of medication, and give family and caregivers more time to interact with their patients. Our idea is to create a physical system (similar concept to a vending machine) that can serve an individual, but is scalable to multiple people, and is controlled by an app. Our idea sets itself apart by allowing one caregiver to manage medication for multiple users, to create preset dosages/timing for each individual, and modify based on the well-being of each individual (for example, if an individual has a cold, add that preset of specific cold medication to the regular medication routine). This is especially crucial in settings such as Assisted Living spaces.

1.2 Background

A partial solution to this problem exists in giving timed dosages but does not allow for a caregiver to adjust medicine or dosages and also requires all of the pills to be sorted by the caregiver manually. Still other solutions have no timers at all and rely on the patient remembering to take their medication at the appropriate time or require assistance from the caregiver.

Our solution seeks to allow the caregiver to simply deposit pills into our dispenser in predefined containers, and then program the patient's pill schedule into the app. In order to succeed with our project, we must be able to provide a platform that can accurately and dependably dispense a patient's medication according to the caregiver's desired schedule.

1.3 High-Level Requirements List

- The device must be able to be programmed to programmed by a user with a schedule for a patient
- The device must be able to dispense the correct medication according to a schedule 100% of the time
- The device must not dispense medication if the previously dispensed medication has not been collected

2. Design

2.1 Block Diagram

Below is the block diagram for our solution; we've broken it into three sections: Mechanical, Power/Processing, and the App. These can be created independently, and then modified to work in conjunction with each other. This design will satisfy the high-level requirements; it incorporates systems that are scalable from an individual user to multiple, it allows for simple control/caregiver input via the app, and has safety features to protect users/patients (i.e. locking card until empty cup, notifications to app (caregiver) if medicines are not retrieved, etc.).

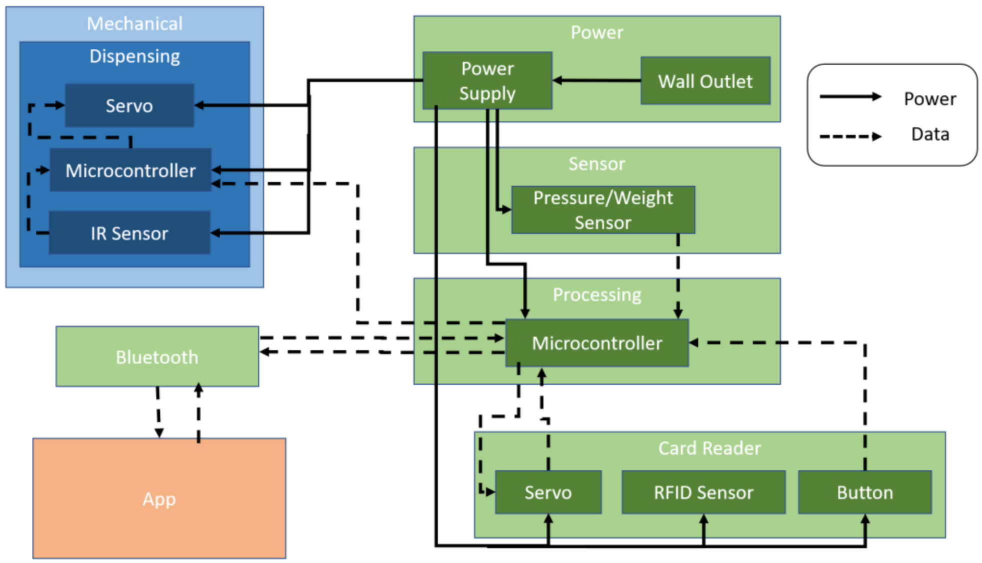

Figure 1-1: Block Diagram

2.2 Physical Design

The overall system is depicted below in Figure 2-1. The reservoirs will sit in the top (please see further for reservoir design), the funnel below, and the cup below it. On the bottom right is the card reader/slot.

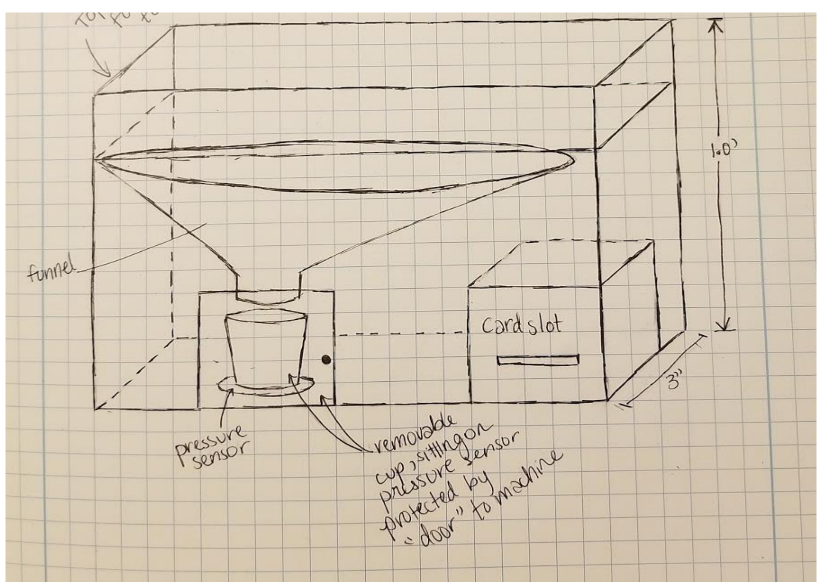

Figure 2-1: Overall Physical System

Figure 2-2 shows the reservoir design for the dispenser portion. One of the biggest concerns with our design is ensuring only 1 pill is dispensed from each necessary reservoir at a time. In the following figures and explanations, we will go into more detail about our reservoir design, and show how we plan on preventing this potentially detrimental concern.

