INTERNET-BASED NALOXONE SAFETY KIT

A Design Project Report
Presented to the School of Electrical and Computer Engineering of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering, Electrical and Computer Engineering

Submitted by
Yilu Zhou (yz2797), Zane Parker (zap8)
MEng Field Advisor: Dr. V. Hunter Adams
Degree Date: May 2023

Table of Contents

Abstract

Master of Engineering Program
School of Electrical and Computer Engineering
Cornell University
Design Project Report

Project Title: Internet-based Naloxone Safety Kit

Author: Yilu Zhou, Zane Parker

Abstract: Our team seeks to address the ongoing opioid epidemic in the United States by designing and fabricating a highly-functional internet-based naloxone safety kit. The safety kit will provide publicly accessible naloxone, so it is easily accessible in emergency situations. Additionally, the safety kit is designed to automatically call 911 when opened using the Twilio VoIP service. This will reduce the amount of time required for individuals experiencing an opioid overdose to receive medical attention. The kit is wall-mountable, so it can be deployed in almost any space. The safety kit features a sleek and intuitive touchscreen display with labeled buttons and menus for easy navigation. A PCB is used to act as an IO hub, connecting the temperature sensor, door sensor, and built-in speakers. The device is designed to store the naloxone at the optimal temperature range to ensure maximum shelf life. This is accomplished by utilizing a temperature sensor and power-efficient fan. The team utilized laser cutting to fabricate the majority of the enclosure. Locking and mounting mechanisms were also manufactured using FDM 3D printing. In the event of a power or network outage, the safety kit will sound an alarm. Comprehensive software settings are provided, giving users full control over the device, including access to a passcode screen that allows only authorized personnel to modify the settings. Our project offers a potential solution to reduce the number of opioid overdose deaths and improve public health and safety. Using our CAD models, circuit diagrams, source code, and instructions provided for replication, it will be easy and straightforward for others to replicate and deploy safety kits of their own.

Executive Summary

The objective of this project was to design and develop an Internet-based Naloxone Safety Kit, with the ability of offering safe, reliable, and temperature-controlled storage space of publicly accessible Naloxone, with automated 911 emergency service reporting when opened. The opioid epidemic in the United States is a substantial public health concern, and the Naloxone Safety Kit provides a practical solution to help reduce fatalities because of opioid overdoses by making Naloxone more accessible and reducing the time it takes for emergency services to reach overdose victims.

To accomplish this objective, the project group designed a hardware system that consists of a printed circuit board (PCB) to function as an IO hub and connect different sensors and audio speakers. The group also developed a customized enclosure utilizing 3D modeling software, which integrates a handle limit switch sensor and an alarm system to detect and report when the enclosure is opened. The enclosure was designed with an artistic reminder of the opioid epidemic while also maintaining an easily recognizable appearance.

The software system consists of a Python application that offers users real-time status updates and can be accessed from the touch screen on the safety kit. The software was developed with multi-threaded processing to permit concurrent operations, and it was programmed in Python, utilizing Twilio service. The team also ensured that the software is compatible with a wide variety of operating systems and hardware for testing purposes.

Finally, the group conducted comprehensive testing on the system to guarantee its reliability and effectiveness. Testing included PCB performance, temperature control, handle limit switch sensor as well as alarm system, automatic 911 emergency service reporting, and software compatibility. The Naloxone Safety Kit successfully passed all tests, and the team concluded that the system is a reliable and effective solution to the opioid epidemic.

In conclusion, the Internet-based Naloxone Safety Kit is an important device in the combat against opioid overdoses. The kit's software and hardware systems were carefully developed, evaluated, and optimized to provide a safe and trustworthy product. The team believes that the Naloxone Safety Kit has the potential to save many lives and looks forward to continuing its development to address various other public health challenges.

Work Distribution

Work on the internet-based naloxone safety kit began in the Fall of 2022 and ended in the Spring of 2023. Yilu Zhou and Zane Parker both made individual contributions that helped accomplish this project's goals. In the following subsections, the individual contributions of each team member are specifically outlined for each semester of work.

Fall 2022

Both students did individual research and collaborated on a high-level design that would meet design specifications. Once a high-level design was agreed upon by both team members and our faculty advisor, work was delegated to each team member. Our high-level design was broken into three main efforts: a mechanical design effort, a circuit design effort, and a software design effort. In the Fall semester, we primarily focused on the software and mechanical design efforts. Yilu was responsible for software design, and Zane was responsible for mechanical design. Although a lot of the work was completed individually, a significant amount of work was also completed collaboratively. During the fall semester, the team met three times weekly to work on our design project as a group.

Spring 2023

During the Spring semester, the work was delegated the same way it was during the Fall semester. Yilu remained responsible for writing the software and also took responsibility for designing our PCB circuit. Zane continued working on the mechanical design and fabrication of the safety kit's enclosure. Both team members collaborated to integrate the designs and test the prototype. The team also worked together on documentation including a device assembly guide and software user manual. During the spring semester, the team met twice a week to work on the design project as a group.

Introduction

The opioid epidemic in the United States has resulted in a rising number of fatalities due to opioid overdoses annually. Naloxone can reverse an opioid overdose if it is administered in time. However, frequently, emergency services show up far too late. To deal with this issue, our group develops and produces a naloxone safety kit that can be deployed in public areas, offering safe and dependable storage space of naloxone at a regulated temperature level. When the kit is opened, it will immediately call 911 and request medical emergency services, potentially decreasing the time it takes for assistance to get to overdose victims. This project report offers a thorough background on the opioid epidemic, specific problems addressed by our project, our method to develop and produce the naloxone safety kit, and a summary section. Our project provides a potential solution to alleviate the growing problem of opioid overdoses and improve public health and safety.

Design Problems and Requirements

Background

Naloxone is a medicine used to quickly reverse the effects of an opioid overdose and is the typical treatment for opioid overdose. On March 29, 2023, the United States Food and Drug Administration (FDA) authorized Emergent BioSolutions' Narcan nasal spray for over-the-counter (OTC) sale, making it the first naloxone product authorized for use without a prescription. This decision allows customers to acquire naloxone nasal spray directly in drug stores, convenience stores, supermarkets, gas stations, and online. The OTC Narcan nasal spray will need a modification in labeling for the currently approved 4 mg generic naloxone nasal spray products that rely on Narcan as their reference listed drug product.

The authorization of OTC naloxone nasal spray offers potential implications for the creation of the Naloxone Safety Kit. Typically, naloxone is provided through intramuscular shots by experts, and since 2015, it has been offered through Narcan Nasal Spray for prescription use. Naloxone safety kits, such as NaloxBox, have been presented to the marketplace and can be installed on wall surfaces of public locations like schools and shopping centers. Nevertheless, these sets cannot call the emergency service automatically when opened up, but rather set off a siren. The Internet-based Naloxone Safety Kit being designed by our team aims to address these issues. It will immediately call 911 when opened to report the location of the event and can be deployed in public areas. With the current FDA authorization of OTC Narcan nasal spray, it might be feasible to include this product in the Naloxone Safety Kit and decrease the cost of the entire safety kit while accomplishing these enhancements.

Design Requirements

The naloxone safety kit was developed with a number of essential requirements in mind. Firstly, the device needed to be reliable and accurate in its measurement of temperature level and detection of door status. To guarantee accuracy, the temperature sensor needed to have a resolution of at least 0.1°C and a maximum error of 1°C. The door sensor needed to reliably detect the state of the door. Second, the device needed to be easy to use and simple to operate, with clear visual indicators for the status of the system. Third, the device needed to be compact, with a light-weight and durable enclosure that could endure a variety of environmental conditions. Fourth, the device needed to be affordable and scalable, with a manufacturing cost of no more than $250 each. Lastly, the device needed to be designed with future scalability in mind, with the capability to integrate with other systems and sensors as needed.

Enclosure Design and Fabrication

3D Modeling

For the mechanical design, we took an iterative design approach. The 3D model of the naloxone safety kit went through 9 design iterations total. The first design iteration was very simplified and more of a concept than a full design. Our first design was essentially a cabinet with 2 compartments. The raspberry pi, battery, and other electrical components would be stored in the top compartment and the naloxone would be stored in the bottom compartment. Naloxone can be stored at a range of temperatures that is near room temperature. To ensure that the naloxone stays at the optimal temperature range, our design also included a vent.In the first design iteration, only one vent was added to the top of the enclosure. The idea was that the heat would rise and escape through the top of the enclosure. Our idea was to add a vent to the top and include some sort of fan as well. The first design was also designed to be 3D printed as one part without needing any support. This would have resulted in a high quality, yet simple to 3D print part. This design is shown in figure 1 below.

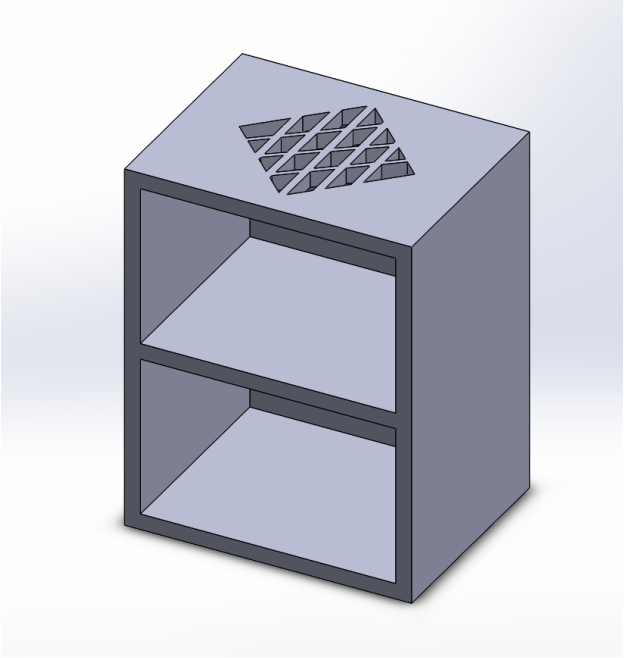

Figure 1: CAD Model for First Design Iteration.

As the design was refined, we ended up with the design in Figure 2. More vents were added to the sides of the enclosure to promote more airflow. We performed some back-of-the-napkin calculations and determined that we may need a small fan. The ventilation was added to the design so that air could be pulled through the bottom of the enclosure and blown out the top. Screw holes were also added for mounting the enclosure, attaching a front panel, and attaching

the cabinet doors. We also tried to improve the general appearance of the enclosure and make it look more professional. This design is shown below in Figure 2.

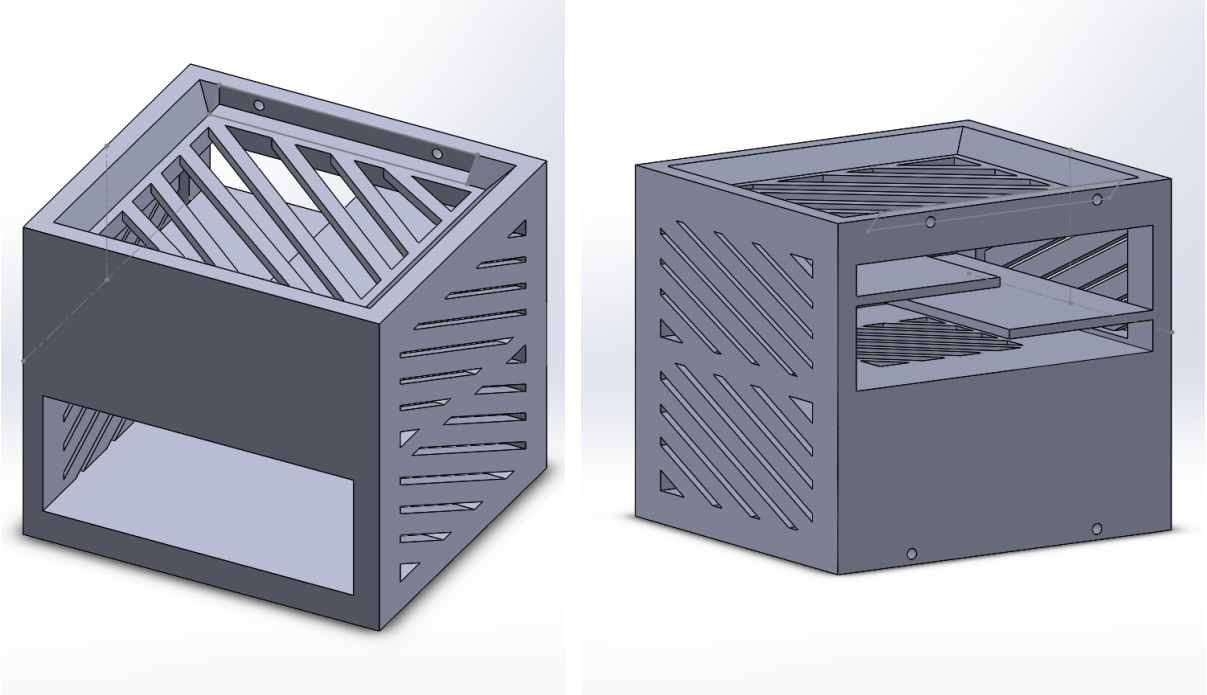

Figure 2: CAD Model Final 3D Printing Based Design

A few iterations of this design continued that featured slight differences.

At the end of the Fall semester, we decided to move away from 3D printing. Using 3D printing would have taken a long time and it also would have severely limited the maximum size of the safety kit. We had decided to implement a touch screen on the front of the enclosure, so the size could not be reduced by much. For these reasons, we decided to rework the design and manufacture the safety kit using laser cutting.

Using the same concept, we came up with the design shown in Figure 3. This design is similar conceptually to the design shown in Figure 2, except it can be manufactured via laser cutting. We chose to manufacture our enclosure via laser cutting due to its simplicity and speed. An enclosure that took days to 3D print could now be manufactured in less than a day. We also added holes for cords, a door, and a locking mechanism. The locking mechanism is simple to open, but must be opened deliberately. This will prevent the box from being opened accidentally. We also decided that our enclosure should be red. This will let users know right away that it is a piece of medical infrastructure.

