Medicine Dispenser Final Documentation

Bryce Dickerson
Aaliyah Duffy
Kara Garvey
Sydney Oskin
Eric Zheng

Senior Design EE 41430

Michael Schafer

May 8, 2024

1 Introduction

Most people in the United States have themselves, or know someone who has, taken prescription medications. If we consider those who take over the counter medications or supplements then that covers even more of the population. According to the Health Policy Institute at Georgetown University in 2019, over 131 million people or 66% of the adult population in the United States use prescription drugs[1]. In 2022, the total number of prescriptions went up to 6.7 billion in the United states[2]. Previously, in 2018 it was at 6.1 billion prescriptions[2]. These statistics show the prevalence of prescriptions in the United States. Each of these prescriptions, and prescription users, has their own story and ailment that has caused them to seek treatment. This varies from conditions as common and treatable as a sinus infection, or as complex and difficult as cancer or parkinsons. According to the Center for Disease Control and Prevention (CDC), 60% of Americans live with a chronic health condition[3]. Not every chronic condition requires the intervention of prescription drugs and not everyone who has a chronic condition has access to affordable health care. That being said, by comparing that 60% of Americans with chronic conditions to the 66% of Americans who use prescription drugs, it is clear that most prescriptions are for chronic conditions.

Chronic conditions include heart disease, cancer, diabetes, and many others. Notably, Alzheimers, other dementias, and ADHD are considered chronic conditions. When a patient is prescribed medication it is important for them to take it as the doctor advised them to. The adherence rate is the term used to describe the percentage of doses taken as directed. Though individual studies vary widely, through extensive literature review it has been determined that adherence is approximately 50% in developed countries[4]. Patients who do not adhere to their medication directions are more likely to have a lower quality of life, greater disease progression, increased hospital visits, and several other consequences. Non-adherence has a few different types, failure to have a medication filled, stopping a medication without consulting the medical professional, or non conforming which can be skipping doses, or taking a medication at the incorrect dosage or incorrect time. This last one, non-conforming, is the one we are targeting.

Of all the patients who have chronic conditions there are many who struggle with conforming to their medication instructions. There are two main sub groups we are aiming to assist. Patients of any age (18 and above) who have ADHD or similar struggles, and older people who might have dementia or memory problems. These subgroups of patients are two of the highest when it comes to medication non-conformity. In the case of the ADHD patients, out of sight out of mind is an extremely common theme throughout their daily lives. For older people who have dementia or early dementia symptoms, they may forget that they have medications to take, what time to take them, which ones to take, or in more severe cases they may even forget who or where they are. At any level of dementia, it can be very hard for them to comply properly with their medication directions. Often a caretaker is implemented to help them take their medications properly. This is usually a family member or someone close to them. The role of caretaker for an elderly patient is often an extremely taxing role as there are many aspects of life which they need assistance with. We have come up with a product which can help ease some of that burden off of them and help the patient maintain more independence.

Our solution is to create a system to dispense the medications and remind the patient to take them. The system will have three main components, a website, the medication dispenser, and an auxiliary reminder system. The website will require them to create a user and input the relevant prescription information which can be found on the prescription bottle. Once that is completed the user or caretaker will fill the dispenser with the proper medications as they would for any weekly pill organizer. When it is time to take the medication, the dispenser will rotate the lid to reveal the current medications to be taken. At this point there is a screen on the dispenser which will display the medications being taken as well as any directions that go with the medication (ex. Eat or drink with this medication). At the same time the system will send the reminder to the auxiliary system, which can be kept in any room of the home, and it will play audio stating “time for your medication” as well as lighting up an LED indicator as well. A reminder is also sent through the website to the patient's phone. The purpose of these multiple reminders is to help ensure that the patient takes the medication promptly as well as to cater towards the different patient needs. An elderly patient may have auditory or visual issues and need one or the other type of reminder whereas a young patient with ADHD would likely prefer the phone reminders in addition to the in-home reminders. When the patient takes the medication there is a button on the dispenser to press which indicates they have taken the medication and the reminders will stop. If the prescriptions change at any point, the user or caretaker would simply need to log back into the website to alter the medication list.

Our system met our expectations. It dispensed the medication on time, displayed the information from the website, and sent the reminder to the auxiliary system. Some things could use some improvement though. The individual medication slots could be a bit larger to accommodate more medications and for easier retrieval of the medication. The timing for how long the motor runs to display the next doses could have been a bit more precise though it was satisfactory for a first demonstration of the system. There were some issues with the main systems' board connecting reliably to the computer to receive code which we theorized was due to the USB-C connector not being designed from repeated plugging and unplugging. The website connection to the system had some issues that we were unable to fix in the time that we had. It was not able to send data to the board easily, and we did not have time to implement a function to alter or remove previous prescriptions.

2 Detailed System Requirements

To address the complexities of medication management and ensure the effectiveness of our system, each requirement plays a crucial role in solving the problem of medication non-adherence and enhancing the user experience.

Ensuring constant power is fundamental to the reliability of our system to maintain uninterrupted operation. Any interruption in power could lead to missed medication doses and compromise the health of the user. Thus, the system must have constant power to ensure that no medication dispenses or reminders are missed. This is true for both the main system as well as the auxiliary system.

A user-friendly interface is essential for facilitating easy prescription input and allowing users to override reminders when necessary. The interface should be intuitive, accessible, and inclusive, catering to users with varying levels of technological proficiency and physical abilities. Users should be able to easily input their medication details such as brand, dosage, and any special instructions like whether the medication should be taken with food. Additionally, the interface should allow for overrides to stop reminders and for unforeseen circumstances. This capability empowers users to maintain control over their medication management process, enhancing their confidence in the system's reliability.

To promote consistent medication adherence, our system must offer a variety of reminder types that are repetitive and adaptable to diverse user needs. Auditory, visual, and phone/web reminders should be available, ensuring that users receive notifications through their preferred option. Additionally, reminders should be accessible from multiple locations to account for users' mobility and varying proximity to the main device.

The primary function of our system is to dispense medication promptly and accurately according to the user's schedule. This demands precise timekeeping capabilities to ensure that doses are administered at the correct times, thus reducing the risk of incorrect dosage intake.

In addition to reminders, our system should provide users with relevant information about their medication at the time of intake. This may include instructions on whether to take the medication with food or other pertinent details. By displaying this information when needed, the system enhances user awareness and promotes safe and effective medication use.

After medication intake, reminders should promptly cease to avoid unnecessary interruptions. Our system should be equipped to detect when medication has been taken, through manual confirmation. This ensures that reminders are terminated in a timely manner, maintaining user comfort and minimizing potential annoyance.

While not expressly part of our initial requirements, we recognize that flexibility in adjusting basic settings is crucial to accommodate individual preferences and changing needs. That is why that is the primary target of our to-market changes for this product. Users should have the ability to customize reminder settings according to their lifestyle, preferences, and medication regimen. This includes adjusting reminder volume and type (auditory, visual, or both), as well as for easy modification of prescription information, such as dosage adjustments or medication changes, to reflect any updates from healthcare providers. We will discuss this idea further in the To-Market Changes section.

By meeting these detailed system requirements, our solution will effectively address the challenges associated with medication management, empowering users to adhere to their prescribed regimens with confidence and ease.

3 Detailed project description

3.1 *System theory of operation*

Our Medicine Dispenser Reminder System is an integrated solution designed to ensure patients adhere to their medication schedules. The complexity of managing multiple medications has been simplified into an intuitive system that not only reminds but also dispenses medication accurately. At the heart of the system are several ESP32 microcontrollers responsible for various tasks, with robust ESP-Now networking ensuring seamless communication between each subsystem. These controllers are specialized for their particular role; one for managing the dispensing mechanism, one for the user interface including the LCD, and others placed strategically to serve as nodes in the greater network providing audio and visual prompts.

The operation revolves around a meticulously orchestrated schedule, aligning with the patient’s prescribed medication times. Utilizing the ESP32's internet capabilities, we acquire accurate time data from NTP servers, which ensures that reminders are timely and precise in their execution. Our system is comprehensive in its functionality. Upon the prescribed time, the central unit triggers a rotational motor controlled by precise timing algorithms which presents the correct medication. Concurrently, relevant information about the medication is displayed on an LCD screen. This information is fetched from our cloud-based database, Back4App, where it is initially entered via a web interface into a secure and organized structure. Simultaneously, the system prompts the auxiliary subsystems to alert the patient through customizable audio cues and visual indicators such as LEDs. Should the reminder go unnoticed, the system is designed to repeat reminders and can be customized to increase frequency or escalate the alert methods to ensure the patient's attention is caught.

Once the patient acknowledges the reminder and takes their medication, a simple button press is all that is needed to inform the system of the completed action. This interaction is crucial as it serves to disable the reminder sequence and logs the event, providing an important data point for caregivers or healthcare professionals about the patient's adherence. In the event of a system error or malfunction, we have built-in diagnostic tools ready to be deployed, such as error logs and maintenance alerts, enabling remote troubleshooting and prompt resolution of any issues that could otherwise disrupt the dispensing schedule. In essence, from its mechanical components to digital interfaces, the Medicine Dispenser Reminder System is poised to provide a failsafe method of managing medication that molds to the lifestyle and needs of the patient, ensuring adherence, and by extension, improving health outcomes.