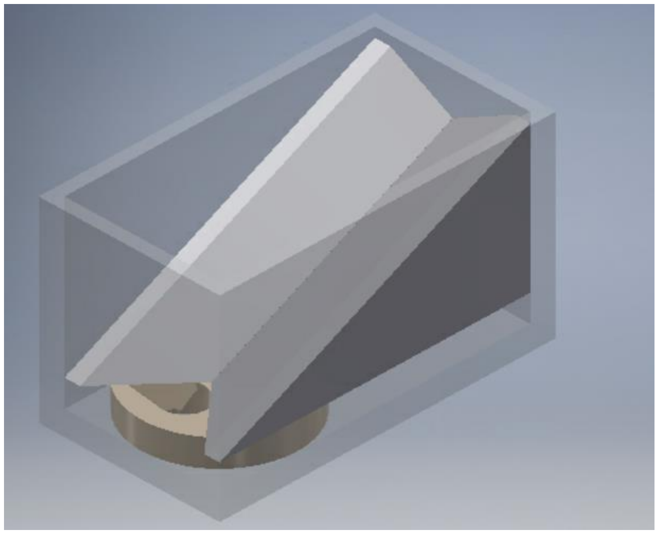

Figure 2-2: Isometric view of reservoir

Figure 2-3 is the wheel design. The slot has the diameter of .4094", and has the depth of .02756". These are the dimensions of an M&M (which we are using as our uniform sized "pills"). This means only one pill can fit in the slot at one time. There is a ramped part in the slot to help pills that are misaligned, and helps orient them correctly.

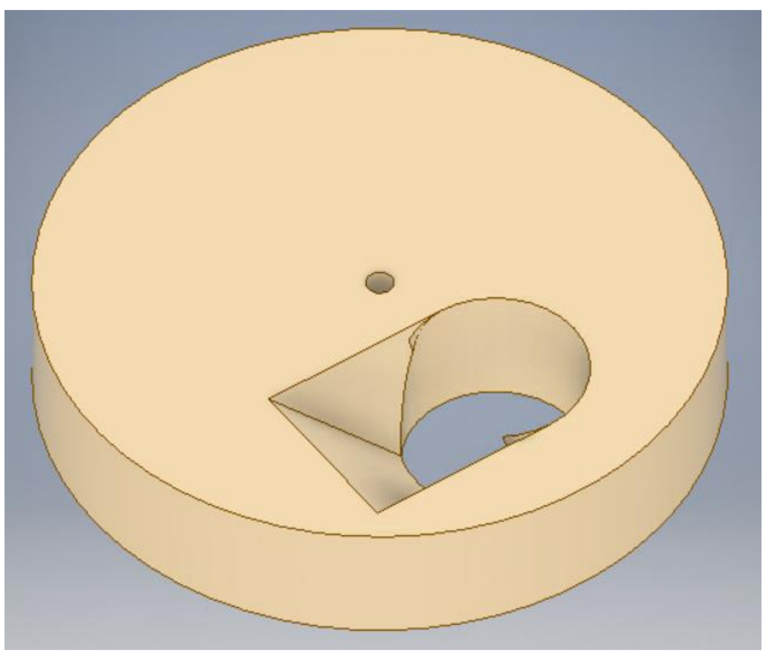

Figure 2-3: Slotted Wheel

Below is the empty reservoir (Figure 2-4 shows isometric view, Figure 2-5 shows top view, looking into the reservoir). When looking at the Figure 2-5, you can see the wheel sits in the bottom area. A pin connecting the wheel and servo, runs through the smallest hole in the reservoir so the wheel can rotate on that axis. Of the two bigger holes, each serves a different purpose. The hole on the right will have the fixed IR sensor (connected to the ramp piece shown later), and a transparent material covering it. When the servo rotates so the wheel slot is aligned, we'll take a measurement. If the sensor only sees the transparent material (no pill), the wheel will reverse to attempt to retrieve another pill and return to this position. If, when the wheel slot is aligned with this rightmost hole, a pill is detected, the wheel will continue to rotate counterclockwise to the topmost hole, where, once aligned, the pill will drop into the funnel below.