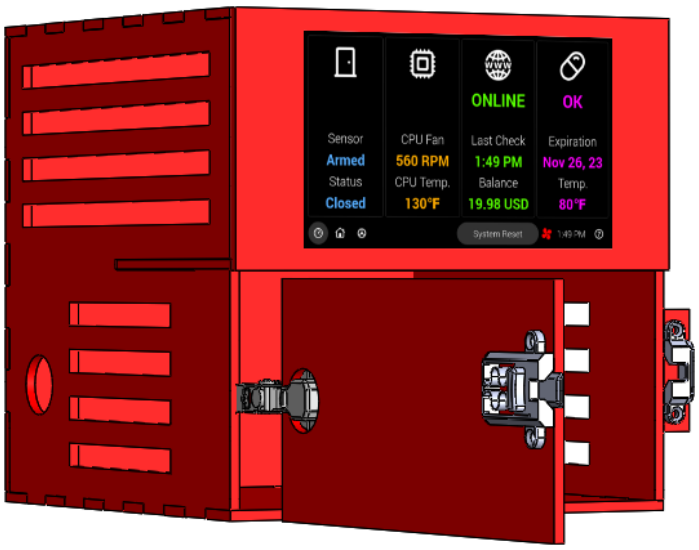

Figure 3: 3D Model for Final Design Iteration

Manufacturing Process of the Enclosure

The manufacturing process of the enclosure was designed to be relatively simple. First, the parts shown in Figure 4 were each cut out of 12” x 12” acrylic sheets. Each part is roughly 10”x10”. This results in a very large enclosure, while also ensuring that variations in the acrylic sheet’s dimensions will not be a problem. After all the parts have been cut out, they can be glued together using acrylic glue. The acrylic glue melts the acrylic parts together so they can be joined in a simple, yet durable way. Small holes were drilled into the enclosure to screw on the touch screen, cabinet hinge, and locking mechanism. We simply drilled through the acrylic with a hand-held drill and connected our components using nuts and bolts. The locking mechanism was manufactured using 3D printing. However, the locking mechanism could easily be replaced with an off-the-shelf alternative.

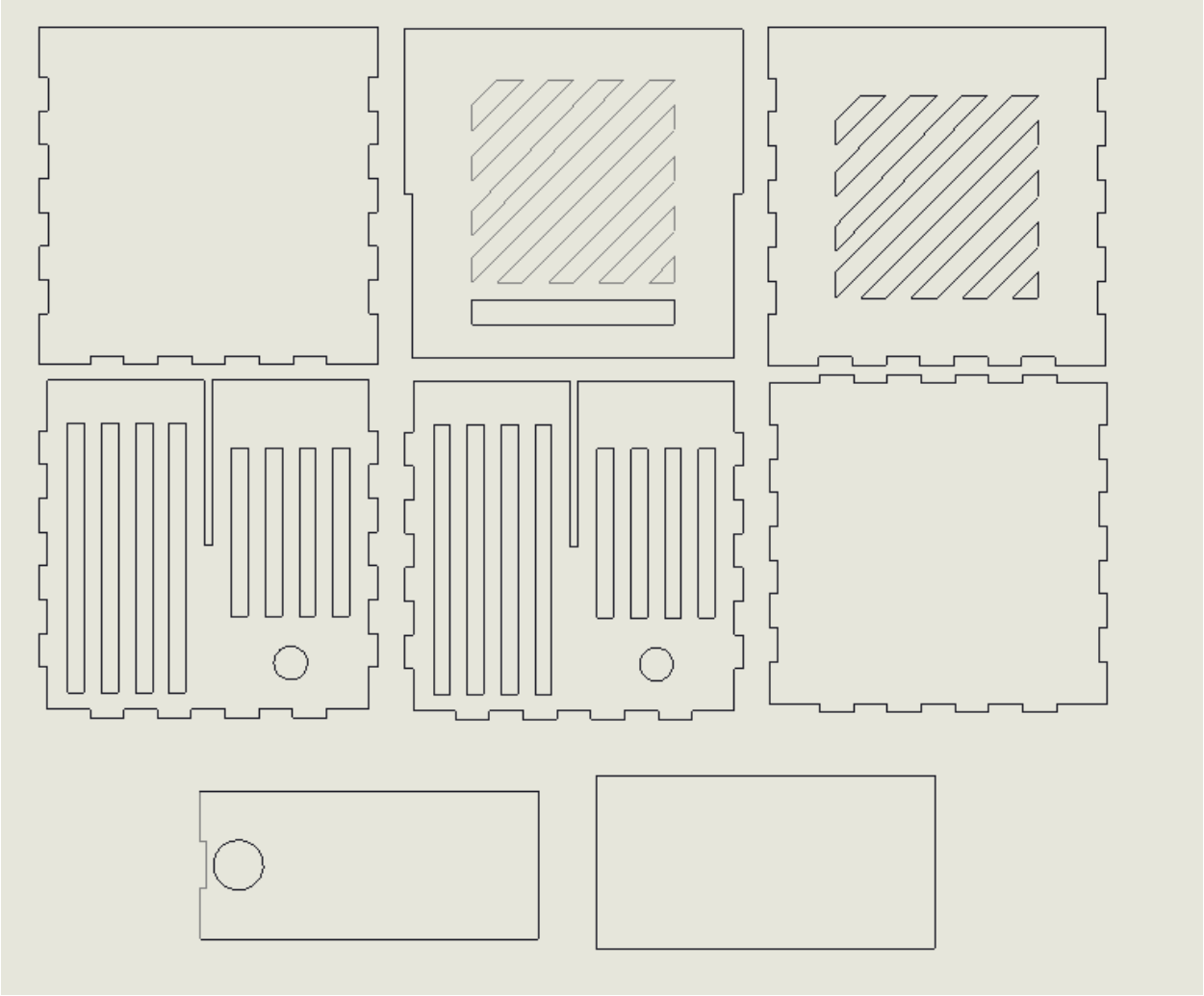

Figure 4: Schematic Containing all Enclosure Parts

Electrical System Design and Implementation

Overview

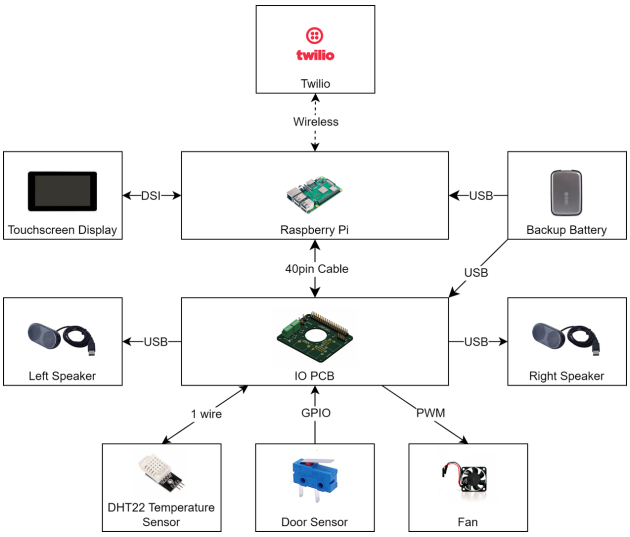

Figure 5: Block Diagram.

The Naloxone Safety Kit utilizes a Raspberry Pi and PCB to act as an IO hub, connecting the DHT22 temperature sensor, door sensor, and built-in speakers. The device is designed to prevent internal temperature from exceeding the recommended storage temperature for Naloxone by a power-efficient fan. In the event of an emergency, the device automatically calls emergency services via the Twilio calling service or plays an alarm through the speakers if the service is unavailable.

Cooling

The maximum permitted storage temperature for Narcan nasal spray is 104 degrees Fahrenheit. Above this temperature, authorized personnel should replace the Naloxone due to overheating. Besides Naloxone, the Raspberry Pi also needs a cooling fan to prevent the CPU from overheating. Since the Raspberry Pi is the only heat source in the safety kit, the team assumes that it is operating at 6.4 Watts, according to the research done by The Dramble (Geerling, n.d.). Under these assumptions, the airflow requirement for the fan is 2.18 cubic feet per minute, assuming the ambient temperature is 68 degrees Fahrenheit and the maximum temperature is 77 degrees Fahrenheit. If the team allows the maximum temperature of Naloxone to be 104 degrees, the final design would not need a cooling fan for the Naloxone cabinet because it is unlikely that the cabinet temperature will be as high as 104 degrees.

Under these assumptions, the team decided to design a cooling system that primarily targeted the Raspberry Pi instead of the Naloxone. By installing the cooling fan at the bottom of the top cabinet, it can extract the cool air from the outside through the meshes. The cool air will also

help reduce the temperature of Naloxone before cooling the Raspberry Pi. Since the heated air has a lower density, the heated air will then move upward through the top panel. The team chose the official Raspberry fan sold by the Raspberry company because of the low cost. The fan also allows us to design our fan curve since the PWM signal controls its speed.

Display Selection

It will be helpful to use some indicators to tell the user about the system's status, such as the phone call status, during the usage. The team needs to find the most suitable technology for displaying the information. The top priority is to make things clear and intuitive to the users.

There are multiple selections to display helpful information on the safety box. One possible solution is to use the LED matrix to show patterns to the users. A sufficiently large LED matrix is needed to achieve the level of detail for showing characters. Also, the LED matrix needs to be powered externally from the wall socket because of the enormous power consumption of around 10 Watts for a 16*32 matrix, the smallest one to show multiple characters (Adafruit Industries, n.d.). The final design cannot show anything on the LED matrix during a power outage.

Another viable solution is to use the e-ink display because of the meager power consumption when idling. The image will still be on display during a power outage, although it cannot refresh itself during an outage. One major drawback is that the time interval between refreshes is very long. From the manufacturer Waveshare, the team found that the refresh time for different displays varies from less than one second to more than 30 seconds. Since the team wants a responsive display, the e-ink technology is not ideal.

The final selection is to use a 7-in touch display in our system. Because of the large screen size, the team can put virtual buttons on the display so the user can interact with the system using a touch display. The team also designed the software around the touch display so the user can enter information using a virtual keyboard.

Door Sensor Selection

Door sensor is used to signal when someone opens the door. Our initial selection is to use a reed switch because of the simple design. A reed switch is two separate metal clips. When a magnetic field is present, the two metal clips will attract together so that the circuit is now closed. However, after the testing, the team found that the metal clips can become magnetized gradually and always make contact even without the magnet. Therefore, the team abandoned the reed switch because of the flaw.

The second selection is to use the microsensor, such as a small inertial measurement unit (IMU), as the door sensor. IMU does not have external mechanical parts compared to other switches and is thus durable. However, using the IMU requires us to redesign the system so that the wire can pass from the back of the door to the Raspberry Pi. Also, it will have a higher false alarm rate

when the whole box falls to the ground. Some machine learning algorithms can be used to reduce the false alarm rate, but it exceeds the project's scope.

The third selection uses a photocell inside the box to detect the light when the door is opened. Because of the low cost and small size, it will offer greater flexibility on the mount location over the above sensors. However, the photocell will not work at night and thus is not selected for the project.

The final selection is to use a handle limit switch as the door sensor. Compared to the IMU, the handle limit switch will have a lower false alarm rate because the door has to be intentionally opened by someone to trigger the switch. Because the handle limit switch does not utilize an external magnet for triggering, its lifespan will likely be longer than that of the reed switch.

Power Supply

A 47Wh Voltaic battery pack powers the system. It can power the system for 4 to 5 hours during a power outage.

Alarm Synthesis

The alarm is used when it is impossible to make Twilio phone calls via the Internet. The alarm can be extremely useful when there is a power or network outage. The team built the alarm system so that it can be used to say the words the admin wants the system to say instead of using a simple buzzer. The alarm message is played through the device’s speakers to alert people in the vicinity of an emergency. Google’s text-to-speech engine is used to generate the MP3 file to be played during an emergency.

IO Board Design

Design process

Among the important elements of the Naloxone Safety Kit is the printed circuit card (PCB), which serves as an IO hub, linking different sensing units and audio speakers. The PCB is a crucial element since it functions as the main point of control for the whole device. It is accountable for gathering and processing information from the sensing units, managing the temperature level, and starting the emergency situation reaction procedures in case of an overdose. Although the PCB does not have chips built onto it, all the sensor signals need to be passed by the PCB.

Developing the PCB included a number of actions, consisting of identifying the required parts and connections, laying out the circuitry, and testing the board's performance. To guarantee the PCB's compatibility with the various other elements, the group needed to thoroughly choose the sensors and audio speakers that would certainly be compatible with the device. They likewise needed to think about the power demands of each element and guarantee that the PCB might

provide sufficient power to run everything effectively. Besides, the width of the tracks needed to be appropriate so that the power can be carried safely without increasing the temperature to a dangerous level. Figure 7 is the CAD design of the circuit board.

When the development was completed, the group sent out the PCB design to JLCPCB for prototyping. The PCBs used a process called surface mount technology (SMT), which enabled accurate positioning of components on the board. The group after that evaluated the board to guarantee that it functioned properly and could communicate with the various other elements of the device.

Generally, PCB development was an essential aspect of the Naloxone Safety Kit project. It enabled the group to produce a device that could effectively check temperature level and react rapidly in emergency situations. By thoroughly choosing parts, laying out the circuitry, and screening the board's performance, the group had the ability to produce a reliable and efficient PCB that functioned as the foundation of the whole system. The PCB development procedure likewise showed the team's capability to deal with complicated systems and incorporate several components to produce a working device.

Door Sensor Circuitry

The door sensor circuitry is a vital part of the Naloxone Safety Kit PCB design, as it detects when the kit is opened and starts the emergency response procedure. For this purpose, the team utilized a handle limit sensing unit to identify the door open event. The switch is typically closed when the door is closed. To limit the total power of the sensing unit, the team attached a 1M ohm resistor before the switch and a 100k ohm resistor after the switch.

When the door is closed, the voltage across the sensing unit will be about 0v, suggesting that the switch is closed. However, when the door is opened, the voltage across the sensor will be 3.3V, suggesting that the switch is open. The team then utilized the GPIO pin 15 on the raspberry pi to poll the signal every second. This enables us to identify when the door was opened and set off the automatic 911 call feature of the kit. The door sensing unit circuitry is crucial to the general function of the Naloxone Safety Kit, as it guarantees that emergency services are alerted as promptly as possible in case of an overdose.

In order to guarantee that the connections between the door sensor switches and the PCB board were constantly reliable, the team utilized terminal blocks to create the connections. The terminal blocks offered a secure and user-friendly method for connecting cables to the PCB board, and helped to prevent any unintentional disconnections or damages to the cables. Each switch was linked to a different terminal block, which was then connected to the PCB board using screws. Using terminal blocks also made it simpler to replace or repair the switches if needed, without the need for any specific equipment or tools.

Overall, the door sensing unit circuitry is a relatively straightforward but important component of the Naloxone Safety Kit PCB design. By utilizing a handle limit sensing unit and proper resistors, the team was able to reliably detect when the kit's door was opened and start the emergency response procedure. This guarantees that naloxone is administered as quickly as possible in case of an opioid overdose, possibly saving lives.