3.2 *System Block diagram*

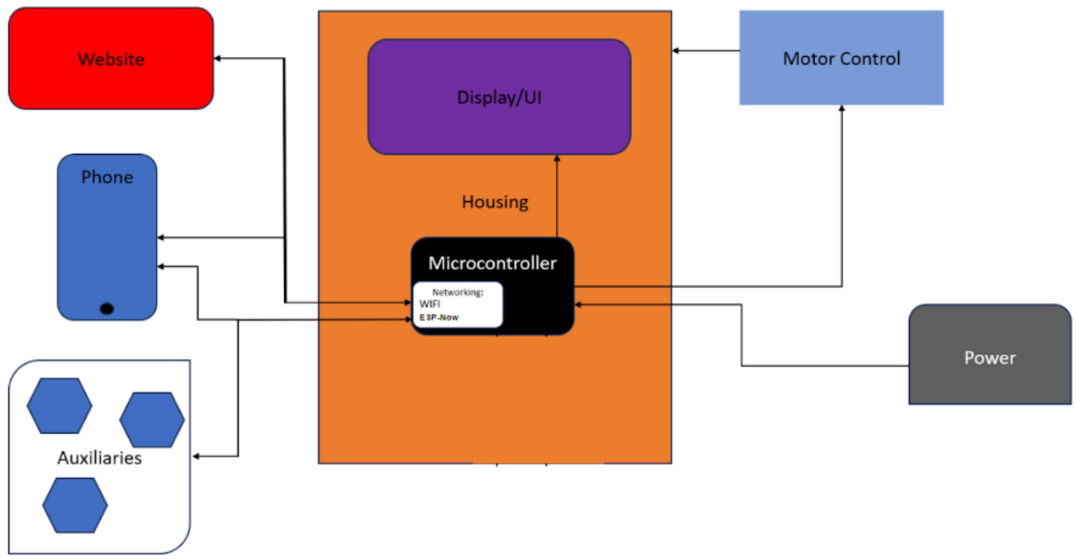


Figure 1 The overall system block diagram.

The above diagram describes the overall system architecture of the medicine dispenser. All of the microcontroller units, main and in auxiliary, are powered through USB-C. The user will first input information such as dosage and time of medication on the website. After submitting the needed info, the website will update the information that will be displayed on the LCD screen and the timing of medicine dispensation through a WIFI connection. As the microcontroller gets the real time through WIFI, the motor, connected to the microcontroller through GPIO pins, will operate when it is time for the user to ingest the medication. The LCD screen, connected to the microcontroller through the SPI protocol, will then display the drug information that was inputted by the user. The main unit will then communicate to the auxiliary units through the ESP-Now protocol, that the medicine is dispensed. The auxiliary system will then output repetitive auditory and visual reminders. The auxiliary microcontroller is connected to the speakers through I2S, and the LED is connected to the GPIO pins. The audio file that is played is read in the auxiliary system through SPI. Finally, the system will also push a notification to users phone, or email as the medicine is dispensed.

3.3 *Detailed operation of Housing subsystem*

The Housing subsystem has two main requirements. It must dispense the medication in an accessible manner and it must house the electronics of the main system. The Housing is 3D printed to meet these requirements. Originally, the design of the housing was to include conveyor belts with medicine bins attached to them which would dump the next dose of medication into a funnel which would send the medication into a tray for the patient. This design seemed like it would have fewer issues than other ideas we brainstormed such as moving a tray of medication bins to allow the medications to fall out through a hole in the bottom. Throughout the process of designing and printing the conveyor belts a few issues were found. The pieces of the conveyor belts with hinges were being printed with varying levels of thickness resulting in many of the hinges being broken off upon receiving them from the print shop. This issue combined with concerns about the weight of the conveyor belts varying which could affect the motor requirements lead to brainstorming other possible Housing ideas.

The design which was chosen after that brainstorming session is the one that was ultimately implemented. The design is a circular container with 15 medication bins, one extra as the home position which would not hold medications, and 14 to be used as a weeks worth of morning and night medication or two weeks worth of once a day medications. The lid of the medication bins has teeth on the inner edge to interface with the gear on the anchored motor so that the lid will rotate to reveal the next dose of medications. This design was chosen for a few reasons. The lid rotating would have a constant weight associated with it so the motor requirements would be consistent throughout the dispensing of each dose of medication and because there are fewer parts that could be misprinted.

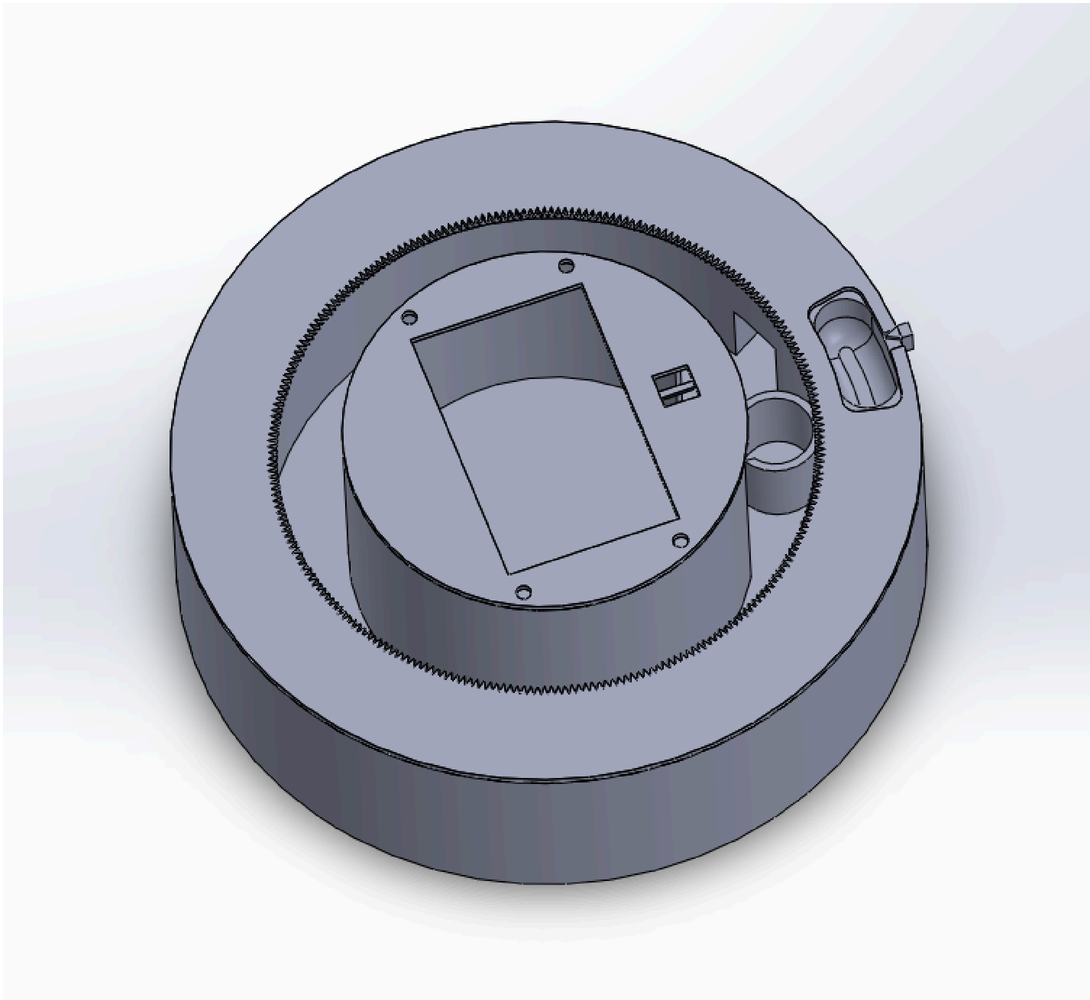


Figure 2 The final Housing design

In Figure 2, the final housing design is shown. The outer ring is where the medicine bins are located. The lid on top of the outer ring is to cover all the other doses of medication so that only the correct dose of medication is accessible to the patient. The lid also has a ring on the bottom which protrudes downwards to ensure that it stays centered on the bins. This ensures that it will not be able to fall off accidentally if the system is moved. This ring on the bottom of the lid fit too tightly to the inner edge of the medicine bins which would have prevented the lid from rotating to reveal the medications. To fix this, sandpaper was used to shave it down as uniformly as possible. Following this fix the lid still fit snuggly but was able to rotate. There is a small gap in the opening of the lid which is intended to help ensure that the lid is centered when it is placed following the distribution of the medications into the correct bins. The height of the bins is to ensure that the lid is at the same height as the gear on the motor. The small circular opening at the bottom of the housing inside the outer ring is to hold the motor. There is a gap in that circle which can barely be seen in figure 2, that is for the wires to motor to be threaded to the main electronics. The size of this ring is as close to the motor size as possible to ensure that the motor is unable to move. If it were to be too loose of a fit then when the motor runs it could turn the entire motor in place instead of turning the lid of the container. There is also a hole under the medications bins to run the power cord through so that it would not interfere with the accessibility of the medications or the ability to remove the lid when refilling the medications.