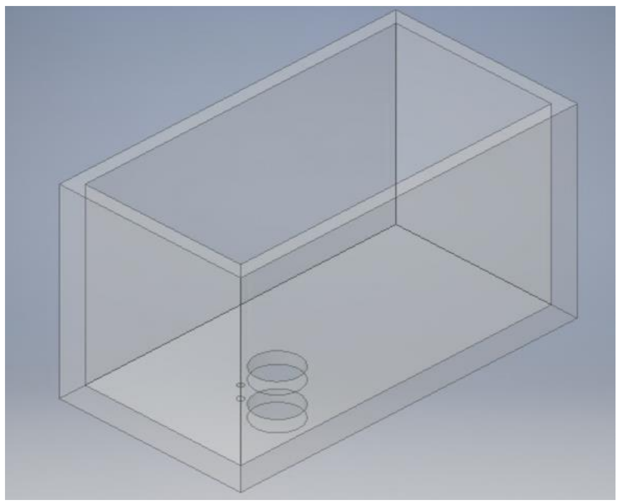

Figure 2-4: Isometric view of Reservoir

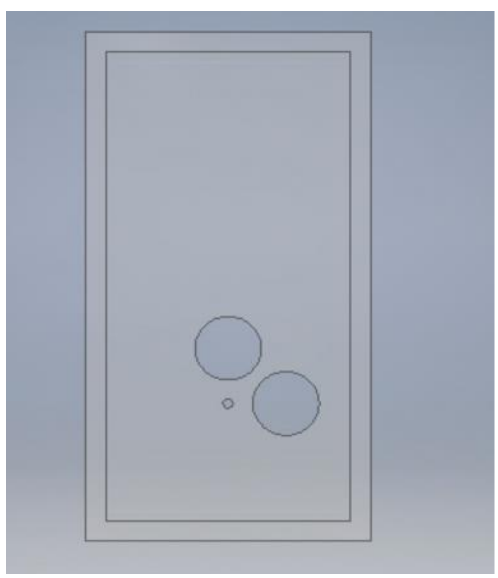

Figure 2-5: Top view of Reservoir

Below is the ramp insert for the reservoir. It has a V-shape to help direct the pills to the center, where the wheel slot will be aligned when picking up a pill.

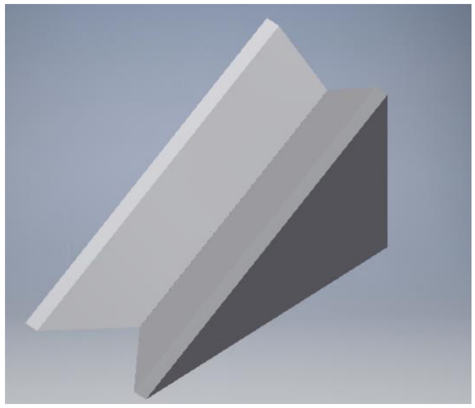

Figure 2-6: Ramp insert

2.3 Block Design

1. Power Supply Unit:
The device will be powered through a wall outlet. The majority of the power consumption will be through the micro controllers and the bluetooth module

1.1 AC/DC Converter

Requirement	Verification
The converter must be able to convert 110v from the wall outlet to a voltage of 5±.5V	Plug the converter into a wall outlet, then measure the voltage between the V+ and ground outputs with a multimeter to ensure that the output voltage is within the acceptable range.

2. Control Unit
The control unit consists of a microcontroller which communicates with the microcontroller in the dispensing unit, the card reader, the pressure sensor and the app via bluetooth.

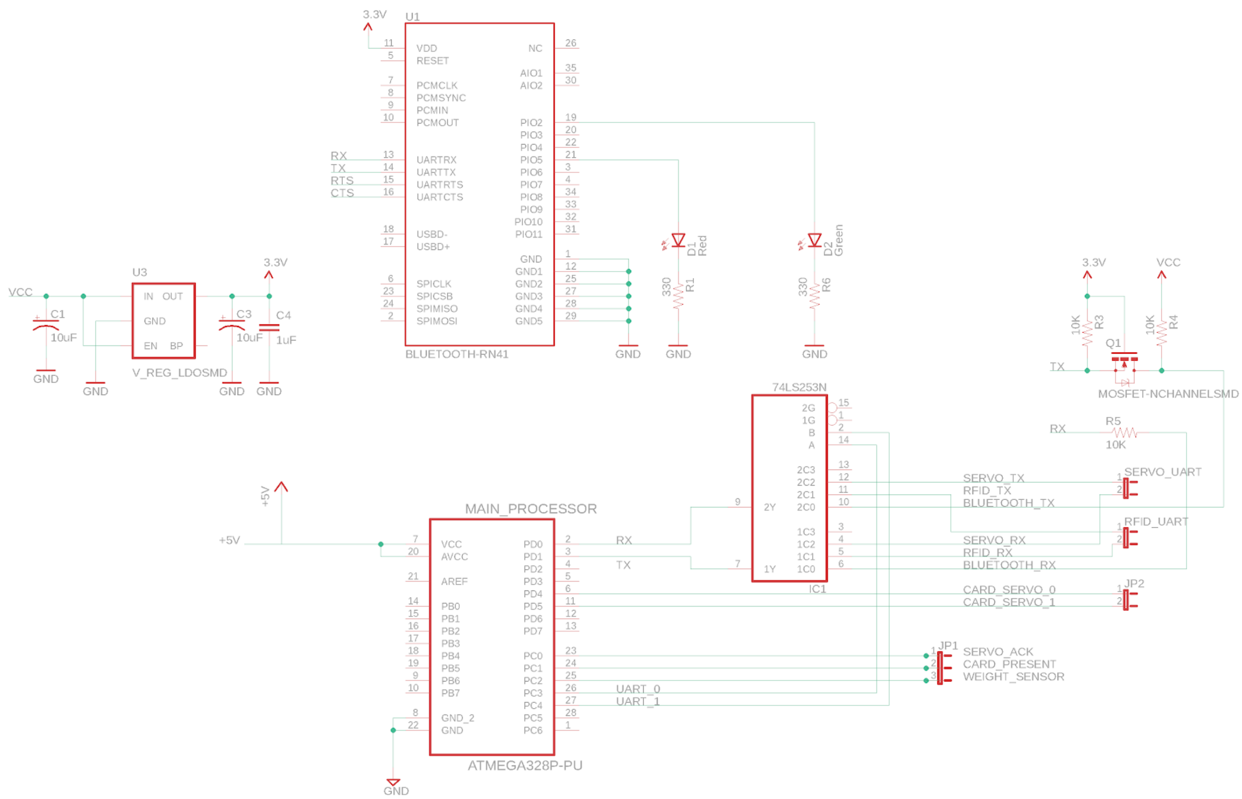

Figure 2-7

2.1 Microcontroller

The microcontroller receives information about the user's pill schedule via the app and will store that information in flash memory. The controller communicates with the Card Reader to recognize the user's card. The information about that user is then sent to the appropriate dispensing unit. After the dispensing unit returns that the medication has been dispensed. The controller must monitor the pressure sensor to ensure that the medication has been removed and the cup returned before telling the card reader that the user's card can be returned. If the user does not collect their dosage within an hour of the scheduled time, an alert will be sent to the app.