Temperature Sensor Circuitry

For the temperature sensor, the team chose to utilize the DHT22 sensor, which is an inexpensive, highly precise temperature and moisture sensing unit. Among the primary benefits of the DHT22 is that it utilizes a one-wire protocol for communication, making it simple to integrate into our circuit design. Furthermore, the DHT22 offers high precision and stability, with a temperature range of -40 to 80 degrees Celsius and an accuracy of +/- 0.5 degrees Celsius.

To guarantee the accuracy of the temperature readings, the team took a number of steps in the circuit design. First of all, the team placed the DHT22 sensor as close as possible to the naloxone storage area. Second of all, the team made sure that the sensor was well safeguarded from any type of outside factors that might affect its readings, such as direct sunshine or air flow. Finally, the team took care to route the sensor's signal trace away from other traces that might introduce interference, such as power or ground traces. By taking these steps, the team was able to guarantee that the temperature readings from the DHT22 sensor were both precise and steady, offering a reliable indicator of the temperature inside the naloxone storage area.

Audio Circuitry

The audio circuitry of the naloxone safety kit is developed to boost the functionality of the alarm system by allowing the audio speaker to say words rather than utilizing a simple buzzer. This function offers more info about the emergency situation and can help bystanders to better understand the situation and take appropriate action.

First Version

In the first attempt of building the audio circuit, the team tried to use the PAM8403D audio amplifier to amplify the audio output from the 3.5mm audio jack of the Raspberry Pi. The team planned to use the amplifier to directly power the speakers so that the user could increase the volume of the alarm. Because the single USB port on the battery can only supply 10 Watts, the team chose to use the other USB port on the battery to directly supply the power to the amplifiers and speakers. However, after the manufacturing, the team found that the SHDN and MUTE pins should be connected to 5V so that the amplifier will output the signal. However, in the design, the team made a mistake and connected these two pins to the GND. Therefore, there is no audio output even though there is power supply to the amplifier. The team tried to fix the problem by disconnecting these two pins from the GND. The team then found that the audio amplifier would only work 50% of the time. The team also found that when the audio level exceeds 50%, there is a clipping of the audio signal. The team believed that the problem is caused by the input signal

that is too strong or has a high peak-to-average ratio. Because of the high cost of manufacturing the PCB, the team decided to use existing speakers from the market instead of using customized amplifier circuits to amplify the alarm sound.

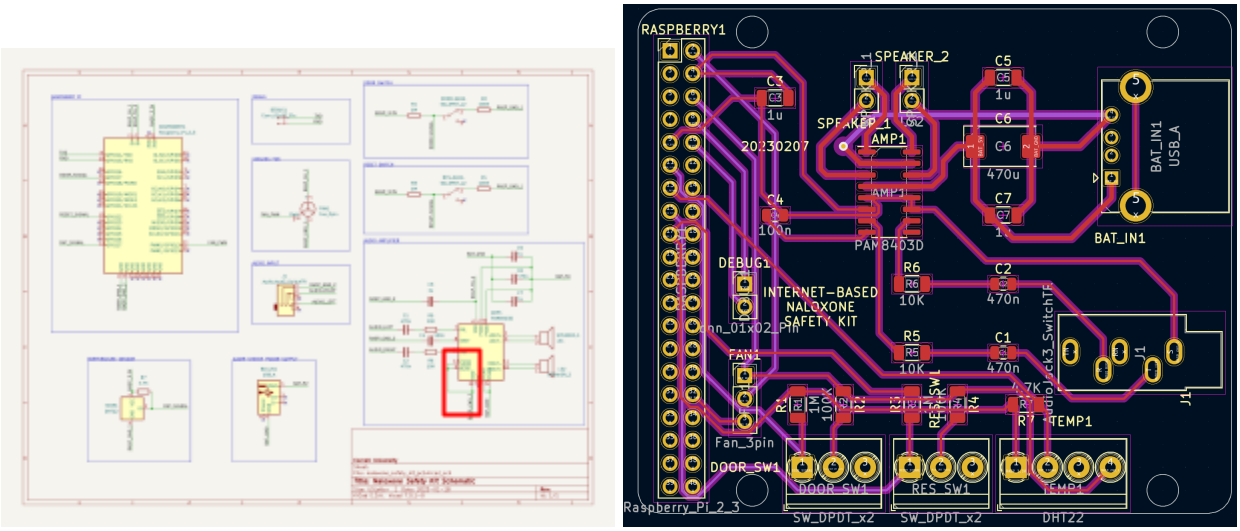

Figure 6: First Version Circuit Design. The two pins in the red rectangle were connected to GND by mistake.

Second Version

After the first attempt, the team created a second PCB design. The team tried to replace the analog amplifier with a digital circuit so that the overall PCB design and testing would be simpler and easier to debug. The revised PCB takes a 5V USB input as the power source and outputs power directly to the USB speaker. A dedicated LED is utilized as a power indicator. The display also takes power from the PCB, and the door switch and temperature are likewise linked to it. The PCB is linked to the Raspberry Pi through a 40-pin port, and there are enough USB ports to support up to 2 USB speakers. Two terminal blocks are also included for the door switch connections. It is possible to use the other terminal block for the wake/sleep switch.

Additionally, the audio signal for the USB speaker comes from the Raspberry Pi's USB port, which is also connected to the PCB. The width of the track for the power of USB speakers is increased to 1 mm to limit temperature rise, and the center of the PCB is cut out to make area for a fan. All the subsystems are put as far as possible to minimize EMI. The team also designed the PCB so that it could be mounted directly above the Raspberry Pi. The team also reserved space to accommodate the cooling fan at the center of the PCB. Additional silkscreen texts were added to the PCB so that the user could connect the wires without referring to the user manual.

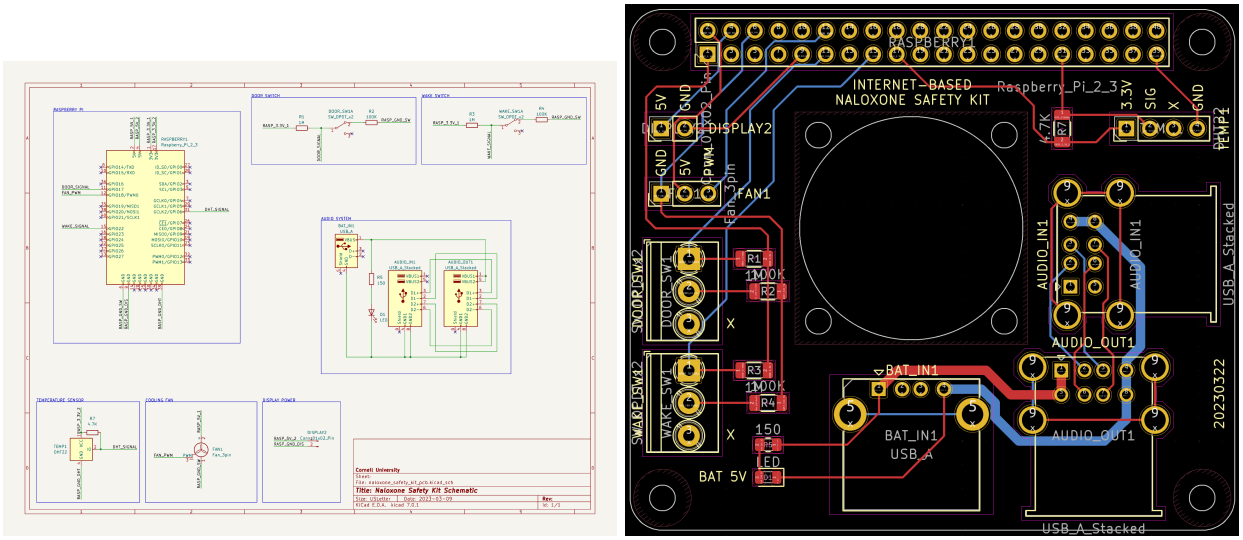

Figure 7: Second Version PCB Design.

PCB Prototyping

For the prototyping of the PCB, the team utilized JLCPCB, a popular PCB production and assembly service. JLCPCB provided an easy-to-use online platform for developing and ordering the PCB, and their inexpensive prices and quick turnaround time (around one week) made them a perfect option for our project. We were able to assemble the PCB with SMT components from their stock, which were almost impossible to do by ourselves, with the JLCPCB service. The team ordered the second order of PCB after finding that the first version had design flaws.

Figure 8: First Version PCB Prototype. The pins in the red rectangle were connected to the GND by mistake.

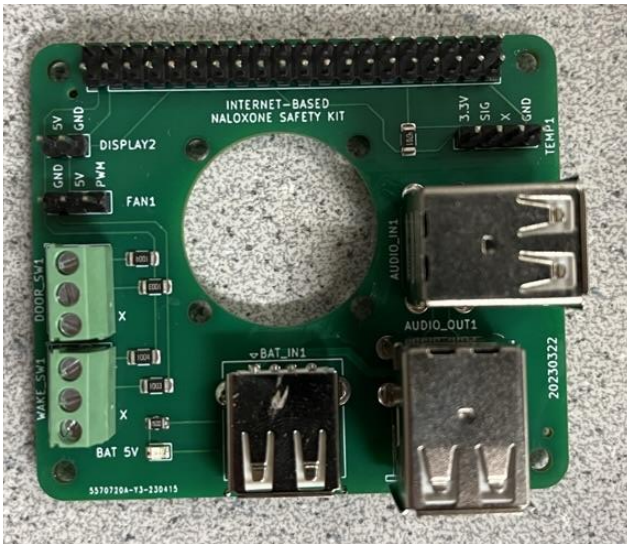

Figure 9: Second Version PCB Prototype.

Software Development and Implementation

Programming Languages and Frameworks

The software for the naloxone safety kit is designed using the Python programming language. Python is a high-level language that is commonly utilized for developing a range of applications because of its simplicity and ease of use. It is an interpreted language, which implies that it can operate on various systems without the need for compilation. Python also provides a rich basic library and various third-party libraries and components that offer extra functionality, making it a perfect option for building complex applications such as the naloxone safety kit.

Along with Python, the software application also utilizes PyQt, a set of Python bindings for the Qt application framework, for building the graphical user interface. Qt is a cross-platform application framework that offers an extensive set of tools and libraries for constructing graphical user interfaces, in addition to for networking, threading, and database connectivity. PyQt enables designers to develop extremely customizable and feature-rich user interfaces that can be quickly incorporated with the application logic. With PyQt, the team had the ability to produce a modern and user-friendly interface that enables individuals to interact with the naloxone safety kit and quickly access its features.

Before transitioning to PyQt, the team also used Tkinter to build the GUI for demonstrating the functions of the device.

Concurrent Programming

The graphical user interface will show all critical readings from the sensors, such as the current temperature and door status, on the touch screen. Meanwhile, we need to check the server connection every ten minutes to ensure the correct alarm method is used when the user opens the door. It will be too complicated to put everything into one single process. Therefore, concurrent programming is needed in the software development process.

The team initially decided to use multi-process design for the software because it was error-proofing. However, after considering the resources on the Raspberry Pi and experimenting with multi-thread design, the team decided to switch to the multi-thread design.

Multi-process Design

Under this design, each process is designed to perform one task, such as reading GPIOs or keeping track of the door switch. A process monitor is also designed to monitor these processes. If any process exits, the process monitor will respawn a new process to continue the task. However, the communication between these processes will be a problem since each process has its separate memory process. A shared memory array is used instead of a message queue to achieve inter-process communication.

The design contains seven processes. They are

1. Process for showing the graphical user interface on the touch display;
2. Process for managing all GPIOs, including temperature sensor, door switch, and fan PWM;
3. Process for calling 911 when needed;
4. Process for monitoring the network connection;
5. Process for synthesizing the alarm;
6. Process for monitoring the naloxone and contact admin for replacement;
7. Process for monitoring all the above processes.

The process monitor records the process ID of the spawned processes when using fork(). A while loop is used to get the return value of waitpid() from these processes. If any of the child processes returned a value indicating a crash, the process monitor will fork the crashed process. By implementing the process monitor, the program gains some degree of immunity to program crash. When the user wants to quit the program by entering CTRL+C, the process monitor will send the same signal to all child processes. A signal handler is also designed to terminate the child process once received the signal. The process monitor will wait for all the child processes to exit before exiting to avoid producing zombie processes. Figure 10 shows the basic flowchart to spawn these processes and their roles in the software.

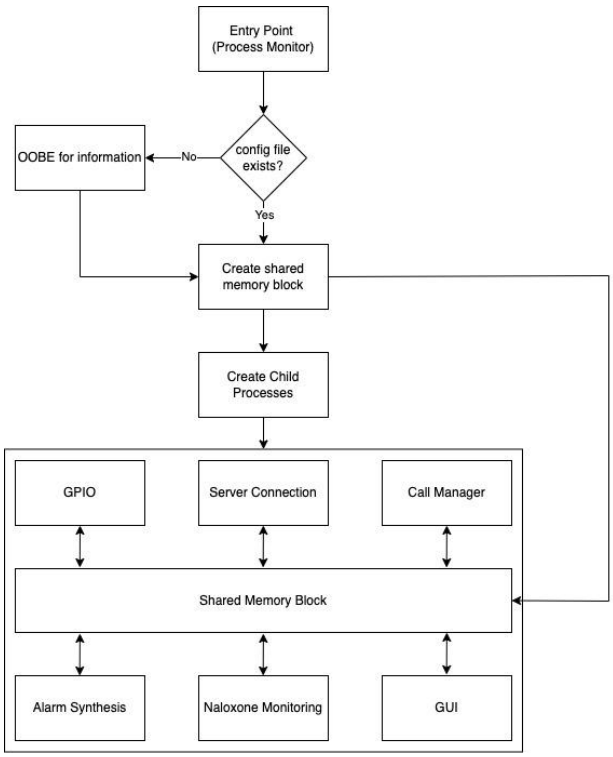

Figure 10: Process-based Software Flow Chart.

Since simultaneous writes to the shared memory block will cause racing conditions, all the processes in the programs share a lock. Each individual process needs to acquire the lock before reading or writing the shared memory block. After the operation has finished, the process will release the lock and let other processes modify the shared memory.