The inner ring of the Housing is where most of the electronics are located. The ring itself is taller than the outer ring, and there is a sizable gap centered on the motor to ensure that the inner ring does not interfere with the motor or motor gear. The gap was also intended to allow for the power cord to be threaded through the exit hole as well for the wires to the motor to have space. The size of the inner ring was decided based on the dimensions of the LED screen. The lid for the inner ring is to encapsulate the electronics where the patient or caretaker would not mess with them. For the purposes of developing this product the lid is not very secure as to allow our team access to the electronics as we develop them. Similar to the outer ring lid, this lid also has a small protruding ring on the bottom to fit it to the ring on the main print. The screen was originally going to be mounted on pegs from the bottom of the housing and screwed in through the lid. For easier access to the electronics though it was decided that we would only attach the screen to the lid. It was intended that the screen would be screwed into the lid using the screw holes printed in the lid and the screw holes in the LED screen device itself. Upon receiving the print, the holes were not placed properly and did not align with the holes on the screen. Despite this issue, the screen fit into the hole for the screen very well and did not need any additional hardware to be secure, tension was enough. The other hole in the lid is for the button. The button did not have screw holes or anything to secure it with. The bar seen under the hole for the button is to ensure that it would not fall through when it is pressed.

A DC motor was used for rotating the lid. This was decided as it was a cheaper option and would fulfill the requirements of the system. The motor will be turned on and off with a motor driver board from Bojack. More specifically, the motor driver boards will use the L298N to drive the motor at 5V power and 36mA. The motor will only go one direction so one input pin will always be high which allows us to use only three input GPIO pins as shown in the connections document. The power supply to the driver board was split to power both the driver board and the motor from the same voltage. Upon testing the motors ability to rotate the lid, after it was sanded down, it worked very well. There was no stopping or starting in the middle of a rotation, the teeth on the gear and the lid fit together very well, and the power supplied from the main board was sufficient. Through running the program a few times with different time settings for how long the motor should run to ensure that the lid was centered on the next medication bin it was determined that it should run for 925 milliseconds.

Overall this design shown in figure 2 turned out to be functional for all of our requirements of the system and clean. There are a few things that could be improved upon which will be discussed later on in the To-Market Design Changes section.

3.4 Detailed operation of Network subsystem

To enable communication between microcontrollers we opted to use the protocol ESP-Now. ESP-Now is a wireless communication protocol developed by Espressif Systems specifically for the ESP32 and ESP8266 microcontrollers. It facilitates peer-to-peer communication between ESP devices without relying on traditional Wi-Fi network infrastructure. This protocol operates in the 2.4GHz frequency band and is adept at handling low-power scenarios making it ideal for IoT applications, sensor networks, and home automation systems. The key capabilities of ESP-Now stem from its efficient and direct communication approach. It boasts good range, low latency, and minimal overhead, making it suitable for real-time applications. ESP-Now, additionally, supports simultaneous communication between multiple devices, enabling efficient data exchange within a network of ESP devices.

In comparison to other communication protocols such as Wi-Fi and Bluetooth, ESP-Now exhibits several advantages and limitations. Firstly, it facilitates direct device-to-device communication without the need for a central access point, unlike Wi-Fi, thereby simplifying network topology and reducing infrastructure complexity. Secondly, it offers lower latency and higher throughput than Bluetooth, which is beneficial for real-time data transmission. However, ESP-Now operates in the crowded 2.4 GHz frequency band, which may lead to interference issues in environments with numerous wireless devices. The protocol also only has a max payload size of 250 bytes. We ultimately determined ESP-Now was the best choice for our system given its ease of implementation, as well as its capabilities. Given a user is operating the system in the comfort of their home, we are expecting low interference issues, while maintaining good latency and data transmission throughout the building. Also, keeping in mind the size of data we need to send to-and-from devices, 250 bytes is ultimately sufficient.

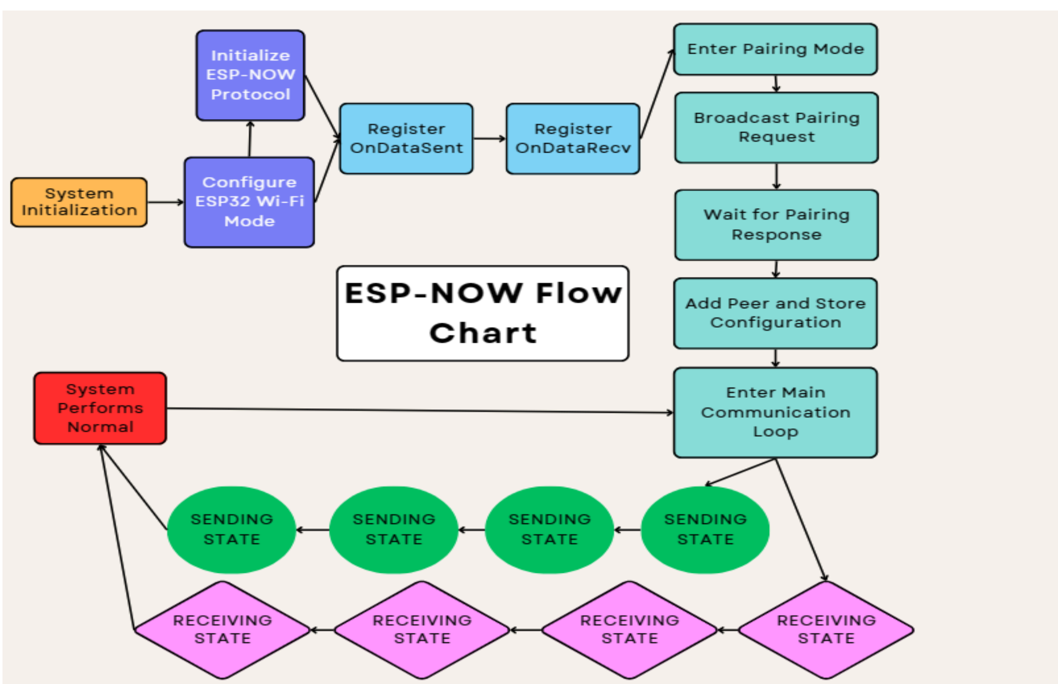


Figure 3 Flow chart of ESP-Now protocol implementation

In practice, the ESP-NOW protocol within our system functions as the backbone, connecting various modules such as medication dispensers, environmental sensors, and user interface components. For instance, when a medication dispenser identifies that it's nearing time for a dosage to be delivered, it utilizes ESP-Now to instantly notify the central microcontroller, which then processes this information and activates a series of reminders. These reminders may manifest as visual alerts on displays, auditory signals through speakers, or even tactile notifications for users with special needs. Furthermore, the integration of ESP-Now into our project resolves several design challenges. For instance, the capacity for a device to dynamically choose the best communication channel allows us to maintain reliable communication even in environments littered with potential sources of wireless noise. Additionally, the straightforward pairing process inherent to ESP-Now simplifies the user experience, making it less intimidating for individuals who may not be tech-savvy.

The medication dispenser needs to alert users when it's time to take their medication. The dispenser obtains time updates from an NTP server and determines when to trigger the reminders based on the medication schedule stored in the dispenser's system. When a reminder is due, the dispenser sends an alert over the ESP-Now network to the main controller that manages the reminder alerts. We've employed ESP-Now to its full potential, allowing for shortened latency amidst active communication and boosting the reliability of message delivery in real-time scenarios. Our implementation leverages struct_message and struct_pairing types to encapsulate the information sent across the network. Struct_message carries essential information, such as the message type (DATA or PAIRING), the ID of the sending device, and the reading ID, whereas struct_pairing is used for initial pairing requests.

Our main.cpp file showcases the ESP-Now setup, including the initial peer-to-peer pairing. When a device boots, it attempts to connect with the network's main controller by broadcasting a PAIRING message type. Upon receiving a PAIRING message, the main controller responds with the necessary configuration data, including its MAC address and the channel to use for future communication, thereby establishing a link with the new device. This dynamic peer addition, as seen in our addPeer function, gives our system the desired flexibility and room for scalability. The send and receive callbacks, OnDataSent and OnDataRecv, handle acknowledgments of outgoing messages and process incoming data packets, respectively. This ensures that our system is always synchronized, maintains data integrity, and operates efficiently. These callbacks are pivotal for evaluating the message types and for implementing actions based on received data.

In response to received messages, our system may present information on an LCD display monitored by a TFT_eSPI library or trigger auxiliary signaling like alarms or lights. In aux.cpp, we highlight the use of accessory functionalities, such as the audio module and LED indicators, both of which are controlled via received ESP-Now messages. These peripherals create a more intuitive and user-friendly experience, a testament to how we've embedded ESP-Now communication in actual device operations. The design consideration stretches to error handling and maintaining the overall system integrity. The millisDelay setup in both files ensures that ESP-Now communication is non-blocking and does not adversely affect the system's other real-time tasks. We spaced out ESP-Now transmissions (EVENT_INTERVAL_MS) to avoid network congestion and ensure efficient power usage. Our system's versatility is further exhibited by including a web server component, as tagged in our code snippet "Start Web server." It serves a dual purpose: one for configuration via a web interface and the other as a point for data retrieval via API requests to update and maintain the medication schedules.