Requirement	**Verification**
1. The controller must be able to communicate through UART with 2 devices	1. Another microcontroller will be set to toggle an LED when it receives a UART signal from our main controller. It should successfully toggle 9/10 times. This should be tested with 2 different UART port pairs.
2. The controller must be able to read the outputs from the pressure sensor.	2. A blind test will be run 10 times. An observer should be able to differentiate between cup only, cup + meds, and no cup 9/10 times.
3. The controller must be able to output a signal to control servos on the card reader.	3. A multimeter will be used to test the output of the controller for both +5v ± .5v and -5v ± .5v

4. The controller must be able to recognize when a card has been presented to the device.	4. A blind test will be run 10 times. An observer should be able to differentiate between a card and no card 10/10 times.
5. The controller must be able to read an acknowledgement from the dispensing unit.	5. A blind test will be run 10 times. An observer should be able to differentiate between an acknowledgement and no acknowledgement 10/10 times.

2.2 Bluetooth Module
The bluetooth module is the serial communication replacement between the caregiver's app and the microcontroller for scheduling the medications. Latency should not be an issue for our device as there are no time-based conditions.

Requirement	Verification
1. The microcontroller must be able to communicate with a computer that is running our app through bluetooth.	The controller will be set to toggle an LED when it receives a bluetooth command from the app. It should successfully toggle 9/10 times.

3. Sensors

3.1 Pressure sensor
A pressure sensor is required to ensure that user takes out their medication and returns the cup to the device.
Requirements: The pressure sensory must be sensitive enough to verify that the returned cup is empty i.e. must be sensitive to a degree under the weight of a single pill.

Requirement	Verification
1. The pressure sensor must be sensitive enough to a degree under the weight of one pill(estimated to be 1 gram)	1. A random number of pills between 0 and 10 are to be placed in a cup on top of the sensor and the number of pills in the cup must be able to be correctly determined from the output of the sensor 10 out of 10 times

4. Card Reader
The card reader should detect when a user has inserted their card, fully pull the card into the device and recognize the user based on the card.
4.1 2x Servos:
The servos pull the card into the device when prompted by the processing unit.

Requirement	Verification
1.The servo must be able to be fitted with rubber wheels for traction against the ID card.	Place a card between the rubber wheels and make the servos spin. The test is passed if the wheels grip onto the card and pull the card in
2.The servo must be able to fully pull the ID card into the device	When the card reader pulls in the card in, the entire card must be within the enclosure

4.2 RFID Card scanner:
The scanner recognizes the user with their card

Requirement	Verification
1. The scanner cannot make any mistakes when identifying different users	1. Two different tags will be scanned on rotation, the scanner must be able to correctly identify the tag 10 out of 10 times

4.3 Button
The button should be able to sense when the user has provided a card to the card reader.

5. Dispenser

The dispenser houses several medications and must be able to receive instructions from the control unit and dispense the appropriate medication in the correct dosage