Multi-thread Design

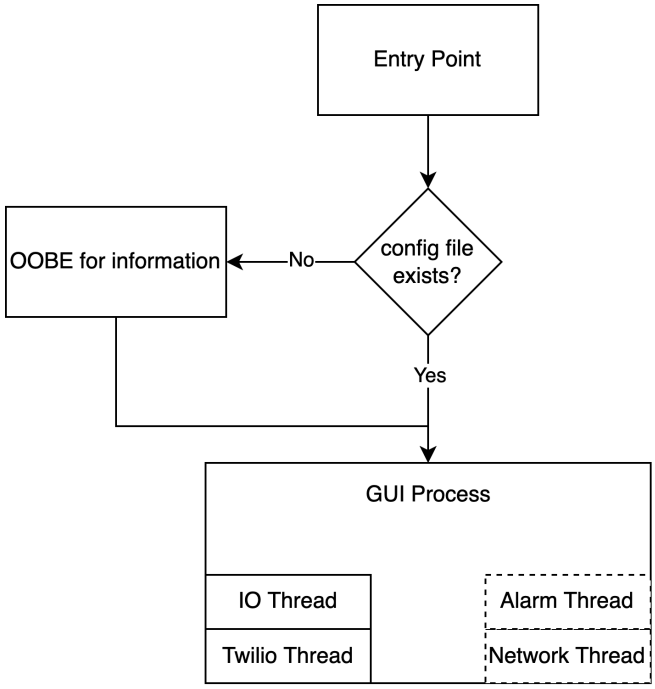

Figure 11: Thread-based Software Block Chart. Not all threads are shown in the figure. The shown threads directly control the behavior of the device.

To avoid crashing the whole program while one sensor returns invalid readings, the first version of the software used a multi-process design so that when the GPIO process crashes, the GUI and other processes will be unaffected. During the debugging and testing, the team found that multi-process software was unnecessary.

1. The multi-process design significantly increases resource consumption. Since each process has its own copy of the memory space, a significant amount of resources were wasted.
2. When the team wanted to read from the GPIO, there would be a delay caused by the process of acquiring the lock. The GPIO process will first try to read the physical component, then try to get a lock on the shared memory block. After getting the lock, the GPIO process can write to the shared memory block. Then, the GUI process will also try to do the same thing. First, it will try to get a lock of the shared memory block, then it can read from the shared memory block. After that, the GPIO thread of the GUI process can emit signals to change the GUI. The whole process takes around 2 to 3 seconds in the

worst-case scenario since both the GPIO process and the GPIO thread of the GUI process are updated on a 1s time interval.

3. The locking mechanism of the shared memory process increased the chance of program freezing. If one process failed to release the lock, the shared memory block will be inaccessible to other processes and the program freezes.
4. It was impossible to update the memory space of the other process instantly after user input. For example, if the user changes the expiration date of Naloxone in the GUI process, the program will first write the new expiration date in the configuration file because the shared memory cannot share QDate() objects directly. After that, the naloxone monitoring process will read from the configuration file and change its behavior. The configuration file is frequently read from the written to, leading to high disk IO. This also caused a time delay between the user input and the GUI update.
5. A long program exiting process is needed to shut down all the processes. In the previous design, the team used a signal handler to signal the program close. However, if the process monitor fails to use waitpid to clean up all the zombie processes, it will contaminate the process table.
6. The system call used on Linux, the os.fork(), is not supported on Windows and macOS platforms. It is hard to debug the software if the team wants to debug the software without using the Raspberry Pi.

Because of the above limitations of the multi-process design, the software was modified to use a multi-thread model. The thread all belongs to the same GUI process. There are several threads in the updated design:

- IOWorker: the thread to interact with the sensors. It also records the naloxone status since it is easier to compare with the max temperature of Naloxone in this thread.
- NetworkWorker: the thread to keep track of the connection with the Twilio server. The user can now view the remaining balance in their account as well.
- TwilioWorker: the thread to handle both call and message service. in_queue is used to send requests and out_queue is used to send notifications to the GUI thread.
- AlarmWorker: the thread to play the alarm.mp3 when the Twilio service is unreachable.

Cross-Thread Communication

Three queues are used for inter-process communication. The request_queue transmits data about the Twilio request information, such as the message and the destination phone number. The status_queue transmits notifications that will be displayed on the GUI. The io_queue allows immediate changes to the io_worker, such as the disarmed and fan threshold temperature. Each thread can also modify the GUI directly by emitting a signal to the GUI thread.

Operating System and Hardware Compatibility

The operating system and hardware compatibility of the naloxone safety kit software were essential factors to consider throughout the development procedure. While the software application can be operated on any type of system, it was mainly developed to operate on the Raspberry Pi, a popular single-board computer system. The Raspberry Pi's reduced power usage, little form factor, and GPIO (General Purpose Input/Output) pins made it a perfect option for the naloxone safety kit. Nevertheless, the software application can be installed on any type of device that satisfies the necessary hardware requirements.

The installation script for the software was particularly designed for Debian-based systems, such as Raspbian, Ubuntu, and Debian itself. Individuals that want to run the software application on other OS will have to set up all the required dependencies themselves. These dependencies are listed in the user manual. The setup directions for these dependencies are readily offered on the internet, making it feasible for individuals to run the software application on a wide variety of devices for testing purposes.

One noteworthy aspect of the software's compatibility is that displays related to GPIO will only work on the Raspberry Pi. This is due to the unique hardware configuration of the Raspberry Pi, that includes GPIO pins that can be utilized to connect sensing units, audio speakers, and other equipment components. The software is developed to communicate with these pins and show info on the display. While other devices might have comparable abilities, they might not use the exact same pinout setup, making it challenging to adjust the software to deal with those devices. However, the user can modify the code in the threads so that it uses APIs on another platform.

User Interface Design

The user interface design was a major challenge for the team because the team had little experience in artistic design before. The team explored different technologies to implement the interface and came up with multiple designs throughout the whole project.

First Version

The first attempt of the UI design focuses on meeting the design requirements. The artistic design was not considered during the design process. The team managed to demonstrate the superiority of the graphical user interface over the mechanical buttons and LED interface after the first attempt. The first version of the UI was created using Python and Tkinter. There were three states in the software, corresponding to three different windows. When the door is opened, the current window would be closed and the new window would be created. Therefore, the screen would momentarily turn blank during state transitions. The first version of the software used the multi-process design for concurrent programming.

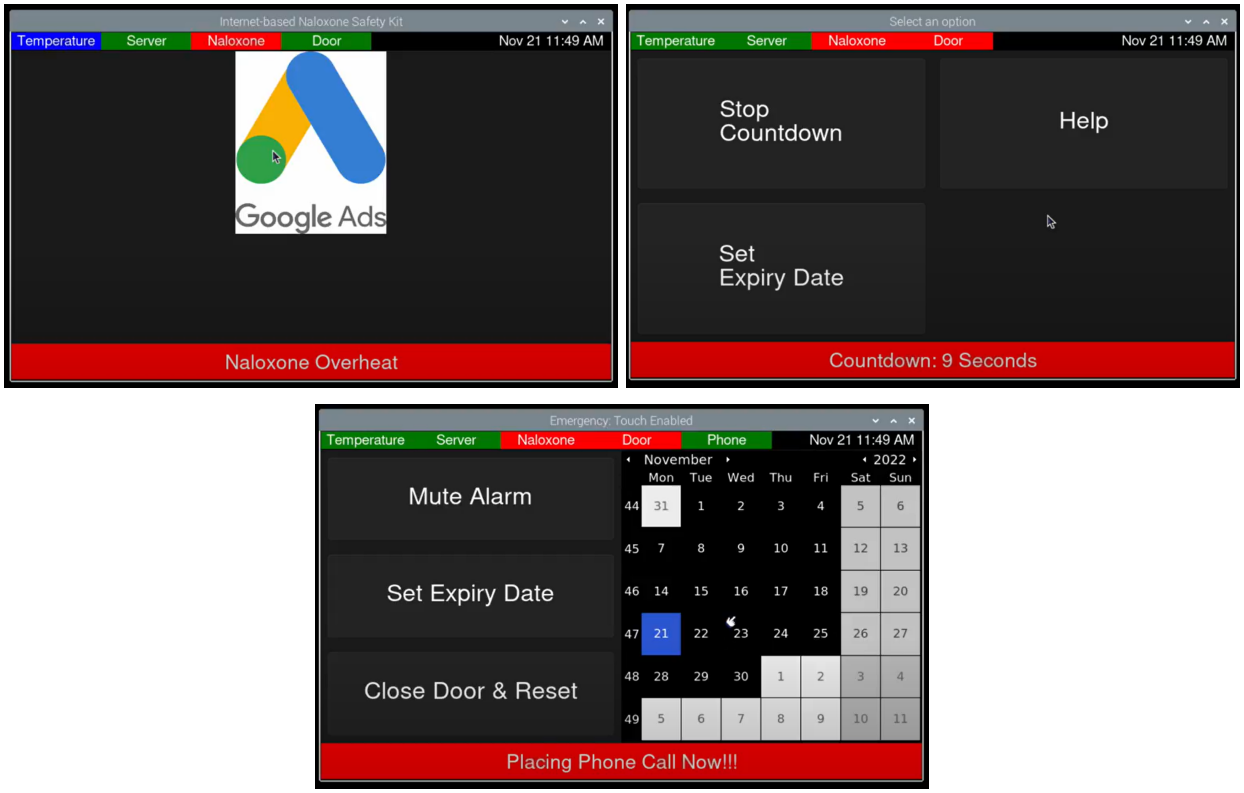

Figure 12: First Version of the UI. The top row shows the indicators of the kit's statuses. The bottom row shows the notifications. The center of the UI can be used to display educational contents or pushbuttons.

Second Version

The second version was developed using another UI technology called PyQt. The team migrated to this technology because it would provide us with some advanced UI features. The team could easily build the UI using Qt Creator by dragging widgets into the window, instead of writing code to generate the widgets. This powerful feature enabled the team to split the frontend and backend completely. PyQt also supported the signal-slot model, which made dynamic user interface possible. It also provided profound control over the style of the widgets by providing Qt Style Sheets (QSS), which could be used to change the shape and color of all widgets using CSS style programming. In this version, the program was able to switch between different layouts using the stacked widget of PyQt so that the software could be more organized and more intuitive for users. A configuration file is also created to save the user settings. This version also eliminated the screen blank between state transitions. This version and all subsequent versions of the software uses the multi-thread design.

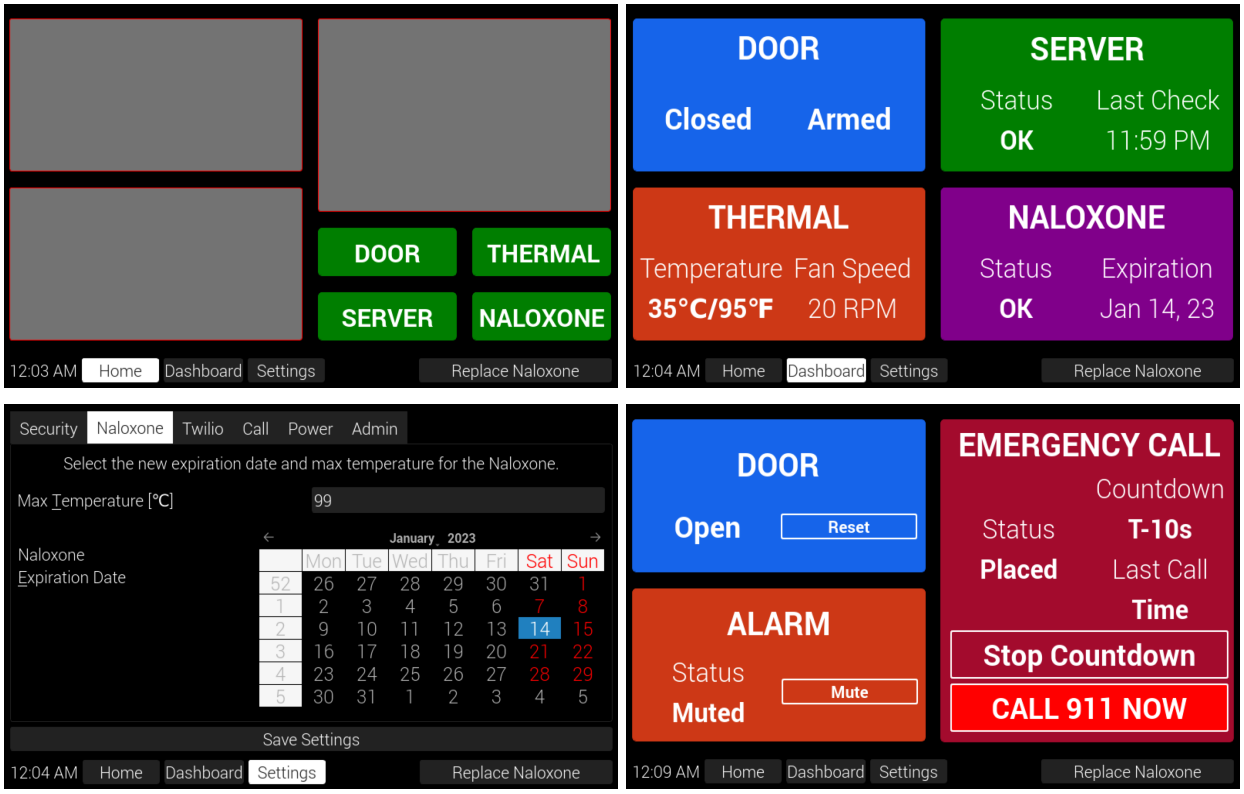

Figure 13: Second Version of the UI.

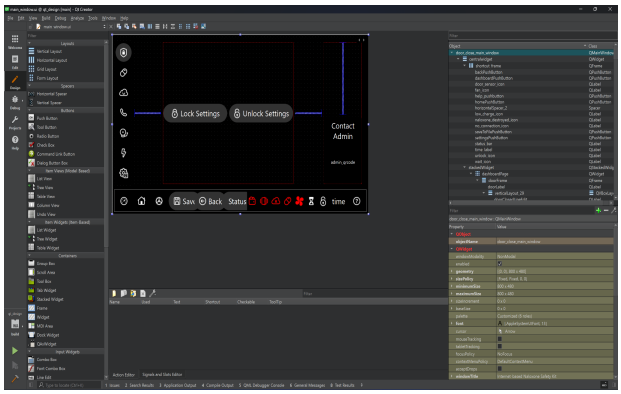

Figure 14: Qt Creator. The team used it to drag widgets directly into the window, which greatly reduced the time and effort to build UI.

Third Version

The third version focused on improving the artistic design of the UI. The team also made buttons and important information more obvious for the users by changing the background color of the buttons and backgrounds dynamically. In this version, we added some functions for the paramedics so that they could replace the expired Naloxone after the administrator granted them access to the settings. They could request the passcode by providing their phone numbers. The administrators could also retrieve the passcode if they forgot it.