Finally, our implementation plan addresses potential limitations such as the small payload size (250 bytes) by optimizing the struct_message format to send only necessary data. We are mindful of possible interference within the 2.4 GHz frequency band and mitigate this by carefully choosing transmission channels and considering the domestic setting where other wireless devices are present yet operable within the band's scope. In essence, our conscientious application of the ESP-Now protocol within the Medicine Dispenser Reminder System exemplifies how a purpose-built network can lead to an efficient, responsive, and user-friendly device ecosystem—qualities paramount in healthcare applications where timeliness and reliability are non-negotiable.

3.5 *Detailed operation of Auxiliary subsystem.*

The Auxiliary subsystem must maintain constant power through the USB-C connection. This subsystem serves primarily as the repetitive reminder function of the medicine dispenser, therefore it needs to output a repeating auditory reminder through the speakers, while lighting the LED for a visual reminder. The audio also should stop playing if the user has indicated that the medicine has been ingested.

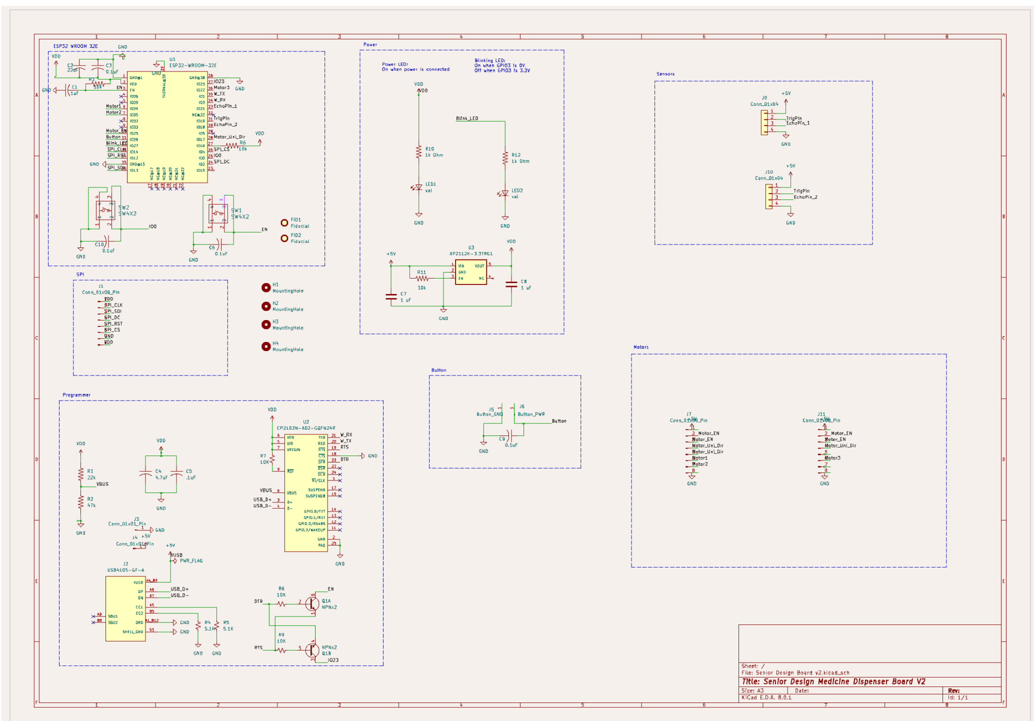


Figure 4 Snapshot of the Auxiliary schematic

The Auxiliary system uses an ESP32-WROOM-32E microcontroller for its internal DAC pins at GPIO 25 and 26. The external peripherals used were connected through I2S and SPI protocol of ESP32. I2S (Inter-IC Sound) is a serial bus interface standard used to transmit digital audio data between integrated circuits in electronic devices, commonly found in audio applications such as digital audio players and amplifiers. This system uses two I2S 3W Class D Amplifier Breakout boards to connect to two speakers for audio output. The system also used a MicroSD card breakout board to read the audio file that is stored in the microSD card with SPI protocol. These components are necessary to output audio through the ESP32 microcontroller. The ESP32-WROOM-32E has ESP-Now capabilities, offering ample pinouts, storage, and comprehensive documentation. To facilitate code testing, we incorporated a debugger/programmer circuit into the PCB design, featuring a USB-C connection, CP2102N chip, and NPN circuit. Additionally, we integrated power and blink LEDs for diagnostic purposes. For external connectivity, we included an SPI interface with a Molex connector for connection to a microSD breakout board. Another Molex connector was designated for the I2S amplifiers. Furthermore, we reserved pinouts for optional LED attachment. To ensure stable voltage regulation, the circuit board incorporated an AP2112K voltage regulator.

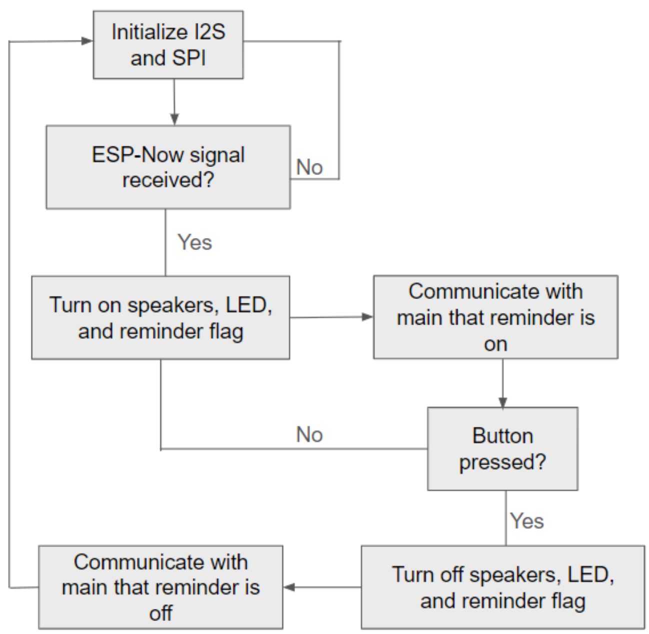


Figure 5 Flow chart of Software Logic for Auxiliary Systems.

As the motor on the main housing system turns, indicating that it is time for medication, the main system's microcontroller will communicate with the auxiliary's microcontroller through ESP-Now. Upon receiving the signal from the main system, the aux system will initialize the SPI and I2S peripherals and output the audio reminder. The LED indicator will also be turned on for a visual indication. When the user pressed the button on the main housing system, indicating that the medicines had been taken, the main controller would again send a signal to the aux controller which would then turn off the speakers and the LED. The coding of the auxiliary system is done in the Arduino Integrated Development Environment (IDE), in the C++ language. This environment is chosen to easily incorporate various Arduino libraries that are needed for I2S initialization and ESP-Now configurations.

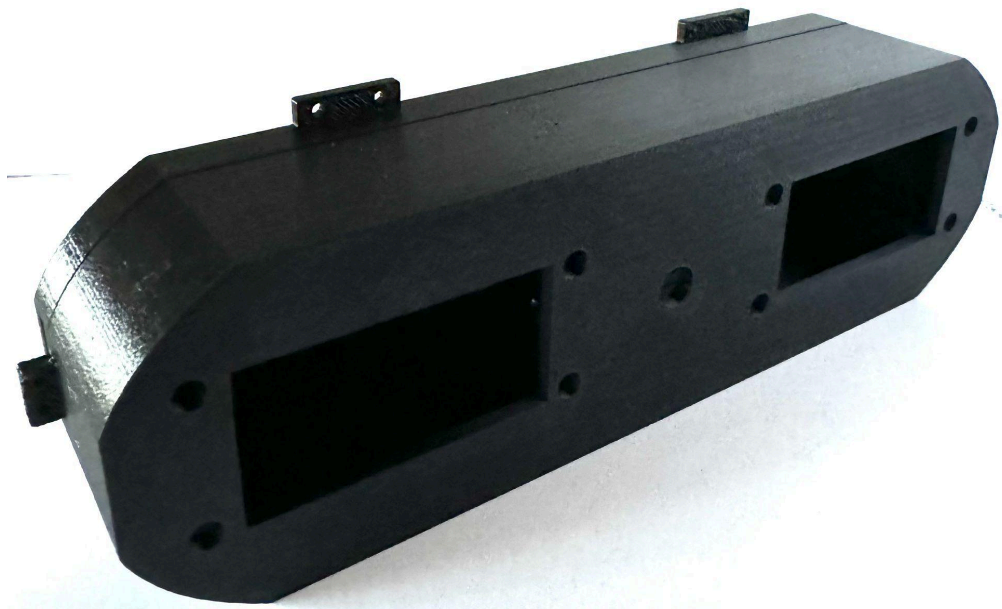


Figure 6 Final 3D printed housing of Auxiliary system

The CAD design of the auxiliary system is done using Fusion 360. The two apertures on the front of the design makes space for two speakers, and the opening in the middle allows visual cue from the LED. The PCB board and other electronics are secured inside of the 3D printed housing. Since the system is powered by USB-C connection, the aperture on the side allows the cord to connect to the microcontroller.