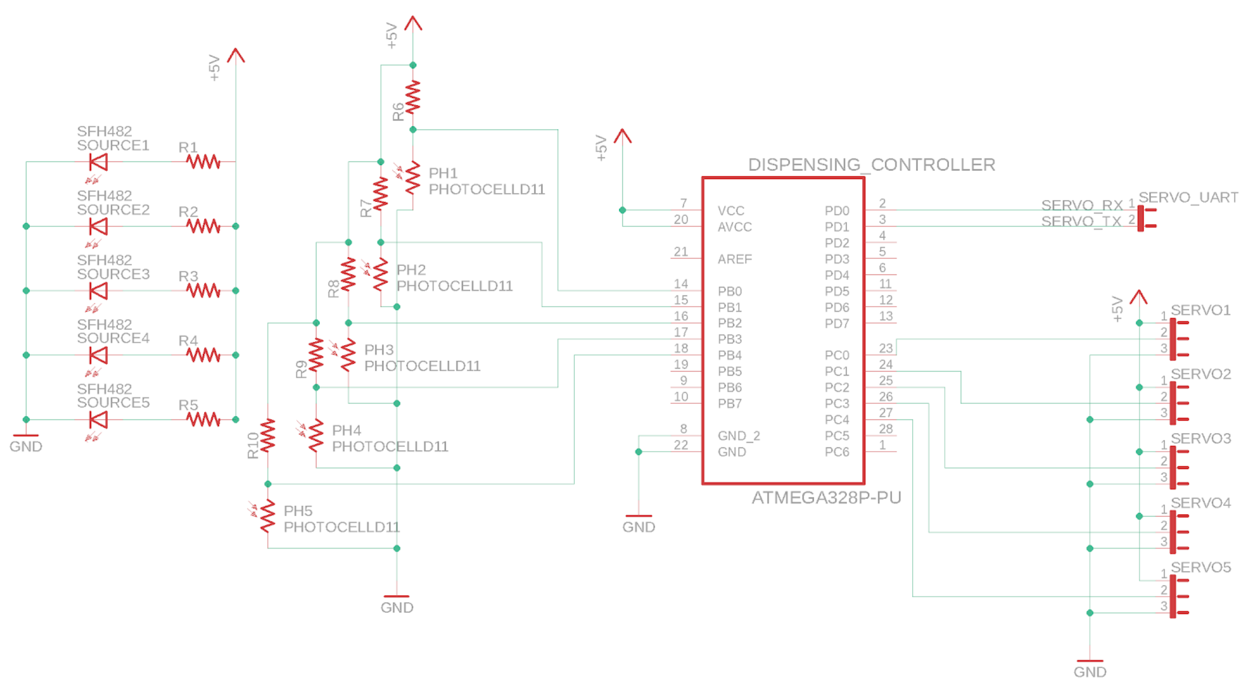

Figure 2-8

5.1 5x Servos

Each of the servos are used to spin a piece that is designed for the specific medication. The piece has a divot in which only one pill fits in, the piece then rotates under a panel to ensure that only one pill is selected. The pill then drops down into the cup

Requirement	Verification
1. Each servo must be able to rotate between 0-155° without getting jammed 2. Servos must be able to accurately line up with three positions: Loading, Check, Drop/Dispense.	1. Servo will be at "neutral"/loading, and an M&M will be placed into the wheel slot. Servo should turn clockwise and counterclockwise without getting jammed. Should correctly do this in 10 out of 10 trials. 2. When servo stops at each destination, it should align such that the M&M is fully visible from below Check area and drops without jamming in Drop area. Should correctly do this in 10 out of 10 trials.

5.2 IR Sensor

The sensor is used to detect when a pill is present in Check state

Requirement	Verification
1. Sensor should detect/differentiate between pill (any M&M, regardless of color) and no pill.	1. A blind test will be done, with one group member watching sensor output, and the other either loading pill into slot, or leaving it empty. When an M&M is placed in wheel slot in loading and twisted to Check state sensor will recognize pill. When wheel slot is empty and twisted to Check state, sensor will return "no pill" output. In 15 trials, Sensor will accurately determine pill vs. no pill with 100% accuracy.

5.3 Microcontroller

The controller must be able to communicate with the controller in the control unit and interpret the instructions to determine which medication needs to be dispensed. The controller then causes the corresponding servo to spin until a pill is dropped.

Requirement	Verification

1. The microcontroller must be able to communicate with another microcontroller through UART	1. Another microcontroller will be set to toggle an LED when it receives a UART signal from our main controller. It should successfully toggle 9/10 times. This should be tested with 2 different UART port pairs.
2. The microcontroller must be able to read the output from the IR sensor.	2. A blind test will be run 10 times. An observer should be able to differentiate when a M&M is within the slot on the disk 10 out of 10 times
3. The microcontroller must be able to control the appropriate servo when needed	3. The controller must cause the servo with the correct address to spin 10 out of 10 times

2.5 Software

Our software consists of the control logic run by the microprocessors in the Control Unit and the Dispensing Unit and an app.

2.5.1 Control Unit

The Control Unit takes in inputs from the Card Reader, the Dispensing Unit and the Pressure sensor, the Control Unit has outputs to the Dispensing Unit and the Card Reader. The Control Unit first reads receives the patient information from the Card Reader. A variable called count is initialized to the number of medications the user needs. The first medication is then sent to the Dispensing Unit along with the dosage of that medication. The Control Unit waits until the Dispensing Unit returns that it has dispensed the medication. The Control Unit then sends the next medication until the patient has the entirety of their prescription dispensed. The Control Unit then makes sure that the patient has taken their medication and returned the cup to the device before releasing the patient's card.