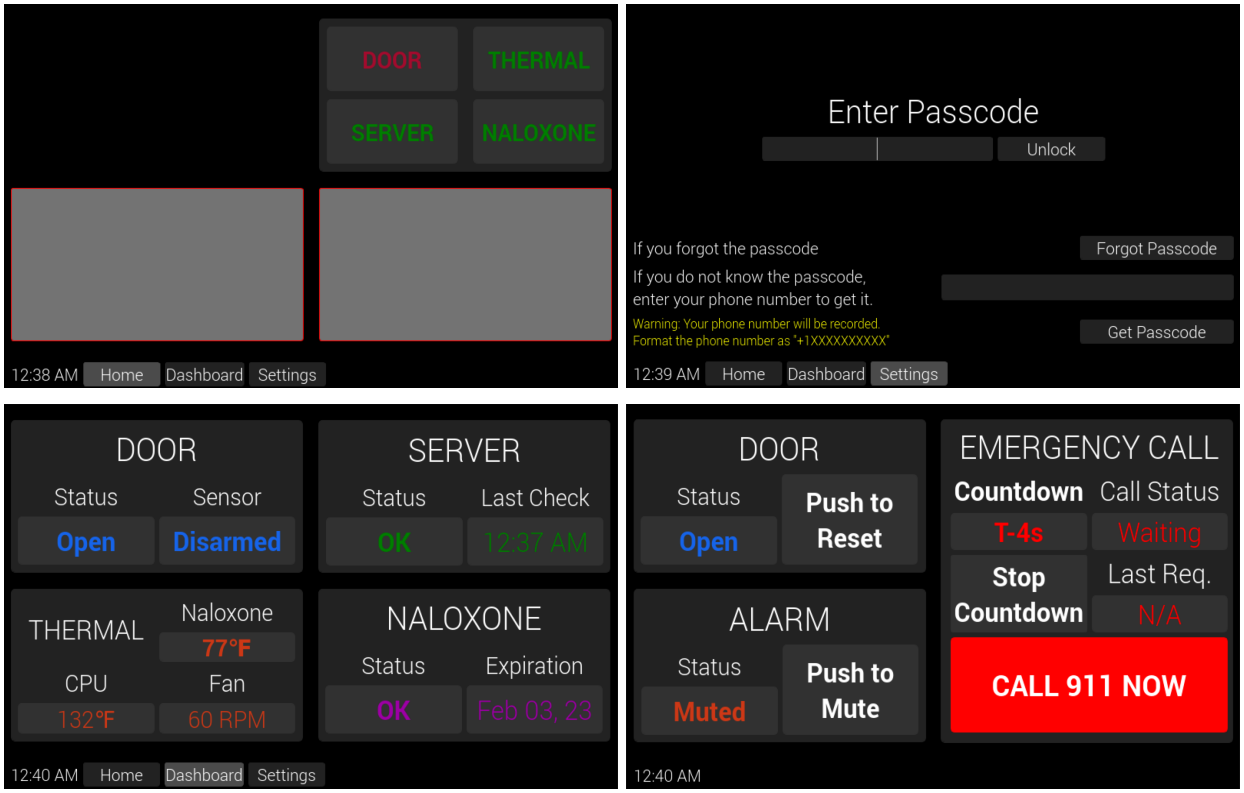

Figure 15: Third Version of the UI.

Final Version

The final version of the design of the software interface for the naloxone safety kit was influenced by the Android Auto infotainment system. The Android Auto platform is known for its user-friendly and easy to use user interface, which provides smooth connectivity and easy accessibility to important functions while reducing driver distraction. Our software design intended to replicate these functions and characteristics, focusing on ease of use and simplicity for individuals that may be experiencing a high-stress situation, such as during an overdose incident. The team included a completely redesigned home screen. Some educational contents could be placed on the home screen. New indicators were introduced to the task bar just like the indicators on the car's dashboard. The team also reduced the visual complexity by using icons, instead of texts, on some buttons. The result is a clean, modern user interface with clear and concise buttons and text, making it easy for individuals to navigate and operate the safety kit's features. By drawing inspiration from the Android Auto platform, we aimed to create an interface that is not only effective and efficient, but also comfortable and familiar to users.

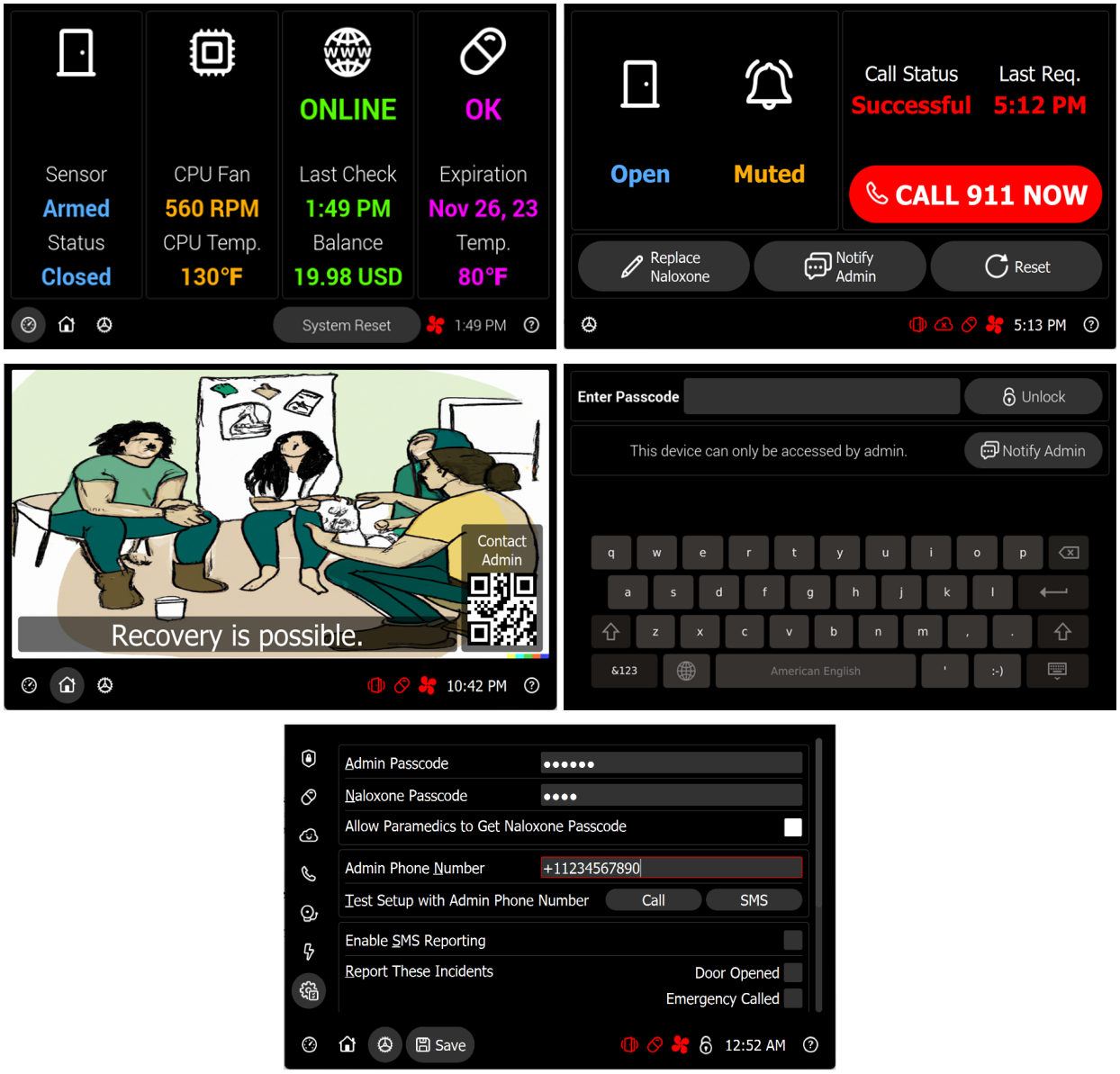

Figure 16: Final UI Design.

Dynamic User Interface

The graphical user interface (GUI) of the software is designed to offer a vibrant and informative experience for the user. The signal and slot model offered by PyQt is utilized to deal with events and update the UI. When a user interacts with a widget, the widget will send a signal and the corresponding slot function is called to perform the needed actions and update the UI appropriately.

Apart from user input, the UI is also updated based upon the system's status, which is constantly monitored by a different thread. For example, the temperature level and door sensor readings are monitored, and whenever there is a change, a signal is sent out to the GUI thread to update the

corresponding widgets. This enables the user to see the current temperature level and door sensor status in real-time.

In summary, the dynamic UI of the software enables ease of interaction and better monitoring of the naloxone safety kit. The signal and slot model allows effective event handling and UI updates, while the real-time display of the system's status offers important information to the user. The mix of these functions makes the software a user-friendly and informative tool for managing the naloxone safety kit.

Data Storage and Management

For data storage and management, a configuration file was used instead of a database to keep user setups. This is a standard practice on Linux systems as individuals can make modifications to these configs themselves utilizing a text editor. The configuration file is only read when loading the settings, and the software will maintain a duplicate of the config in memory to save disk I/O.

In case the config file is missing, the software will prompt the individual for the necessary information before running. This design enables simple personalization and adjustment of the settings, without the need for a complicated database management system. Additionally, it minimizes the general footprint of the software and minimizes the impact on system resources. Overall, the configuration file system offers a reliable and efficient means of data storage and administration for the software.

Security Considerations

Because the device needs to be accessible to the first responders and paramedics who may not know much about the software design of the safety kit, the team decided to use the passcode to protect the settings. The team also made all necessary functions accessible to them without the need for a passcode. The detailed description of the security system is described in the user manual Security.

Testing

Test Plan

The team decided that the safety kit design should follow the design criteria from the beginning to ensure that it meets the requirements. Since it was hard to make changes to the system after it was assembled, the team agreed that the testing should be progressively, which means that each new part of the system should be tested as soon as possible. After assembling the safety kit together, an acceptance test would be conducted to ensure that the device met the overall design requirements.

Embedded System Testing

The software testing was a crucial part of the project. The whole system was essentially controlled by the python software and if the software testing failed to catch bugs, the device would not work properly. Therefore, several phases of testing, such as unit testing, integration testing, and system testing, were involved in testing the software when the software was in development. Because there is a tight integration between software and hardware, the team decides to put the electrical system and software together when performing the testing.

Unit Testing

The software was first developed in different parts depending on the functionalities of the code independently. After testing each module thoroughly, the team moved to the next module. For example, the team first wrote the code for door sensor detection. We ensured that this part of the code could work without a single misdetection for 24 hours to rule out possible bugs in the code. Similarly, the code for detecting Naloxone temperature was also tested by recording the temperature readings into a log file. The team found that sometimes the temperature sensor would not return the correct temperature because the sensor requires strict timing to read and Linux is not a real time OS, a result is not guaranteed to be returned. As a result, the team used the previous readings to fix the error to avoid wrong readings to be recorded and label the Naloxone as destroyed by mistake.

Integration Testing

Integration testing required the team to test various integration between different parts of the software. This testing ensures that when combined, the code as a whole would behave correctly. For example, the team combined the door sensor detection code with the Twilio calling code to ensure that the phone call would be placed when the door was opened. The team also tested that when the CPU temperature sensor reads a temperature exceeding the threshold temperature, the CPU fan would be started correctly.

System Testing

System testing requires the team to test the entire system as a whole to ensure that all the components are working correctly together. The team also simulated real-world scenarios to test the responsiveness of the device. For instance, the team tested a scenario when the device was disconnected from the Internet. In this case, the device would choose the speaker to play the alarm instead of trying to place a Twilio phone call.

Acceptance Testing

After testing the whole system, the team moved the system into the box so that we could test the device against the design requirements and project objectives. The acceptance criteria was established at the beginning of the project. The team designed the software to meet the design criteria to ensure that it met all necessary requirements. After the testing, we found that the device was functioning exceptionally well.

Results

The team prototyped the whole device within 2 semesters and tested the device at the end of the second semester.

Assembled Enclosure

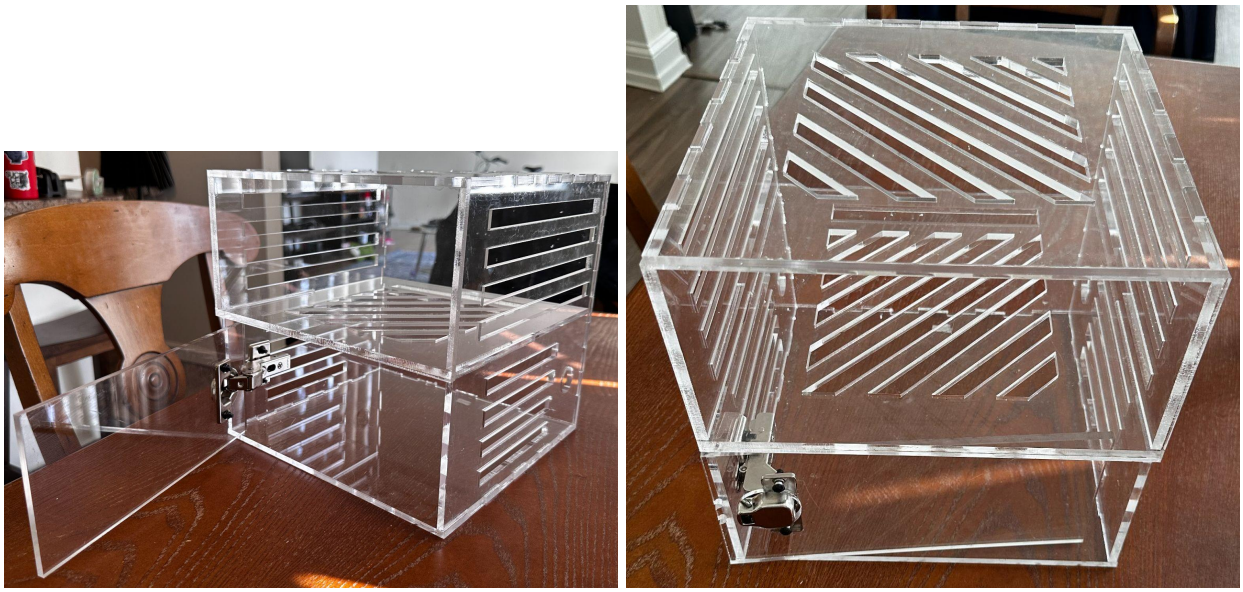

Figure 17: Assembled Enclosure.