Subsystem Testing:

The Auxiliary system was tested in individual components. The auditory component of this system consists of I2S and SPI protocol, which are connected to the I2S amplifiers and the microSD breakout boards. To ensure each of these components operate correctly, we first sampled codes to test the audio output of the speakers, and the initialization of the microSD breakout board. Then, we uploaded the audio file that will play through the reminder to the microSD card once everything worked correctly.

To test the ESP-Now communication, we again sourced test codes to observe how well the boards communicate. We verified that data packets are successfully transmitted and received between the boards before implementing it into our main codes.

3.6 *Detailed operation of Website subsystem*

The subsystem requirements for the website/application entail several key components to ensure functionality and usability. The subsystem we are discussing here is the user interface for the prescription input and reminder notification in our web-based application. The user interface subsystem primarily consists of software components, so we will focus on a flow chart to illustrate its operation. The flow chart in Figure 7 outlines the process of user authentication, prescription input, the backend, error handling, and communication with the microcontroller.

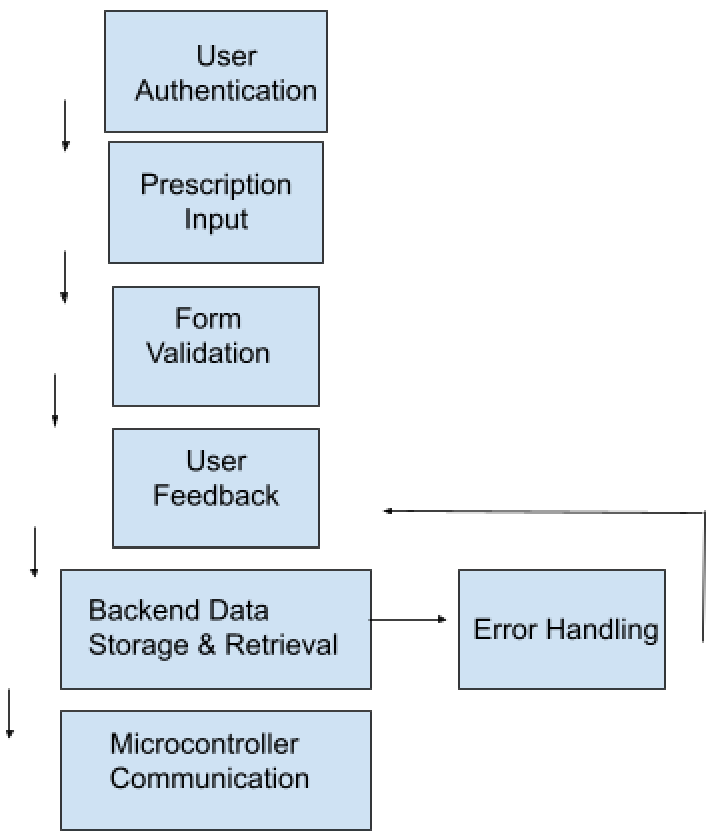


Figure 7 User Interface Subsystem Flow

The function of the user interface subsystem is to provide a user-friendly interface for users to input prescription details, authenticate their registration/login, and provide notification reminders. It interacts with the backend system for data storage and retrieval and communicates with the microcontroller to send reminder commands.

We used a combination of React for front-end development, Java for back-end, and Visual Studio Code for development. We chose React for the frontend development due to its component-based architecture, which allows for modular and reusable code. Additionally, React provides a smooth and interactive user experience. Java was selected for backend development as it offers robustness, scalability, and security. We used Visual Studio Code (VSCode) as our integrated development environment (IDE) for its lightweight yet powerful features and excellent support for React and Java development. Back4App was chosen for user data storage due to its robust features and ease of integration. Back4App was chosen as the backend service for its ease of use, scalability, and security features. It provides a cloud-based backend platform that simplifies data storage, user authentication, and real-time communication.

The development process began with creating an outline of the website's structure and flow, focusing on intuitive user interfaces for prescription input. This process involved implementing input forms for prescription details using React components, incorporating Back4App for user data storage and retrieval. Next, creating a confirmation screen to display user inputs and allow for editing. Authentication functionality was then added to enable users to register and login securely. Finally, we connected the user's ID to their respective controller so that prescription inputs are associated with specific user IDs in Back4App, to ensure personalized reminders. The subsystem was designed to communicate with other subsystems through secure communication protocols, ensuring seamless integration with the backend system and the microcontroller.

The subsystem was designed to communicate with other subsystems through secure communication protocols, ensuring seamless integration with the backend system and the microcontroller. Specifically, the user interface reacts to user actions or events, such as button clicks or form submissions, through event-driven programming. State-driven programming maintains different states to manage the flow of the application, transitioning between states based on user actions or responses from the backend. Communication protocols such as HTTP for RESTful APIs and WebSocket for real-time communication enable data exchange between the frontend and backend systems, facilitating features like data storage, retrieval, and real-time notifications.

Each subcomponent was tested individually to ensure functionality and compatibility within the overall system. For the front-end, we tested user interface elements such as forms, buttons, and navigation menus to ensure they functioned as intended. This included testing for responsiveness, accessibility, and user interaction, ensuring a seamless experience for all users. In the back-end, we conducted thorough testing of data storage and retrieval processes to verify the integrity and security of user data. This involved testing various scenarios, such as data insertion, retrieval, updating, and deletion, to ensure data consistency and reliability. We also focused on testing security measures, such as encryption and access control, to safeguard user information from unauthorized access or manipulation. Additionally, integration testing was done to ensure seamless communication between the front-end and back-end components. This involved testing the flow of data between different layers of the application, ensuring that data was transmitted accurately and efficiently. We verified that the front-end components could successfully communicate with the back-end services and that the integration points were functioning correctly. Overall, rigorous testing processes were employed to guarantee the reliability and effectiveness of the subsystem in meeting user needs. They helped identify and address any issues or inconsistencies early in the development process, ultimately leading to a robust and user-friendly interface for the medicine dispenser application.

3.7 *Detailed operation of Controls subsystem*

The control subsystem primarily contains the inputs and outputs on the main housing element. This includes the LCD display, motors, button interfaces, and general PCB design. The main requirements of this subsystem are to accurately determine if it is time for a new set of medication and reveal that medicine at the designated time. The system then needs to display information regarding the medicine to be taken and notify the adjacent systems to turn on further notifications. The subsystem also needs to intake a notification when the medicine has been taken as a button being pressed. This will turn off the notifications. This subsystem will directly work to complete the following general product requirements:

- Show Drug Reminder Information at Time of Medicine Intake
- Timely End of Reminders Following Medicine Intake

This subsystem consists of a 4” LCD screen with SPI capabilities, a motor, and button. The PCB subsystem is powered by an outlet through a USB-C connection in order to provide constant power to the system. The system works around a ESP32-WROOM-32E which can interface through GPIO, I2C, SPI, and UART. The ESP32-WROOM-32E also has internal communication protocols called ESP-Now which was another reason to use the ESP32-WROOM-32E. The motors are controlled by the PCB having a GPIO connection through a motor driver. The LCD display uses SPI which is manageable by the ESP32-WROOM-32E. The button is read through the GPIO connection using an interrupt.

The real time clock portion of the control subsystem utilizes the ESP32-WROOM-32E’s capability for accurate real-time clock functionality by referencing NTP global servers for precise time tracking. It also syncs the real-time clock with external time sources, accounting for timezones and daylight savings time changes. A failsafe mechanism prevents medication being revealed if the current time does not match the scheduled intake time.

The centralized control system for both the main device and auxiliary systems create a unified control interface accessible through the device's GUI and the web-based platform. We ensure seamless communication between the main system and auxiliary devices through ESP-Now, retrieve medical information from the prescription database, and display relevant information, such as dietary instructions, side effects, or specific precautions on the device's interface during the medication intake time. The output audio (speakers) and visual (LEDs) reminders will ensure correct intake time. We will also implement a feedback mechanism, a button, to signal successful medication intake.

The display interface, crucial for providing information to users, uses an LCD display with the SPI protocol. Powered by the board, and connector types are specified to seamlessly integrate the display into the dispenser. This component requires the code to interface with the website to retrieve the medicine data for the specific user.

The motor control interface consists of the microcontroller controlling the input voltage to the motors, controlling an external motor driver board which in turn will drive the motors. The voltage can be provided by the PCB at 5V.

The subsystem requires an input voltage range within specified limits and stipulates the necessary current requirements. Additionally, voltage regulation mechanisms must be implemented to ensure stable operation. The board employs an USB-C port for power and connection. The PCB board is then able to output 3.3V or 5V power to each of the connected components.

In a broader look at this subsystem the following figure is a snapshot of the larger schematic shown in the Appendix. This snapshot focuses on the components that ended up used in the final design. Primarily this excluded the JST connectors for the sensors and second and third motors which as discussed in the housing subsystem section became obsolete with a design change.