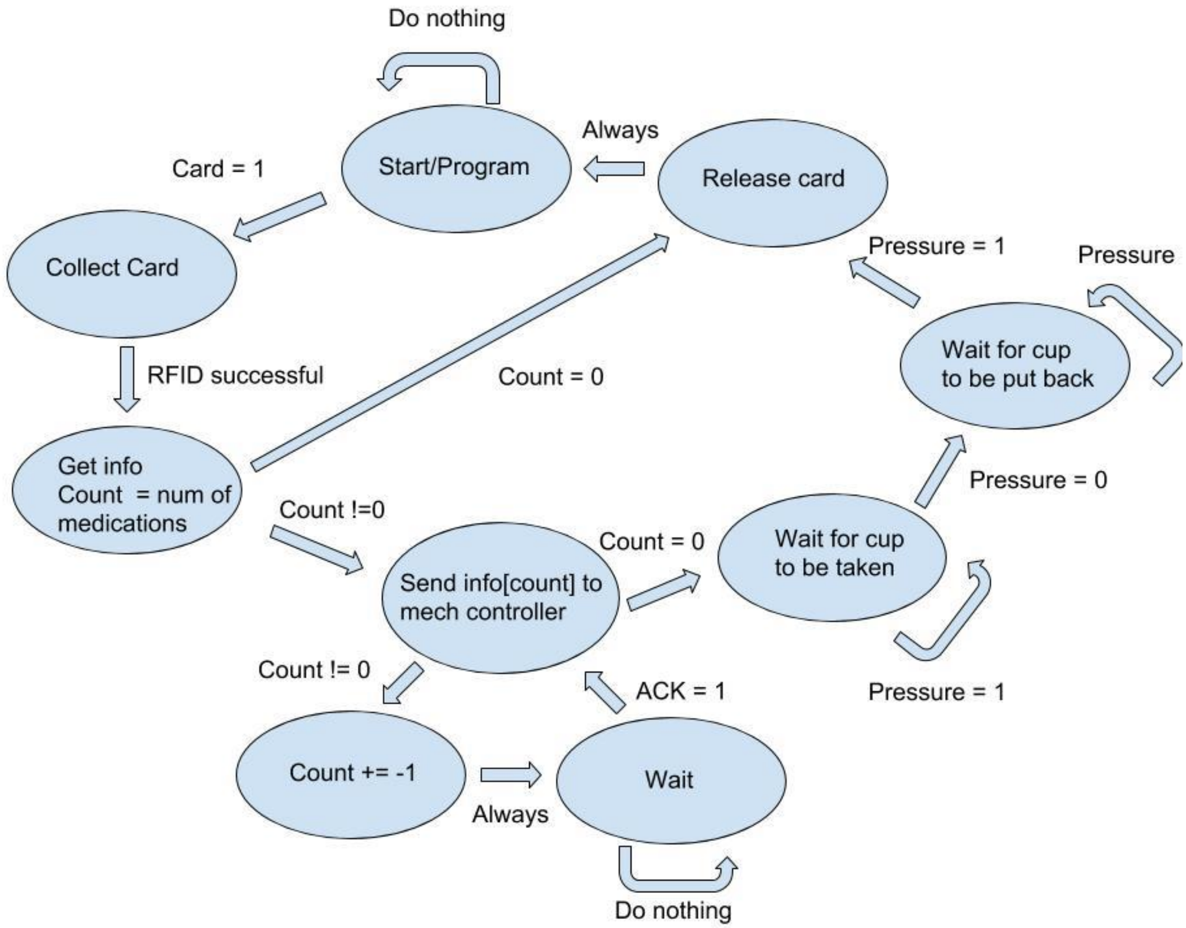

Figure 2-9 Control Unit State Diagram

2.5.2 Dispensing Unit

The Dispensing Unit takes in inputs from the Control Unit and the IR sensor, and sends outputs to the Servos and the Control Unit. The Dispensing Unit first receives the address of a medication along with the dosage from the Control Unit. The Dispensing Unit then initializes a variable named Count to the required dosage. The IR sensor is then started and the servo is spun until a pill is detected within the disk on the servo by the IR sensor. The pill is then dispensed and this process is repeated until the proper amount of medication has been dispensed. The Dispensing Unit then sends a Done signal back to the Control Unit indicating that all the medication has been dispensed.

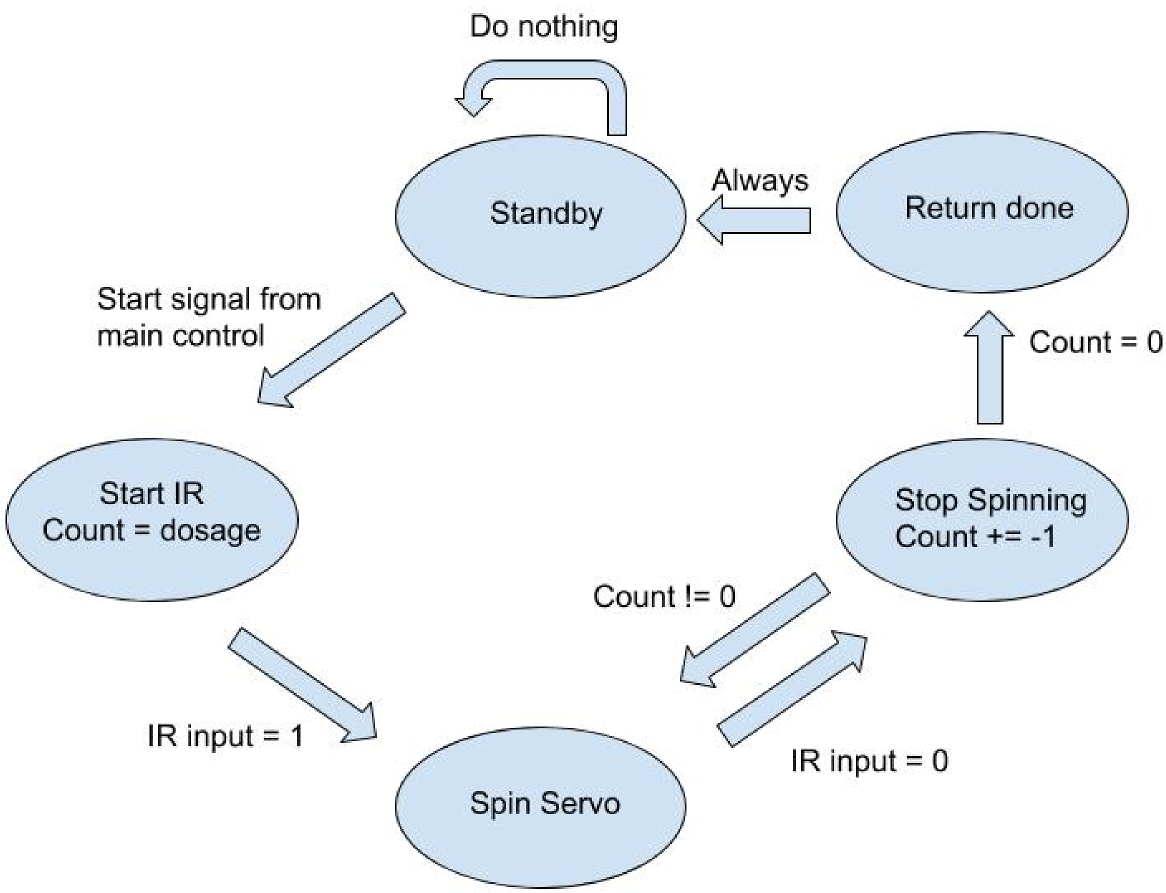

Figure 2-10 Dispensing Unit State Diagram

2.5.3 App

The app will be used by the caretakers in conjunction with the device to help care for the patient. It will have information such as which medication is in which location as well as the complete schedule for each patient. When a dosage is missed, it will also alert the caretaker that the patient missed a dose.