Assembled Circuitry

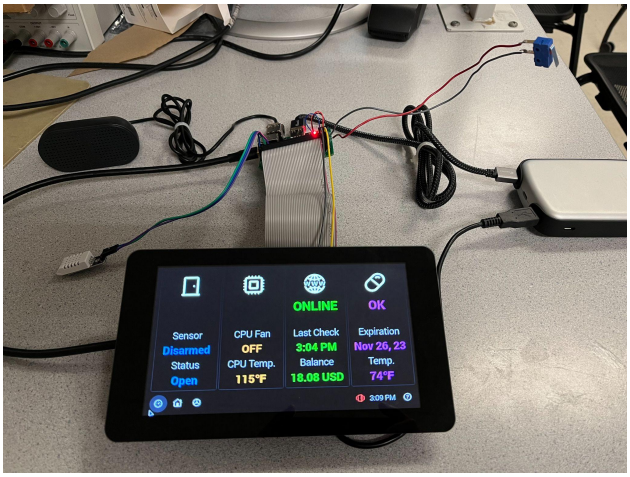

Figure 18: Assembled Circuitry.

Assembled Safety Kit

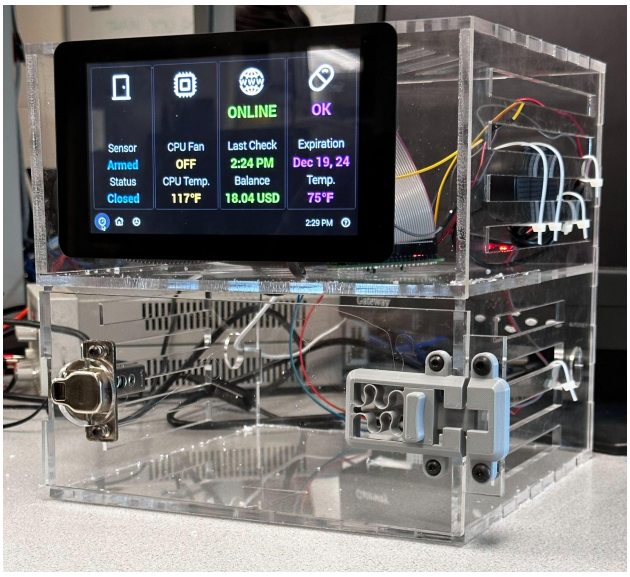

Figure 19: Safety Kit.

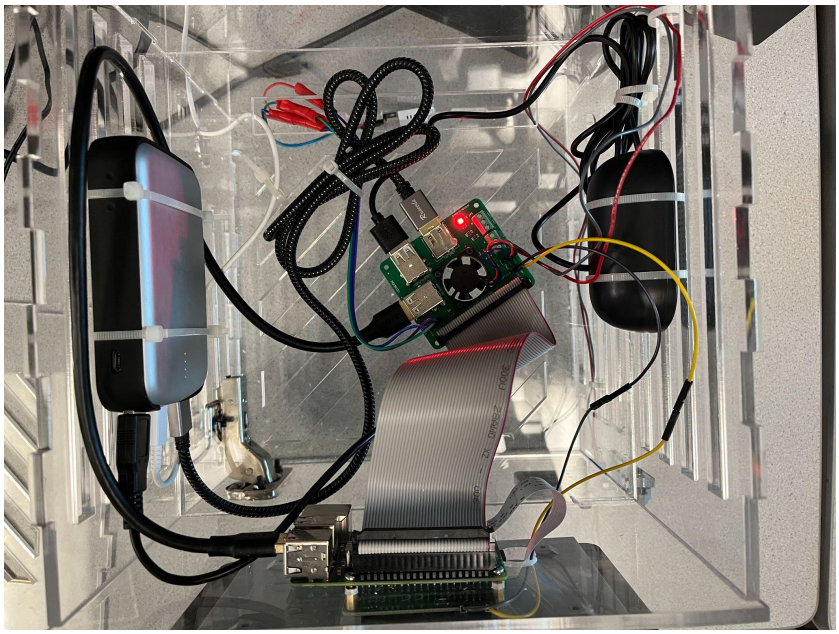

Figure 20: Safety Kit Circuitry.

Conclusion

Summary of project objectives and outcomes

The main objective of the Internet-based Naloxone Safety Kit project was to design and prototype a device that could help prevent opioid overdoses. The device was designed to provide a temperature-controlled storage space for storing the Naloxone. In addition, the device could sound an alarm or call emergency services in event of an overdose depending on the network connections.

The outcomes of the project include the design and a prototype of the device. In addition to the safety kit itself, the team also produced documentation detailing how it was made as well as the source code for the software. This document will provide step-by-step instructions on how the team made the safety kit. The goal of this document will be to make the design reproducible so others can deploy safety kits in their communities. A detailed user manual was also provided.

Discussion of limitations and areas for improvement

One limitation of the device is the reliance on Raspberry Pi as the main hub for the device. Although Raspberry Pi could provide all the features we needed, it is a relatively expensive device compared to other microcontrollers. At the time of writing this report, there was a supply shortage of Raspberry Pi on the market. Future work could explore the use of more cost-effective microcontrollers, such as Arduino IoT 33 or Raspberry Pi Pico to reduce the manufacturing cost of the device.

Another room for improvement is in the audio subsystem of the device. The team used USB speakers because of the failed attempt to use amplifiers. Although the USB speakers could provide adequate sound quality, it would not be loud enough in certain places like supermarkets or schools. Future work could explore the use of more powerful audio components such as more powerful audio amplifiers to improve the sound output volume.

Reflection on the design process and lessons learned

One thing the team learned was the importance of thorough testing of verification of the system. In the early stages of the project, the amplifier circuit was not properly tested, which resulted in a malfunctioned audio circuit. If the team could test the circuit before manufacturing, the problem could be avoided. To address this, the team developed a comprehensive testing plan when designing the new PCB and tested the components before sending the design to the manufacturer.

Another thing the team learned was the importance of communication and collaboration between team members. The project involved multiple disciplines, including electrical and computer engineering, software engineering, and mechanical engineering. Effective communication and

collaboration are necessary to ensure that all aspects of the design are integrated properly. This was achieved through regular project meetings and clear documentation of the design process.

A third thing the team learned was the importance of considering the user experience and requirements during the design process. The team originally planned to use mechanical switches and LED indicators for user interaction. However, after considering that the end-users were individuals who were handling overdose victims, the team decided to simplify the user interaction by including an intuitive graphical user interface. The team also simplifies the process of making emergency phone calls by including a simple countdown timer instead of complex combinations of switches.

Contribution to the field of study and potential for further research

The Internet-based Naloxone Safety Kit project presented a new way of reducing the deaths related to opioid overdose. By combining hardware and software technologies, the project created a reliable, user-friendly, and inexpensive solution for storing naloxone and alerting paramedics in emergency situations.

There is a huge potential for further research in this field. One area for further research is to explore ways to utilize machine learning and artificial intelligence in placing emergency phone calls. By doing so, the paramedics could get more relevant information about the location and nature of the incidents and respond to the incident more effectively. Another area for further research is to investigate the effectiveness of the safety kit in a real-world scenario. The team did not have a chance to conduct investigations to learn about this information. Further investigation could focus on user testing and obtaining feedback from individuals, such as paramedics or volunteers, who have experience with naloxone and drug overdose prevention. They could help refine the design and improve the effectiveness of the device.

Acknowledgement

We would like to express our deepest gratitude to our advisor, Dr. V. Hunter Adams, for his advice and support throughout the project. His vital input helped shape the direction of our project and ensured its successful completion.

We would also like to thank the faculties of the ECE department at Cornell University for their assistance and resources, which were instrumental in the completion of this project.

We extend our sincere appreciation to the technical team at the Cornell Maker Club and Cornell Rapid Prototyping Lab for their help with equipment and technical support.

References

Adafruit Industries. (n.d.). *Medium 16x32 RGB LED matrix panel - 6mm Pitch*. Adafruit Industries. Retrieved December 5, 2022, from https://www.adafruit.com/product/420

Geerling, J. (n.d.). *Power Consumption Benchmarks*. Raspberry Pi Dramble. Retrieved December 5, 2022, from https://www.pidramble.com/wiki/benchmarks/power-consumption

Appendix A: 3D Model

See Google Drive.
https://drive.google.com/drive/folders/1c8pj9ZNWs1HWwNE0ZuqJEyI0o524ndgp?usp=share_link

Appendix B: Source Code

See the GitHub.
zyl120/Naloxone_Safety_Kit: Repository for Meng ECE design project (github.com)

Appendix C: Bill of Materials

Name	Link	Number	Unit Price	Total Price
Total				**$275.04**
Raspberry Pi 4	https://www.raspberrypi.com/products/raspberry-pi-4-model-b/	1	$35.00	$35.00
DHT22 Temperature Sensor	https://www.amazon.com/Temperature-Humidity-Raspberry-Electronic-Practice/dp/B07WP4VZTH/ref=sr_1_3?crid=4GWFSKWDEYGO&keywords=dht22&qid=1666497823&qu=eyJxc2MiOiI0LjAxIiwicXNhIjoiMy45OSIsInFzcCI6IjMuODkifQ%3D%3D&sprefix=%2Caps%2C71&sr=8-3	1	$6.35	$6.35
V50 USB Battery Pack	https://voltaicsystems.com/v50/	1	$65	$65.00
Official Raspberry Pi 4 Case Fan and Heatsink	https://www.adafruit.com/product/4794#technical-details	1	$5	$5.00
Handle Limit Switch	https://www.amazon.com/Taiss-Sensitive-Switches-Million-KW11-3Z-1/dp/B074868HZV?ref_=ast_sto_dp&th=1	1	$0.80	$0.80
H002 USB Speaker Set	https://a.co/d/3j4fGV7	1	$12.99	$12.99

Acrylic Adhesive and Applicator	https://www.amazon.com/Weld-Acrylic-Adhesive-Applicator-Bottle/dp/B0096TWKCW/ref=asc_df_B0096TWKCW/?tag=hyprod-20&linkCode=df0&hvadid=312719382368&hvpos=&hvnetw=g&hvrand=8586199928905648395&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9005779&hvtargid=pla-404766667839&psc=1&tag=&ref=&adgrpid=63696814698&hvpone=&hvptwo=&hvadid=312719382368&hvpos=&hvnetw=g&hvrand=8586199928905648395&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9005779&hvtargid=pla-404766667839	1	$26.50	$26.50
Raspberry Pi Touch Display	https://www.sparkfun.com/products/13733?src=raspberrypi	1	$60.00	$60.00
Half-Mortise Concealed Cabinet Hinge	https://www.mcmaster.com/products/half-mortise-mount-hinges/hinge-type~concealed/	1	$3.87	$3.87
Acryllic Sheets	https://www.estreetplastics.com/1-4-x-12-x-12-Clear-Acrylic-Plexiglass-Sheet-p/1002501212.htm?CartID=1	7	$5.59	$39.13
PCB Prototyping	https://jlcpcb.com/	1	$20.40	$20.40

Table 1: Bill of Materials.

Appendix D: Project Timeline

Task	Projected Completion Date
Write demo code to simulate safety kit functionality	9/30/2022
Generate state machine for temperature control	10/24/2022
Create initial CAD model	10/25/2022
Identify necessary supplies	10/28/2022
Breadboard implementation of circuit	11/14/2022
Finish initial circuit design	11/21/2022
Finish initial software design	12/5/2022
Finish initial CAD model	12/5/2022
Finalize any necessary changes to CAD model, circuit, or software	2/20/2023
First PCB Prototyping	2/24/2023
Second PCB Prototyping	4/14/2023
Final print of safety kit	4/25/2023
Poster finished	4/24/2023
Poster printed	4/25/2023
Add circuitry to safety kit	4/27/2023
Poster session	5/2/2023
Final report draft completed	5/12/2023
Final report submission	5/19/2023

Table 2: Project Timeline.

Appendix E: User Manual - Full Version

Introduction

The United States is currently facing an opioid epidemic, with an increasing number of people dying each year because of opioid overdoses. Naloxone is a life-saving medicine that can be administered to overdose victims, however it takes some time for emergency services to show up after 911 has been called, which can frequently be too late. This has led to a need for publicly accessible naloxone to be made available, which is where the proposal for a naloxone safety kit comes in.

We intended to develop and prototype a safety kit that aims to offer safe, reliable, and temperature-controlled storage space of publicly available naloxone, which, when opened, will automatically call 911 and request medical emergency services. The kit might help reduce fatalities due to opioid overdoses by lowering the time it takes for emergency services to reach overdose victims.

Traditionally, naloxone is provided through intramuscular shots by professionals. However, in 2015, the first naloxone nasal spray, the Narcan Nasal Spray, was authorized by the FDA, making it simpler for any individual to offer first aid in case of an overdose. Some organizations, like NaloxBox, have introduced naloxone safety kits to the marketplace after the authorization of the nasal spray. However, these kits cannot call emergency services automatically when opened, rather triggering a siren. Furthermore, their cost is too expensive to be deployed massively, with each kit costing about $250 to $300 without naloxone.

An Internet-connected naloxone safety kit will solve these issues by automatically calling 911 when opened to report the place of the incident. Such a device will have life saving potential in places where recreational drugs are consumed, and the enclosure will be designed to be an artistic reminder of the opioid epidemic. The kit aims to decrease the cost of the entire safety kit while accomplishing these enhancements.

User Roles and Permissions

Admin

The administrator has access to the program and manages all the settings. They are in charge of constructing, putting in naloxone, and examining the safety kit that the software runs on. While anybody can use naloxone in a crisis by opening the lock of the unit, the administrator and the building manager are liable for supplanting the naloxone after it has been used. The paramedics can also replace the naloxone if the admin permits that. The admin should give the naloxone passcode instead of the admin passcode to the building manager.

Building Manager

The building manager can replace the naloxone and enter the new expiration date and maximum temperature for the replaced naloxone. The building manager can only change the naloxone information settings and have limited access to the software's settings. The building manager should have knowledge of using the software and proper response to drug overdose. The manager should also help the paramedics resetting the device when needed.

Paramedics

After the safety kit dials 911, the patient must be taken to the hospital by the paramedics. By pressing a button on the safety kit, they can contact the administrator for assistance if they require additional information. After receiving their phone numbers, the administrator will call the paramedics to provide additional support. If admin permits, the paramedics can replace the naloxone inside the safety kit as well as making changes to the naloxone information. The paramedics do not have access to other settings of the device.

First Responder

When dealing with a drug overdose, the first responder is absolutely necessary. They are the ones who first notice the overdose and need to act quickly to give the patient naloxone. Unlike the administrator and building manager, the first responder does not have access to the software's settings. However, they can use naloxone by opening the safety kit door. Additionally, the safety kit will automatically dial 911 on their behalf in an emergency. In critical situations, the quick actions of the first responder can help save lives.

Getting Started

Source Code

The source code for the software is offered at https://github.com/zyl120/Naloxone_Safety_Kit and is open-sourced under the GNU Lesser General Public License (LGPL) version 3. This implies the software can be freely distributed, modified, and used for commercial and non-commercial purposes. Individuals are encouraged to review the terms of the LGPL V3 license before using or customizing the software. Furthermore, individuals could contribute to the continuous development of the software by sending bug reports, function requests, or code improvements to the project's GitHub repository.