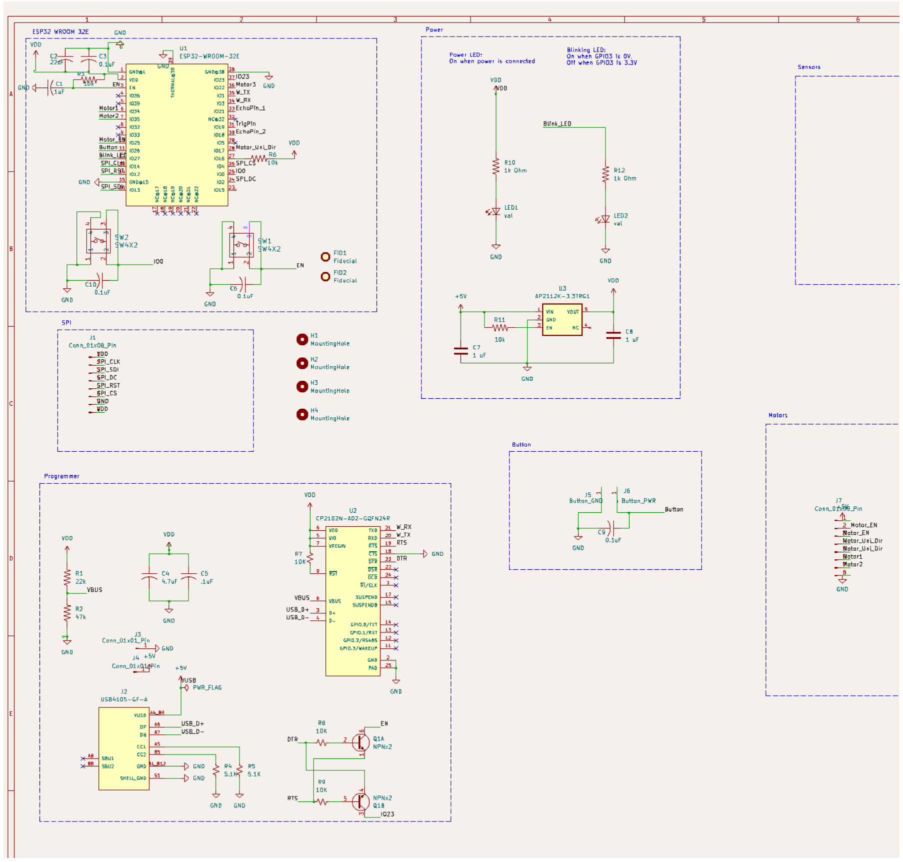


Figure 8 This is a snapshot of the large main PCB schematic cropped to only include the components used in the final design.

We chose the ESP32-WROOM-32E for the additional pins it would provide for all the original components we wanted to include on the board which included three motors and two sensor elements which as discussed early became obsolete. Regardless, the ESP32-WROOM-32E still had ESP-Now capabilities, an abundance of pinouts, storage, and documentation. In the design of the PCB we included a debugger/programmer circuit so we could work on testing the code with this board. This included the USB-C connection, CP2102N, and NPN circuit. We also included power and blink LEDs for testing. From there, we included a SPI interface which was connected through a Molex connector to the LCD display screen. We did another Molex connector for the motor driver boards. We also left open pinouts for the button to be soldered and connected to. The circuit board also included an AP2112K as a voltage regulator for the circuit.

In terms of component choices, the motor was discussed in the housing subsystem section, but the LCD screen and button were decided as part of this subsystem. For both the LCD screen and button the primary factor was size. We wanted a large LCD screen and a large button for easy readability and tactility. In addition, we wanted the LCD screen to have SPI capabilities which this one from Hosyond did. The button was going to be connected to an interrupt input GPIO pin.

This subsystem interfaced with the website as well as the auxiliary ESP32, but these interfaces and related communication protocols are discussed further in the Networks Subsystem.

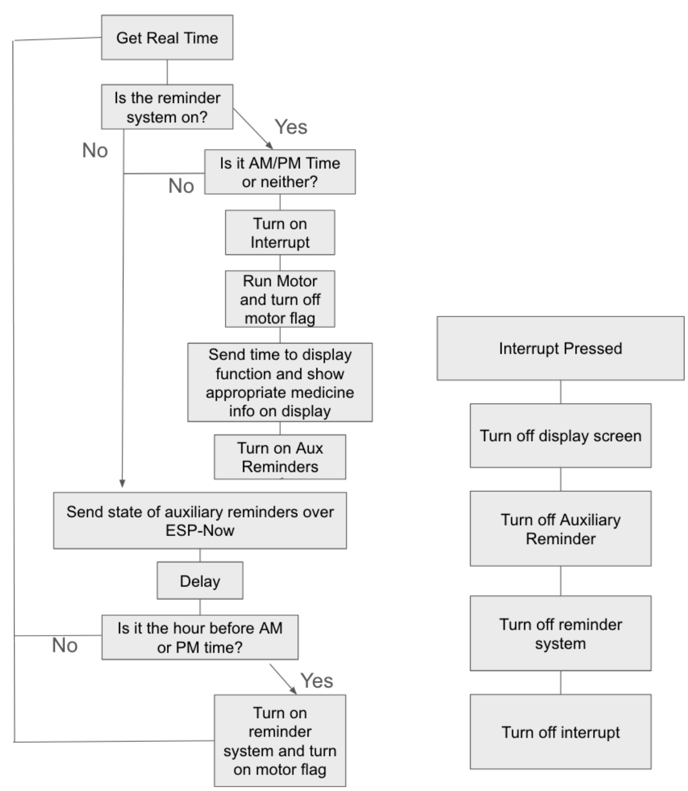


Figure 9 This flowchart shows the primary logic for the main code which is interrupt driven.

The figure above shows the flowchart for the main code. We chose to use C based arduino code because of how easily it interfaced with the ESP32-WROOM-32E and how easy it was to include various libraries in our project. The inclusion of an ability to adjust libraries for our project was essential because we had set up files for our display screen that needed to be edited before we could use that functionality.

The logic for the main code is shown in the figure above and starts each main loop with retrieving the real time values for its timezone. It then checks if the reminder system flag is turned on. If it is, then we next check the time and see if it is either the AM time or PM time set for daily reminders, otherwise we leave the loop. If it is, then we turn on the button interrupt, run the motor once and turn off the motor flag, send the time to the function which places the medicine information onto the LCD display associated with the time of day and turns on the auxiliary flag. If it's not, it skips right to the next function which is sending the auxiliary system the state of the auxiliary flag over ESP-Now. The code then uses a millis delay to stall before checking if it's the hour before the next AM time or PM time. If it is, it will turn back on the reminder system and motor flag. Otherwise it will start at the beginning again getting the time of day. The entire main code is interrupt driven so when the button is pressed it triggers an interrupt in which the display screen is turned off, the auxiliary reminder flag is turned off, the overall reminder system is turned off and the interrupt itself is turned off.

In order to test this subsystem we started by testing the pcb was able to get power and be recognized by the computer. We then checked it was programmable by running the basic blink code. We began adding each component one at a time. That meant going back to the original tester code for each component and running the most basic function. We did this for the motor, button, LCD screen, and ESP-Now. After ensuring all the connections were correct for each of those components we were able to program the main code file onto the board and troubleshoot any last minute issues.

3.8 Interfaces

All microcontroller units, both main and auxiliary, are powered via USB-C. Users begin by inputting medication details such as dosage and timing on the website. Upon submission, the website updates information displayed on the LCD screen and schedules medication dispensation via WiFi. The main microcontroller, synchronized with real-time data via WiFi, activates the motor at designated dosing times through GPIO pins. Subsequently, the LCD screen, linked to the microcontroller via SPI protocol, shows the medication details inputted by the user. The SPI protocol is a synchronous serial communication protocol used to transfer data between a master device and one or more slave devices. It typically involves four communication lines: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Clock), and SS (Slave Select). The master device controls the timing and data transfer, with each slave device being selected individually for communication.

The main unit communicates with auxiliary units using the ESP-Now protocol to indicate that medication has been dispensed. Upon receiving this signal, the auxiliary system emits repetitive auditory and visual reminders. The auxiliary microcontroller is connected to speakers via I2S, and LEDs via GPIO pins. The Inter-IC Sound (I2S) protocol is a serial communication protocol used to transmit digital audio data between integrated circuits in audio equipment. It consists of separate data and clock signals, allowing for high-quality, synchronous transmission of audio samples. I2S is commonly used in applications where precise timing and low-latency audio transmission are essential, such as in digital audio interfaces and audio processing systems. Audio files for reminders are accessed by the auxiliary system also through SPI. Additionally, the system sends notifications to the user's phone or email upon medication dispensation.

4 System Integration Testing

4.1 *Describe how the integrated set of subsystems was tested.*

In terms of integration testing, as discussed in the control subsystem section and interfaces, there were several components that needed to work together in the main system and several in the auxiliary system. The physical hardware components did not require complex integration. We primarily used flags and "if" statements to turn on and off different system components. The button, motors, LCD display, and LED could be easily controlled. The components that had more complex integrations were the real time component, ESP-Now, website interface, and the speakers.

The real time component, ESP-Now, and website interface all require a WiFi component. We used the random nerds tutorial to begin understanding ESP-Now and there was code for both a standard sender and a standard receiver. Originally we thought the main code would be the sender and send the signals to the auxiliary as the receiver, but after seeing how the WiFi component needed for the real time interacted with the sender code we decided to change it to the standard receiver code. This made both the main board and the auxiliary board WiFi stations instead of just the auxiliary board being a WiFi station which was interfering with the WiFi connection the main board had created for the real time functionality. This was still possible because it was two way communication so both the sender and receiver were sending code out and receiving it. We tested that the different values sent and received were both what we expected and then we added it to the code with the real time to ensure the board was still able to find the time.