Requirement	**Verification**
1.The app must allow for the caretaker to make changes to doses and medications	1. A new dosage will be scheduled on the app and then sent to the device. The device should dispense this new dosage correctly 10/10 times.
2. Check some of the history for previously dispensed medication	2. A dosage will be dispensed and then the “history” will be brought up. The history should reflect the dispensed dosage

3.See if a certain patient has gotten their medication yet	3. A dose will be scheduled to dispense and the "current activity" page will be brought up. The dosage should show as "scheduled" until dispensed where it will change to "dispensed."

2.6 Tolerance Analysis

One of the most important tolerances that we will have to deal with will be in the dispensing of the medicine. It is crucial for our device to be able to accurately and reliably pick up a single pill. One piece of the puzzle will be to ensure that our medicine pickup mechanism will only accept one pill at a time. The dimensions of an M&M, which we will be using as a substitute for pills during our design and execution phases, are .4094" in diameter and .02756" deep (3). These are key measurements that we will need for designing our pickup and dispensing system.

Since we only want 1 pill at a time, we need to ensure that the depth of the hole for picking up the pills is only deep enough for one pill to fit, but not so shallow that the pill could be easily knocked out. We have found that a depth of .03" sufficiently fits that requirement.

The next measurement we need to ensure is accurate is our diameter. Similar to the depth it must be easy to pick up the pill but not so large that multiple pills could fit. This calculation gets slightly trickier because the pills could attempt to feed into the slot vertically, causing a jam. In order to calculate this we need to get the distance between two edges of an M&M. The distance between the M&Ms is:

$$2 * (\text{depth} / 2) = \text{distance between 2 M\&M edges}$$

So we can see that the depth of the pill would be needed to be taken into account. In our case it is a simple matter of making sure that the diameter of the circle is small enough that a box that has edges:

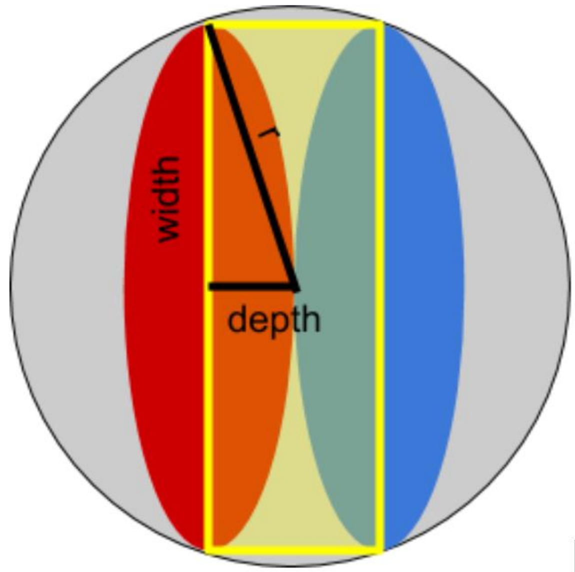

Figure 2-11 M&M Diagram

M&M diameter X M&M depth could not fit into the hole. This is a simple calculation to do since we already know the depth and diameter of an M&M. The diameter of our hole cannot be larger than:

Diameter = sqrt((½ * depth)^2 + (½ * diameter)^2)

In our case this calculates to a radius of 0.2052" or a diameter of 0.4104" or approximately 1/50th of an inch wider than the width of an M&M. The probability of 2 M&Ms getting jammed in this configuration is very small but by keeping the diameter of the hole to specifically this size, we can completely eliminate the possibility.

In addition to keeping the diameter of the hole small, we will also be orienting the pills before they drop into the hole. This requires a calculation into the tipping point critical angle of an M&M if we want to keep it on its flat side in order to minimize further the situation from above. Luckily, M&Ms are fairly pointed disks with hardly any edge so any angle that is not exactly 90° perfectly vertical will fall back onto its flat side.

Cost and Schedule

Cost Analysis

Labor: Assuming a wage of $35 per hour, and assuming each group member puts in 12 hours a week for 10 weeks. The final total labor costs is estimated to be:

3 people *$35hour*12 hours per week*10 weeks *2.5 =$31,500

Description	Part Number	Cost (Ea)	Quantity	Total Cost
RFID Module	SKU 113990014	$12.50	1	$12.50
Bluetooth Module RN-42	WRL-12574	$18.95	1	$18.95
ATMEGA328P-PU	COM-09061	$13.99	2	$27.98
Load Cell 100g	SEN-14727	$8.95	1	$8.95
3.3 Voltage Regulator	COM-00526	$1.95	1	$1.95
dual 4 to 1 mux	SN74F253N	$0.79	1	$0.79
110v to 5v5A converter	B015C5HVOA	$10.99	1	$10.99
IR Sensors	SEN-00241	$1.95	5	$9.75
Servo	ROB-11884	$9.95	7	$69.65
Assorted resistor, capacitor, ICs etc.		$10.00	1	$10.00
			Total	$171.51