Prerequisites

- Raspberry Pi 3B+ or 4 with latest Raspberry Pi OS Full (32-bit);
- Physical Keyboard and Mouse;
- Monitor;
- Stable Internet Connection.

Change Raspberry Pi Settings

- Disable fan control in the Raspberry Pi Preferences. The software to be installed will control the fan.
    a. Click on the Raspberry Pi icon in the top left corner and select Preferences then Raspberry Pi Configuration.
    b. Select the Performance tab.
    c. Next to Fan, click Disabled.
- Disable Screen Blank in the Raspberry Preferences. This will prevent the screen from going to sleep.
    a. Click on the Raspberry Pi icon in the top left corner and select Preferences then Raspberry Pi Configuration.
    b. Select the Display tab.
    c. At the bottom of the short list is Screen Blanking, click Disable.
- Set the audio output to the USB speaker if needed. You can do it by running the following command in the terminal:

    sudo raspi-config

    and then go to the System > Audio to select the USB speaker as the output.

Installation

Depending upon your choice and level of technical expertise, there are two methods to set up the software. The first choice is to use a Python script that automates the setup and configuration procedure with default settings. Individuals with a basic understanding of Python and a preference for a quicker setup should use this method. The second choice is to manually set up the software by following the user manual's detailed instructions. Individuals that have a deeper understanding of system administration and desire more control over the setup procedure should use this method. The device will set up the software in a completely functional way using either method. The first step, regardless of which one you choose, is to use the following command to download the source code:

git clone https://github.com/zyl120/Naloxone_Safety_Kit

Automatic Installation using Script

To install the software, you can use the provided Python script. Please follow these steps:

1. Run the script as a super user by running the command
   sudo python /home/username/Naloxone_Safety_Kit/scripts/install.py
2. When prompted, provide your Linux user name.
3. During the installation process, the device will download and install several additional packages using apt and Python packages, so it is recommended to have a stable internet connection so that the software can be downloaded.
4. The qt virtual keyboard component will take around 30 minutes to compile, and the compiled file will be located at */home/username/qtvirtualkeyboard*.
5. After the installation process is complete, it is recommended to reboot your system.

You will need to run the script as a super user. If you experience issues during the installation process, refer to the troubleshooting section of the user manual for help.

Manual Installation

1. Before installing the dependencies, refresh the package list to get the latest package and dependency information from the server. Open the terminal window and enter the following command:
   sudo apt-get update
   If the output shows that some packages can be upgraded, you can enter the following command in the terminal window to upgrade them:
   sudo apt-get upgrade
   After upgrading the packages, reboot the Raspberry Pi for the changes to be applied.
2. Install the following packages using apt:
   - matchbox-keyboard
   - libgpiod2

- python3-pyqt5
- mpg123
- git
- build-essential
- qtdeclarative5-dev
- libqt5svg5-dev
- qtbase5-private-dev
- qml-module-qtquick-controls2
- qml-module-qtquick-controls
- qml-module-qt-labs-folderlistmodel
- libxcb-composite0-dev
- libxcb-cursor-dev
- libxcb-damage0-dev
- libxcb-dpms0-dev
- libxcb-dri2-0-dev
- libxcb-dri3-dev
- libxcb-ewmh-dev
- libxcb-glx0-dev
- libxcb-icccm4-dev
- libxcb-image0-dev
- libxcb-imdkit-dev
- libxcb-keysyms1-dev
- libxcb-present-dev
- libxcb-randr0-dev
- libxcb-record0-dev
- ibxcb-render-util0-dev
- libxcb-render0-dev
- libxcb-res0-dev
- libxcb-screensaver0-dev
- libxcb-shape0-dev
- libxcb-shm0-dev
- libxcb-sync-dev
- libxcb-util-dev
- libxcb-util0-dev
- libxcb-xf86dri0-dev
- libxcb-xfixes0-dev
- libxcb-xinerama0-dev
- libxcb-xinput-dev
- libxcb-xkb-dev
- libxcb-xrm-dev

- libxcb-xtest0-dev
- libxcb-xv0-dev
- libxcb-xvmc0-dev
- libxcb1-dev
- libx11-xcb-dev
- libglu1-mesa-dev
- libxrender-dev
- libxi-dev
- libxkbcommon-dev
- libxkbcommon-x11-dev

3. Install the following packages using pip3:
   - twilio
   - qrcode
   - adafruit-python-shell
   - adafruit-circuitpython-dht
   - gtts
   - phonenumbers
   - rpi-backlight
4. If you do not want to use a virtual keyboard for the touch screen, you can ignore these steps and just copy the edited "safety.conf" to the directory of the GUI script. Otherwise, go to step 5.
5. Since the PyQt5 will load the module from the prefix path, we need to determine the prefix path using the following command:

   python -c "from PyQt5.QtCore import QLibraryInfo; print('QT_PREFIX_PATH:', QLibraryInfo.location(QLibraryInfo.PrefixPath))"

   - In most cases, the prefix path is at */usr*. However, if your prefix path is not at */usr*, you need to adjust the command in later steps.
   - We define *QT_PREFIX_PATH* as */usr* in later steps. You should adjust the path accordingly before executing the command.
6. You need to download the latest LTS source code of the Qt virtual keyboard, run the following command:

   git clone -b 5.15 https://github.com/qt/qtvirtualkeyboard.git
7. Open the qtvirtualkeyboard folder. By default, we download the source code to a directory called qtvirtualkeyboard in your home directory. To open that directory, run the following command:

   cd ~/qtvirtualkeyboard

   - You can verify the current working directory by using running the following command:

     pwd

- You are expected to see the output of the above command as /home/USER_NAME/qtvirtualkeyboard, where USER_NAME is the name of the account.

8. Run qmake. To generate the makefile automatically, run the following command:

qmake

9. Then you can use the below command to compile the Qt virtual keyboard automatically. This will take around 15 minutes on Raspberry Pi.

sudo make

10. By running the below command, the binary files will be moved to the appropriate locations on the system.

sudo make install

- By default, the destination is at ~\qtvirtualkeyboard

11. Copy libQt5VirtualKeyboard.so.5.
    - In the terminal window, enter the following command:

sudo cp -L ~/qtvirtualkeyboard/lib/libQt5VirtualKeyboard.so.5 QT_PREFIX_PATH/lib/libQt5VirtualKeyboard.so.5

    - If your QT_PREFIX_PATH is /usr, the command will be

sudo cp -L ~/qtvirtualkeyboard/lib/libQt5VirtualKeyboard.so.5 /usr/lib/libQt5VirtualKeyboard.so.5

    - If your compiled version is 5.15.8, you need to adjust the command to match the version number.

12. You need to create the folder *QT_PREFIX_PATH/plugins/platforminputcontexts* using the following command:

sudo mkdir QT_PREFIX_PATH/plugins
sudo mkdir QT_PREFIX_PATH/plugins/platforminputcontexts

    - Again, you should replace the path with the QT_PREFIX_PATH on your system.

13. You need to copy the *libqtvirtualkeyboardplugin.so* to *QT_PREFIX_PATH/plugins/platforminputcontexts* using the following command:

sudo cp ~/qtvirtualkeyboard/plugins/platforminputcontexts/libqtvirtualkeyboardplugin.so QT_PREFIX_PATH/plugins/platforminputcontexts/

14. You need to cop the whole virtualkeyboard folder to QT_PREFIX_PATH/plugins using the following command:

sudo cp -r ~/qtvirtualkeyboard/plugins/virtualkeyboard/ QT_PREFIX_PATH/plugins/

15. Copy virtual keyboard Qml folder. You will also need to copy the Qml's virtualkeyboard folder to *QT_PREFIX_PATH/qml/QtQuick* folder. But first, you need to create the destination folder using the command:

sudo mkdir QT_PREFIX_PATH/qml
sudo mkdir QT_PREFIX_PATH/qml/QtQuick

16. Then, you can copy the whole folder using the following command:

sudo cp -r ~/qtvirtualkeyboard/qml/QtQuick/VirtualKeyboard/ QT_PREFIX_PATH/qml/QtQuick/

Start the Software

A virtual keyboard has been installed on your Raspberry Pi. You can find it in the start menu under Accessories. After finishing the installation, first change the working directory to your home directory by running

cd ~/Nalxone_Safety_Kit/main

Then enter the following command to start the program:

python main.py

You should see the GUI start up. It will search for "safety_kit.conf" to read configurations. If the file does not exist, it will ask the user to enter the necessary information. A "safety_kit.conf" template is provided in the appendix. You can create the file by yourself so that you don't need to type the information on the touch screen again.

Initial Setup

When you first start the software, you will need to enter necessary information. You will see an "Initial Setup" notification in the taskbar. Please refer to Settings for more details on how to enter the correct information.

Autostart

The user can set the software to start automatically when booting the device. The function is utilized by using the Systemd unit file. To enable this function, follow the instructions below:

1. Create the unit file by entering the following command in the terminal:

   sudo nano /lib/systemd/system/kit.service

2. In the nano text editor window, type the following text:

   [Unit]
   Description=Safety Kit
   After=multi-user.target

   [Service]
   User=username
   Group=usergroup
   Environment=DISPLAY=:0
   Environment=XAUTHORITY=/home/username/.Xauthority
   Environment=XDG_RUNTIME_DIR=/run/user/1000
   Environment=QT_QPA_PLATFORMTHEME=qt5ct
   WorkingDirectory=/home/username/Documents/Naloxone_Safety_Kit/main
   Restart=on-failure
   RestartSec=5s
   ExecStart=/usr/bin/python3 main.py

[Install]
WantedBy=graphical.target

- Remember to replace the *username* with your user name. Also remember to replace the *usergroup* with your user group. By default, the user group has the same name as the user name.
- For the environment variables in the unit file, you can view the current environment variable by typing the command

  env

  in the terminal. Double check that the environment variables in the unit file are the same as the current environment variables on your system.
- Double check the *WorkingDirectory*. It should be the absolute path to the *main.py*.
- The unit file is configured so that it will restart the software if it fails after waiting for 5 seconds. To disable the auto restart function, comment the line
  Restart=on-failure
  RestartSec=5s
  by adding # to the front of the two lines.
- Close the nano text editor by using Ctrl+x. Remember to save the file.

3. You will then enable the service so that it will be automatically started by systemd after reboot. To enable the service, run the following command in the terminal:

   sudo systemctl enable kit
4. You can also start the service now to check whether the unit file is written correctly. To start the service, using the command:

   sudo systemctl start kit
5. If nothing happens after the above command, you can check the status of the service by using the command:

   sudo systemctl status kit
6. Now, the software is configured to run automatically at boot. To disable this feature, you can run the command to disable the service:

   sudo systemctl disable kit

User Interface

Taskbar

The taskbar, which displays the time as well as pushbuttons, indicators, and notifications, is an essential component of the user interface. It can always be found on the screen's bottom. It is intended to adapt to the safety kit's current state dynamically. Indicators for the safety kit's status include whether the CPU fan is on and whether the naloxone has been destroyed. Users can quickly check the safety kit's status and take the necessary action by monitoring the taskbar.

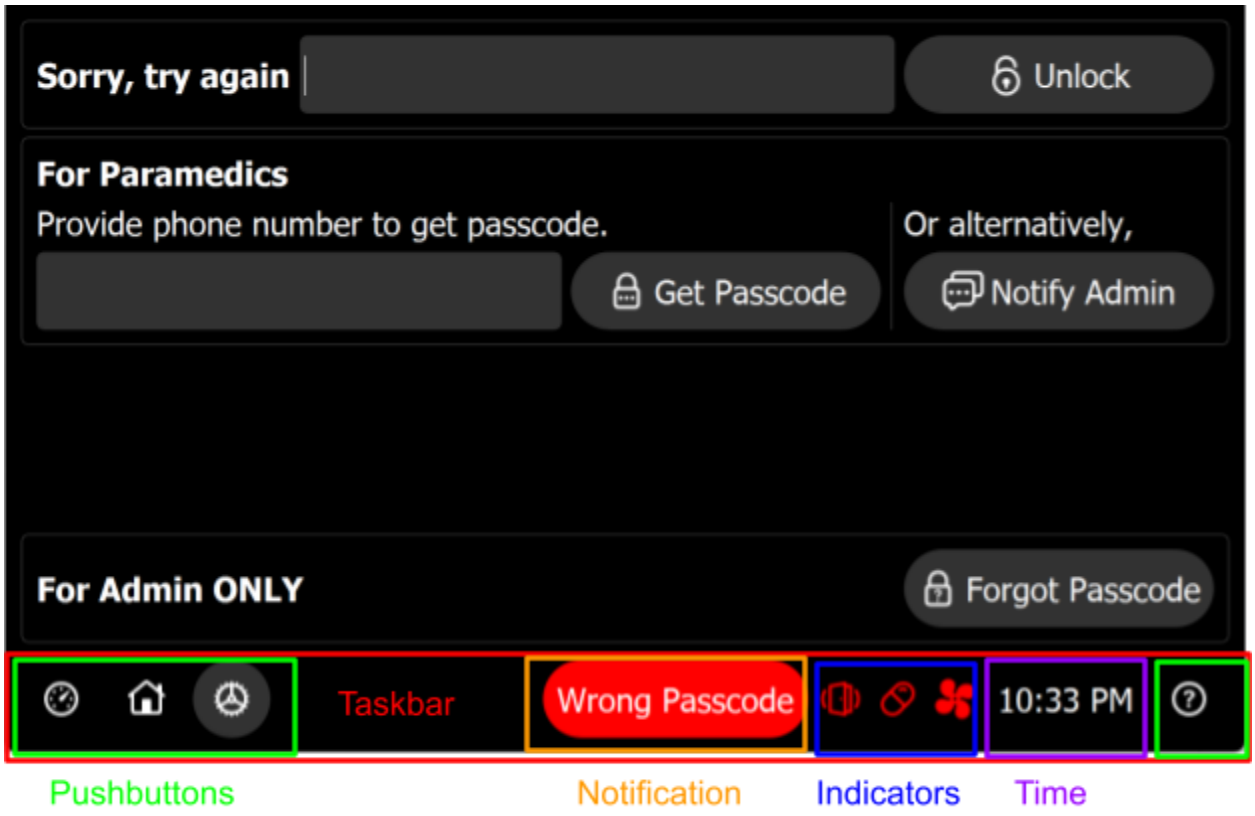

Figure UM1: Taskbar Location.