The ESP-Now and speaker interface also had a conflict when the need to keep reading for packets was interfering with the audio loop playing. In the original code we used to create the audio loop it was alone in the main loop. We quickly learned that implementing any delay would interfere with the audio. Our solution was to create a "while" loop to keep the non-audio loop code in, which, once it left the while loop it would play the audio. The audio loop function needed to be separate in the main loop code so that there would be no interference so we had to work around that. To test it we sent the various signals to and from the auxiliary system to ensure both were receiving what we expected. We were able to test and get it to, at the designated time, turn on the audio function and LED.

4.2 *Show how the testing demonstrates that the overall system meets the design requirements*

The constant power requirement was tested and met by confirming our team's designed PCB board could power on and turn on the related power LED. We integrated the power system using a USB to USB-C connection. The USB-C connection on the PCB board is regulated by a AP2112K to output for various components either a 5V or 3.3V power supply.

The user interface for prescription input and reminder overrides requirement is implemented by two different subsystems that interface together. The website allows the user to input all prescription information which is saved in the Back4App. The control subsystem, which houses the button, is responsible for turning off, or overriding, the reminders.

The website, in addition to the prescription itself, takes in information regarding physician instructions for the medication, dosage information, brand and generic name, and more. This information is aimed to satisfy the ability to adjust basic settings including prescriptions requirements.

Multiple repetitive reminder functions are satisfied with the phone notification, audio, and LED reminder function. These three functionalities work to satisfy the accessibility aspect of our project, for those that need a diverse range of visual, auditory, or personal device type reminders to best suit their needs and lifestyle.

The requirement to dispense the medicine on time is achieved by the real time function determining the time in the timezone and syncing with the AM time and PM time in the code and turning on the motor at the appropriate time to reveal the medicines for that time period.

Showing the drug reminder information at the time of medicine intake is also achieved with the real time function knowing the time in the timezone and showing the AM time medicine as inputted into the website or the PM time medicine as inputted into the website.

Reminders will end with the button interrupt functionality. The user will press the button to change the reminder system flag to off, which will turn off the reminders and send information over to the auxiliary system through ESP-Now to turn off the reminders.

5 Users Manual/Installation manual

5.1 *How to install your product*

Installation of the Medicine Dispenser Reminder System should be a straightforward process accessible to anybody, regardless of their technical expertise. In an ideal scenario, the process follows a simple unbox, set-up, and activate protocol.

Upon unboxing, the patient or caregiver would first find a suitable location for the dispenser unit. A power outlet must be nearby as the unit requires constant power for its real-time functions to operate accurately. Once plugged in, the device's primary systems begin their boot-up sequence, complete with self-diagnostics ensuring all systems are operational. Networking is key, so the next step is to connect the dispenser to the home's Wi-Fi network. The on-device interface walks the user through selecting their network and entering the password, a process akin to connecting any smart device to the internet. Once online, the dispenser syncs with the global NTP servers to accurately keep track of time.

Next, the setup process necessitates logging into the backend database via the web interface. The patient or caregiver will input all relevant data pertaining to the medication schedule, dosage amounts, and any specific instructions related to the medication intake. This information is saved and sent to the dispenser unit. Once the information is loaded into the dispenser, the patient fills the corresponding compartments with medications as per the visual guide on the device. After confirming the setup is complete through the device interface, the system is active, and the preset schedule starts.

5.2 *How to setup your product*

Upon successful installation, setting up the Medicine Dispenser Reminder System for daily use is crucial to its ongoing reliability. This setup, done via the web interface, ensures that the system functions in harmony with the unique needs and preferences of the patient. The initial setup involves personalizing reminder notification procedures. The web interface offers options to choose reminder times, intervals, and methods — be it auditory via the auxiliary devices, visually through LED prompts, or a combination of both. The interface also allows for adjustments in volume, LED brightness, and the type of reminder alert. Ensuring that these settings are tailored to the patient's environment and lifestyle is critical for the system's effectiveness.

Another essential part of the setup process includes detailing the medication schedule. The user can input precise medication details — names, dosages, intake times, and other specific instructions. The system supports flexibility, enabling users to update or change their medication details as often as necessary without complications. During these settings adjustments, connectivity to the auxiliary reminder devices is tested. These nodes are paired with the main dispenser unit via the ESP-NOW protocol, and users must ensure each device's placement within the house is such that signals are not obstructed and reminders are clearly perceivable.

Finally, the user must complete a test run of the medication dispensing mechanism, confirming that the motorized lid opens at the correct time, revealing the correct dosage for the schedule set. This physical movement should coordinate with the activation of all reminder methods set during the personalization of the system. The combination of intuitive installation and comprehensive setup procedures ensures the Medicine

Dispenser Reminder System is not only accessible but also reliable and efficient, leading to increased medication adherence and improved patient health outcomes.

5.3 *How the user can tell if the product is working*

The user should get quick feedback on whether the product is syncing or not because they should immediately see light from the LCD screen on power on. They should also shortly after see the unique identifier on the screen allowing the user to link their web account to their physical product. After they press the button, following inputting the unique identifier into the website, the screen should go black indicating the main program is running. After that, the user should see, at the next AM time the product should reveal the medicine for that time period, display the medicine information the user inputted into the website, and begin playing the reminders. After that, the product should work as expected.

5.4 *How the user can troubleshoot the product*

<u>If the screen does not display a white screen on plug in</u>

- If the user does not see the first screen with the unique identifier, the user should try unplugging the device and plugging it back in.
- If this does not work, move to the “if one component is not working” section and follow directions.

<u>If the user gets past the unique identifier screen, but no additional change at the AM or PM time</u>

- If it is not a power issue, the user should check the WiFi router is working properly to see that the device is able to get the real time and therefore check whether it is time to show the medicine.

<u>If the reminder system does not play or turn on the LED</u>

- Unplug both devices and plug them back in.
- Check the WiFi is working.

- Try moving the auxiliary device around the room or closer to determine if there is a prohibitive surface or wall in the residence that is blocking the signal.

<u>If one of the components is not working: button, motor, LCD screen</u>

- Try previous debugging strategies.
- Call the provider to replace a specific part that is not working. After significant use, the button or motor may wear out.

6 To-Market Design Changes

When going to market there are several features which would need to be made customizable for each individual user. This includes the ability to choose what time the medications are dispensed, how often the reminders are, how bright the LED screen is, how loud the reminders are, and whether they would like to be reminded to refill the dispenser. These are the user interface changes that would make the system commercially viable as without these changes we would need to assist every user in the setup process.

The board for the main system would need a few changes. There is a section which was designed for the sensors which became obsolete through our change in Housing design so that part of the board can be removed. A small but important change would be to ground the strapping pin which allows the upload of code to the board. This is more a change for during further development rather than for the market since individual users will not be uploading any code to their devices.

The Housing has several aspects which should be changed when going to market. The lid to the inner ring which houses the electronics would need to be secured so that the customer would not be able to mess with the electronics. The lid to the outer ring which covers the medications would need to be more secure but also easier to remove and replace to ensure that the customer could easily refill the medications, but that any children in the home would not be able to access the medications. A different motor could be found that is not quite so tall so as to allow the Housing to be smaller. The space between the inner and outer rings should be filled in so that medications could not be dropped into that area by mistake. If a patient who struggles with their ability to pick up medications drops one in that gap it could be very difficult for them to get it out. Another change would be to add labeling around the medicine bins, whether on the outside of the Housing or somehow in the bins themselves, it would be important to make it clear which slot is for which medications. The labels could be customizable by the customer to say what day of the week and what time the medication is being taken.

The Auxiliary has several aspects which should be changed when going to market. The biggest changes that would need to be made for the Auxiliary system would be that the speakers would need to be much louder and the LED much brighter. It might be a good idea to switch the LED to a full screen with words similar to the main system too. The housing for the Auxiliary system would need to be more secure as well, in this version of the product it was held together by tape.

The Website also has several aspects which should be changed when going to market. A big addition that would need to be made would be the ability to see and change what prescriptions are currently in use in the dispenser. It currently allows the user to log in and input new ones and edit the input before accepting it but that does not account for long term use of the product which would involve the patient's medication regimen changing over time. It would also need to be able to send the information gathered to the main system more reliably as it currently takes some work in the back end for the information to transfer. Another huge change would be to add encryption to ensure that the patient's personal medical information is safe and secure. Another change would be to make sure the website has a portion for the user to input wifi preferences so that the real time information can be sent to the systems. This would allow for the time information to be available to the system without it needing to be hard coded on the board which would make it easier for the user to set up the system.

7 Conclusions

The implementation of our medication dispenser system represents a significant step forward in addressing medication non-adherence, a prevalent issue affecting millions of Americans with chronic health conditions. Through this project, we aimed to create a solution that would assist patients in adhering to their medication regimens more effectively. In conclusion, we have successfully developed a comprehensive system comprising a medication dispenser, a website, and auxiliary reminder system, each fulfilling crucial roles in promoting medication adherence and enhancing user well-being.