Our Grand Total cost for the project will be $31,671.50

Schedule

Week	***Eric***	***Alyssa***	***Kyle***
10/8-10/14	***Finalize control and power parts lists and order***	***Finalize mechanical parts list and order***	***Begin Programming Control Module (framework)***
10/15-10/21	***Begin programming mechanical control (framework)***	***Design PCB for control module***	***Continue Programming Control Module***
10/22-10/28	***Unit Test mechanical and power parts and finalize mechanical control***	***Finalize and order first PCB order***	***Unit Test control parts and finalize control software***
10/29-11/4	***Begin App build***	***Assemble mockup of mechanical and power modules for testing***	***Assemble mockup of control module for testing***

11/5-11/11	*App debugging and integration with control module*	*Begin PCB work for final integration*	*Begin initial integration of control module*
11/12-11/18	*Finish integrating app*	*Build physical enclosure and hardware component installation*	*Finish integrating control module*
11/19-11/25 *Fall Break*	*Tie-up final tasks before Mock Demo* *Work on Final Paper*	*Begin planning for mock demo*	*Finish final integration*
11/26-12/2	*Mock Demo* *Prepare for Demo – make changes based on feedback*	*Mock Demo* *Prepare for Demo – make changes based on feedback*	*Mock Demo* *Prepare for Demo – make changes based on feedback*
12/3-12/9	*Demo* *Mock Presentation* *Prepare for Presentation*	*Demo* *Mock Presentation* *Prepare for Presentation*	*Demo* *Mock Presentation* *Prepare for Presentation*
12/10-12/16	*Presentation* *Complete/Submit Final paper*	*Presentation* *Complete/Submit Final paper*	*Presentation* *Complete/Submit Final paper*

Discussion of Ethics and Safety

In dealing with devices that cross into the healthcare domain, there are many things to be conscious about including information privacy, malicious use of the system, and medical regulations.

Since we deal with sensitive information such as an individual's medications and medication schedule, we have to worry about potentially leaking this information, intentionally or unintentionally. This makes us susceptible to possible breaches of Association for Computing Machinery (ACM) codes 1.6 and 1.7 (1), which require respecting privacy and honoring confidentiality. We need to make sure that we are securing the user's data correctly and making sure that we do not access information needlessly.

The best solution to dealing with this information is to encrypt it within the app subsystem and only store critical information on the actual device. Information on the device will be restricted to the timings and the dispensers while the app can have information such as the patient's name, which medications are set to dispense, and when they are set. Since that information is sensitive and private only to the caretaker and the individual, we will set a login for accessing the information on the app. Only after a successful login will the caretaker or individual be able to access scheduling, medicine, and settings for the device. Hosting all of the information on the app will ensure that developers cannot see the private information and the login protects against a malicious user attempting to access that private information.

When working with medicine and peoples' medicine schedule, we do have to take into account the users' health. The Institute of Electronics and Electrical Engineers (IEEE) Code of Ethics also takes this into account with rule number 9 (2). In terms of our project, we must ensure that pills are dispensed correctly every single time without any missed doses or double doses. We plan to guarantee this by placing a sensor to check when each pill is dispensed and will keep spinning the servo until that pill is dispensed. On the other hand, we will create a compartment in the disc that the servo spins that will hold exactly 1 pill. With both the mechanical and sensor design, we can ensure that the correct number of pills is dispensed every time.

Due to the many regulations related to group homes and nursing homes, this device will most likely be targeted towards individuals, their families, and caregivers. Because families and individuals aren't governed by any one group, there isn't a specific list of qualifications to pass for medicine dispensing devices. Primarily, we will have to show these groups that the device is consistent, dependable, and easy to use. This will be taken into account during the entire design process and tested through interviews with potential users of our device.

There are several other safety concerns that we will have to address. The first would be that of a malicious user attempting to access the individual's medication. While this would already be a problem for someone using the device, we can make it even more secure by locking up the medication and only dispensing it when the user presents their unique identification card. In addition, we could also prevent a case in which the user forgets to take their medication on time. To combat that, we are going to have alerts sent to the caregiver when the individual has missed a dosage at a certain time. This can help the caregiver keep track of how often the user is missing doses and if more intervention might be necessary.

Since this device is connected to the wall outlet, there is some danger of electrocution when developing or using the device. In order to mitigate this chance, we will insulate all wires and cords both within the device and going to the wall outlet as well as using the three prong, grounded, outlet plug. In addition, any developers working on this device should unplug the device before working with it.

Citations

(1) ACM Code 2018 Task Force. "ACM Code of Ethics and Professional Conduct." Association for Computing Machinery, www.acm.org/code-of-ethics.
(2) "IEEE Code of Ethics." IEEE - Advancing Technology for Humanity, www.ieee.org/about/corporate/governance/p7-8.html.
(3) Hallock, Betty. "M&M's Mega -- with Three Times as Much Chocolate -- Hits Stores." Los Angeles Times, Los Angeles Times, 25 Apr. 2014, www.latimes.com/food/dailydish/la-dd-mms-mega-three-times-chocolate-hits-stores-20140411-story.html.
(4) Science Buddies Staff. "M&M Geometry." Science Buddies, 28 July 2017, https://www.sciencebuddies.org/science-fair-projects/project-ideas/Math_p022/pure-mathematics/mms-geometry. Accessed 4 Oct. 2018.
(5) Hallock, Betty. "M&M's Mega -- with Three Times as Much Chocolate -- Hits Stores." Los Angeles Times, Los Angeles Times, 25 Apr. 2014, www.latimes.com/food/dailydish/la-dd-mms-mega-three-times-chocolate-hits-stores-20140411-story.html.