The statistics regarding prescription medication use in the United States underscore the importance of our project. With over 131 million adults using prescription drugs and an increasing number of prescriptions each year, it is evident that medication management is a critical aspect of healthcare. However, adherence rates remain suboptimal, with approximately 50% adherence reported in developed countries. Non-adherence can have severe consequences, including disease progression, increased hospital visits, and diminished quality of life. Our system aims to mitigate these issues by providing timely reminders and ensuring medication compliance.

The design of our system was informed by the needs of patient groups that face unique challenges in adhering to their medication regimens, whether due to forgetfulness or cognitive impairments. By developing a user-friendly interface and multiple reminder options, including auditory, visual, and phone notifications, we aimed to accommodate the diverse needs of these patients. Additionally, our system allows for easy modification of prescription information, empowering users to manage their medications effectively.

The functionality of our system was validated through rigorous testing and iterative refinement. Each component, from the website interface to the medication dispenser mechanism, underwent thorough evaluation to ensure reliability and effectiveness. While our system met our initial expectations in terms of medication dispensation, information display, and reminder notifications, there were areas identified for improvement. For example, the size of individual medication slots could be increased for easier retrieval, and the timing precision of the dispenser motor could be enhanced for more accurate dosing.

Despite these minor shortcomings, our system represents a significant advancement in medication management technology. By seamlessly integrating hardware and software components, we have created a cohesive solution that addresses the complex needs of patients with chronic health conditions. The system's ability to dispense medication on time, provide clear instructions, and deliver timely reminders contributes to improved medication adherence and ultimately, better health outcomes for patients.

Looking ahead, there are opportunities to further enhance our system's capabilities and expand its impact. Future iterations could include features such as automatic medication refills, dose tracking, and real-time monitoring of adherence patterns. Additionally, efforts to address issues related to device connectivity and user interface refinement will continue to be a priority.

The development of the medication dispenser system has been a challenging yet rewarding journey. By leveraging innovative technologies and interdisciplinary collaboration, we have created a solution that addresses a critical healthcare need and has the potential to make a meaningful impact on patient outcomes. We are excited to continue refining and enhancing the system as we move towards commercialization, with the ultimate goal of improving medication adherence and empowering patients to take control of their health.

8 Citations

[1] Georgetown University, "Prescription Drugs - Health Policy Institute," Health Policy Institute, 2019. https://hpi.georgetown.edu/rxdrugs/

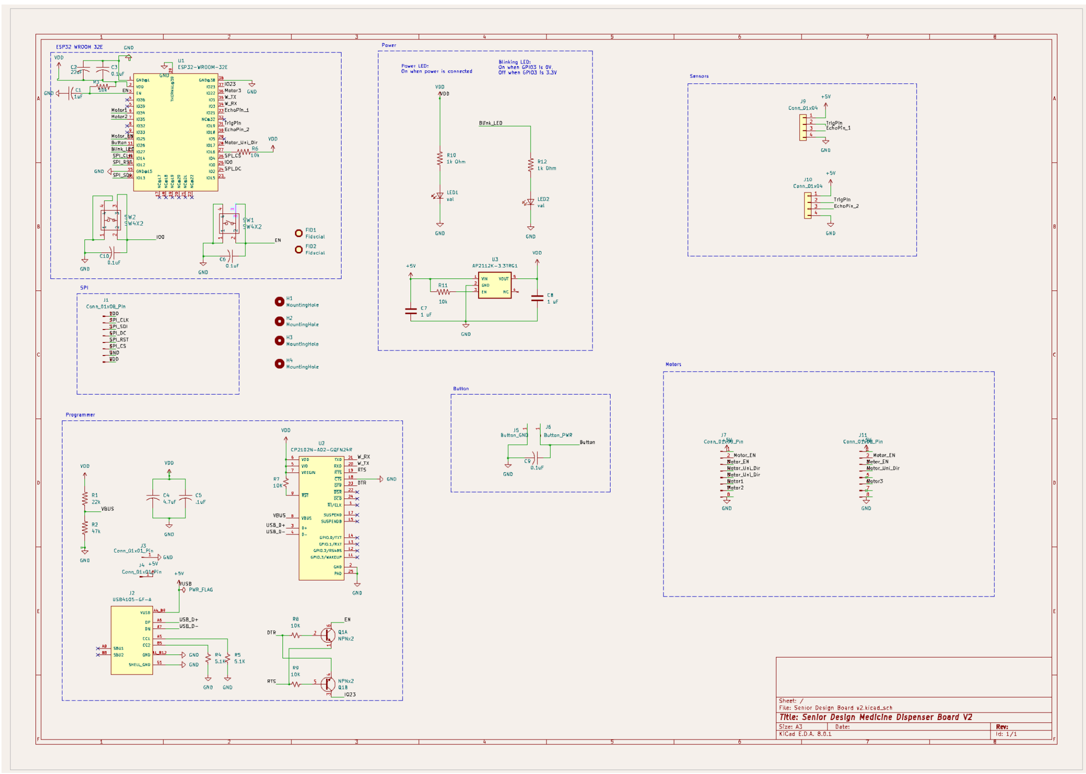
[2] "The Use of Medicines in the U.S. 2023," www.iqvia.com. https://www.iqvia.com/insights/the-iqvia-institute/reports-and-publications/reports/the-use-of-medicines-in-the-us-2023

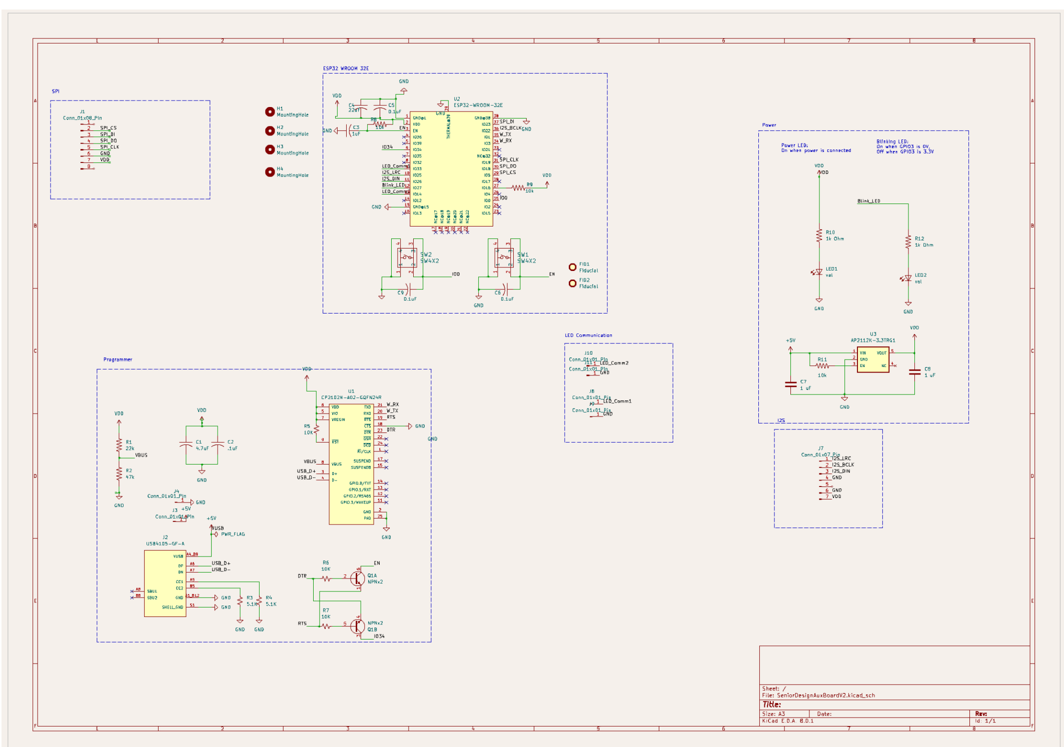
[3] Centers for Disease Control and Prevention, “National Center for Chronic Disease Prevention and Health Promotion,” Centers for Disease Control and Prevention, May 08, 2023. https://www.cdc.gov/chronicdisease/index.htm

[4] B. Jimmy and J. Jose, “Patient Medication Adherence: Measures in Daily Practice,” Oman Medical Journal, vol. 26, no. 3, pp. 155–159, 2011, doi: https://doi.org/10.5001/omj.2011.38.

9 Appendices

9.1 Complete Hardware Schematics





9.2 Complete Software Listings

2.1 Main Project Code and Auxiliary Code:

The main project code, written in C++ for the ESP32 microcontroller, and the auxiliary code, responsible for the operation of the auxiliary subsystem, is included in our website:

https://seniordesign.ee.nd.edu/2024/DesignTeams/pills/top_page.html

2.2 Website/Application Code:

The application code, allowing users to input prescription details, authenticate their registration/login, and receive notification reminders, can be accessed through this github link:

https://github.com/bdavidd09/Medees-master

9.3 Relevant Parts or Component Data Sheets

I2S 3W Class D Amplifier:

https://cdn-shop.adafruit.com/product-files/3006/MAX98357A-MAX98357B.pdf

MicroSD Breakout Board:

https://learn.adafruit.com/adafruit-micro-sd-breakout-board-card-tutorial/download

ESP32-WROOM-32E:

https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

Hosyond 4” Display:

http://www.lcdwiki.com/4.0inch_SPI_Module_ILI9486