Pushbuttons

The taskbar's pushbuttons are an essential user interface component that enables fundamental user interactions. To access various software pages, users can click on the push buttons. For instance, the software's settings can be accessed and modified by pressing the "Settings" push button. The software's detailed usage instructions can also be found by pressing the "Help" button. It is essential to keep in mind that although the pushbuttons' availability may change depending on the device's state, they will always offer users an easy way to navigate the software.

Pushbutton	Action

	Go to the home page.
	Go to the dashboard page.
	Go to the settings page.
	Show help for the current page.
	Save the current settings to “safety_kit.conf”. The settings will be applied immediately after saving. You will only see this push button when you are on the settings page.
	Go back to the door open page. You will only see this pushbutton when you access the settings from the door open page.

Table UM1: Taskbar Pushbuttons.

Indicators

Similar to the dashboard of a car, indicators on the taskbar display the device's current state. The indicators may show the state of the door sensor, the network status, or whether the Twilio account has sufficient balance. Users can quickly and easily assess the situation and take the necessary actions thanks to these indicators, which provide essential information about the device's status at a glance.

Indicator	Meaning
	The settings have been unlocked.
	More than one Twilio request is in the request queue.
	The CPU fan has been turned on to prevent CPU overheating.
	The stored naloxone has been destroyed due to expiration or overheating.
	The device cannot connect to the Twilio service.

	The door sensor is disarmed. Emergency service will not be called when the door opens.
	The remaining Twilio account balance is less than $5.

Table UM2: Taskbar Indicators.

Notifications

Notifications displayed on the taskbar provide more detailed information about critical events or changes in the status of the safety kit. These notifications may include alerts for missing config files, destroyed naloxone, and disarmed sensors. Users should pay close attention to notifications and take appropriate action to ensure the safety kit functions properly. Notifications may also provide instructions on resolving the issue or seeking further assistance.

Notification	Meaning
Config File Missing	The file “safety_kit.conf” does not exist.
Wrong Passcode	The provided passcode does not match the admin or paramedic's passcode.
911 Requested	The emergency call request has been created and placed in the request queue.
Naloxone Destroyed	The stored naloxone has been destroyed because of expiration or overheating.
Door Sensor OFF	The door sensor has been disarmed. The device will not place emergency phone calls until the door sensor has been armed again.
Close Door First	When trying to reset the system, the door is open.
SMS Requested	An SMS request has been created and placed in the request queue.
SMS Failed	The device cannot send the SMS using the Twilio service.
SMS Delivered	The device successfully sends the SMS using the Twilio service.
Call Requested	A non-emergency call has been created and placed in the request queue.

Call Failed	The device cannot send the call using the Twilio service.
Call Delivered	The device successfully sends the call using the Twilio service.
System Reset	The system has been reset to the default state.
Alarm Generated	An alarm mp3 file called “alarm.mp3” has been generated in the./res directory. If the device cannot place emergency calls, it will be played during an emergency.
Settings Saved	The settings have been saved to “safety_kit.conf”.

Table UM3: Taskbar Notifications.

Home Screen

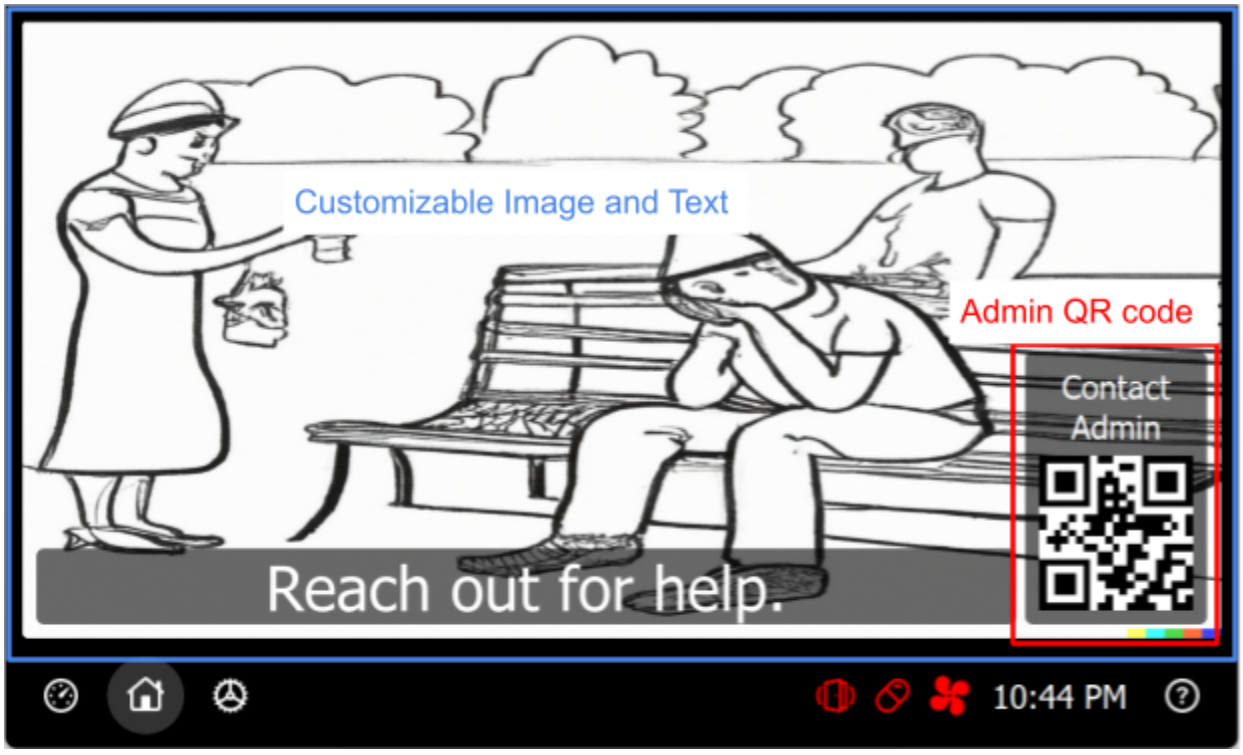

Figure UM2: Home Screen.

Home Screen Image and Text

The home screen of the Naloxone safety kit displays images and text that aim to increase public awareness about the overdose epidemic. This screen serves as an educational tool to raise awareness about the prevalence of drug overdoses and how to respond to an overdose emergency. The admin can customize the pictures and text displayed on the home screen before deployment to meet specific demands and maximize public awareness. This allows the admin to tailor the content to the target audience and location of the safety kit.

QR Code for Admin Contact

The admin's phone number is displayed in a QR code on the home screen in addition to the educational content. Anyone with a smartphone can scan this code to contact the administrator to report any problems or learn more about the Naloxone safety kit. The public can quickly and conveniently contact the administrator through this feature to report whether the safety kit works appropriately.

Dashboard Screen

The device's current state is shown on the Dashboard page. The user can view the current state of various device components on this page and take any necessary action. The Dashboard page helps the user monitor the device and take the necessary steps to ensure its proper operation by providing a comprehensive overview of its status. The dashboard screen will transit to the home screen after 1 minute automatically.

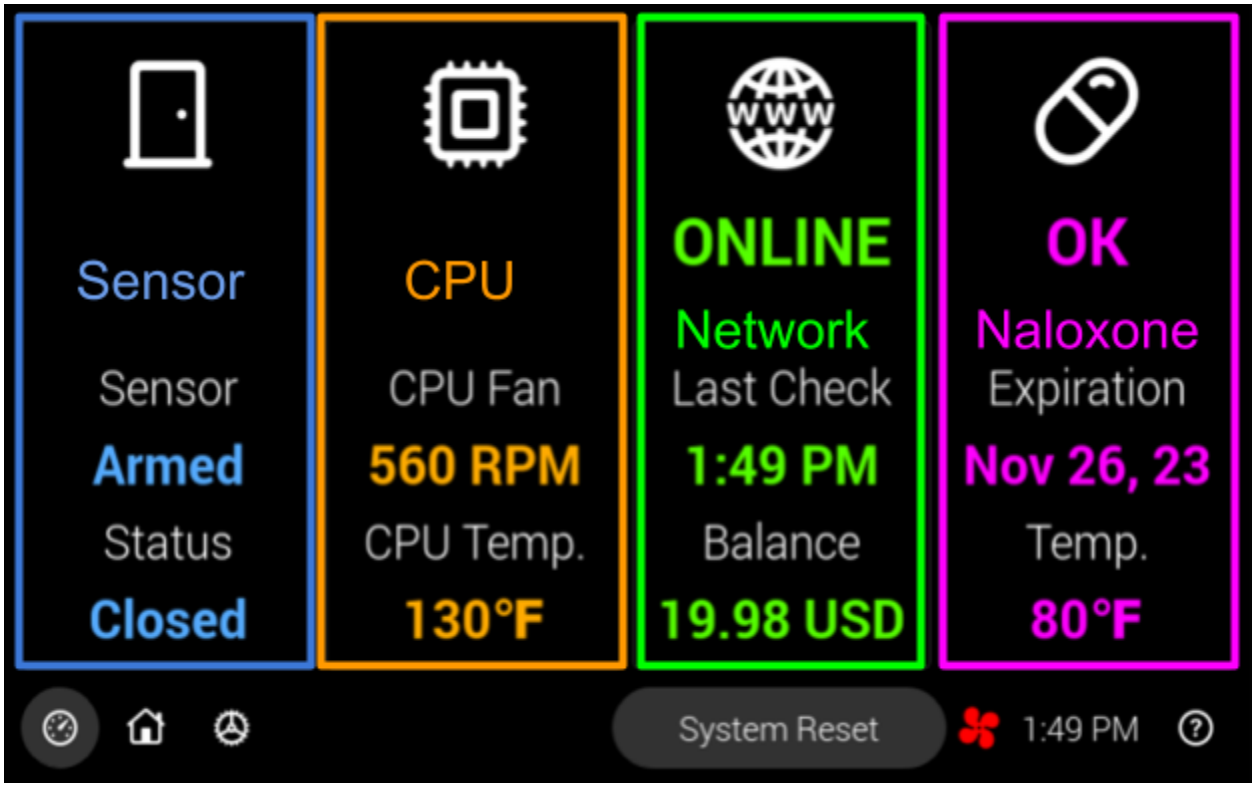

Figure UM3: Dashboard Screen.

Sensor Column

The Sensor column displays the door's status, including whether it is closed and whether the door sensor is disabled. This column provides information on the door sensor of the device.

CPU Column

The CPU column displays the CPU's current temperature and the CPU fan's speed. Monitoring the CPU temperature can help identify potential issues with the device’s hardware, and the CPU fan speed indicates the device’s workload.

Network Column

The Network column shows whether the device is connected to the Twilio service and the remaining Twilio account balance. This information helps the user ensure that the device functions correctly and that there is sufficient balance to cover communication costs.

Naloxone Column

The Naloxone column shows the expiration date of the naloxone and whether the naloxone has been destroyed due to overheating or expiration. The status of the naloxone can help the user ensure that the device has a reliable naloxone supply.

Settings Screen

The Settings allows the user to set different device parts. Security, Naloxone, Twilio, Emergency Call, Alarm, Power Management, and Admin are among the tabs on the settings page. Users can lock or unlock settings through the Security tab. Users can set the naloxone's expiration date and maximum temperature in the Naloxone tab. Users can modify the information associated with their Twilio account through the Twilio tab. Users can set the phone number called in an emergency using the Emergency Call tab. The Alarm tab permits users to create MP3 documents to alarm nearby people and change the volume. The Power Management tab lets users decide whether to enable active cooling and adjust screen brightness. The Admin tab helps users to test their settings utilizing calls or SMS messages. The software can be tailored to meet the user's specific requirements by allowing users to modify these settings.

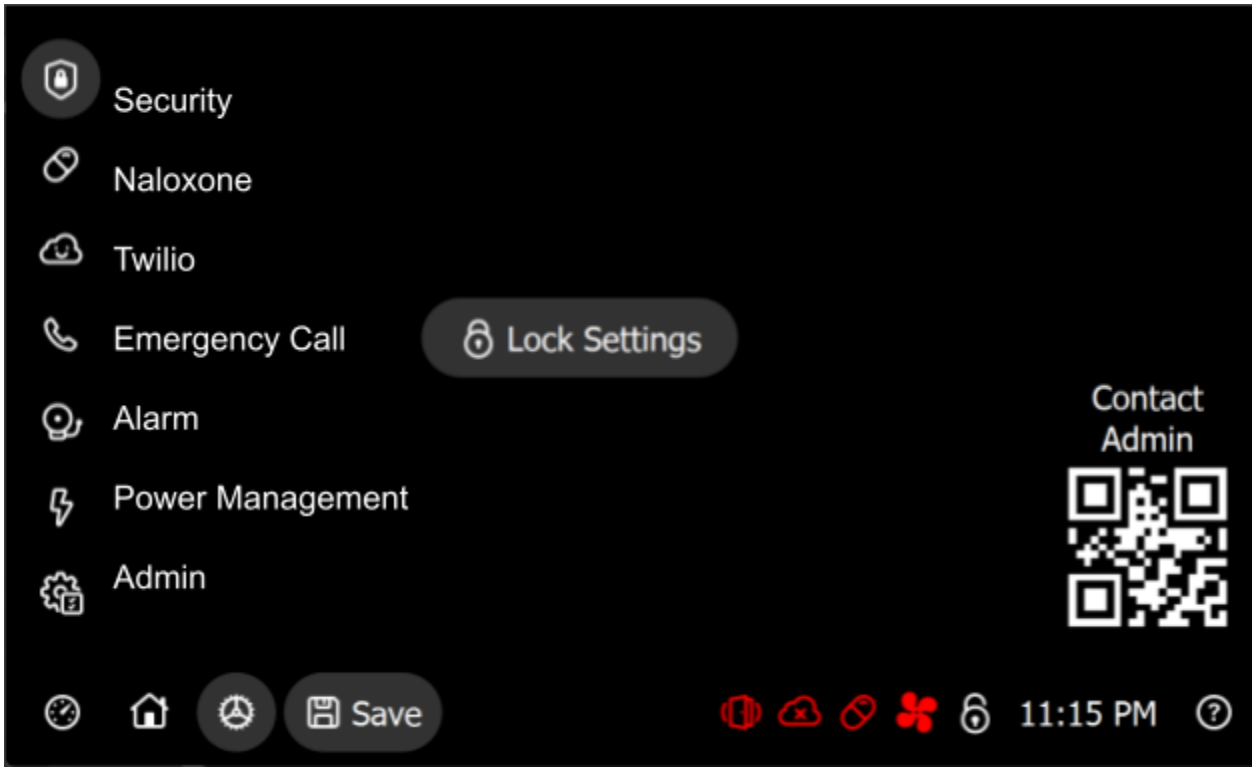

Figure UM4: Settings Tabs.

Door Open Screen

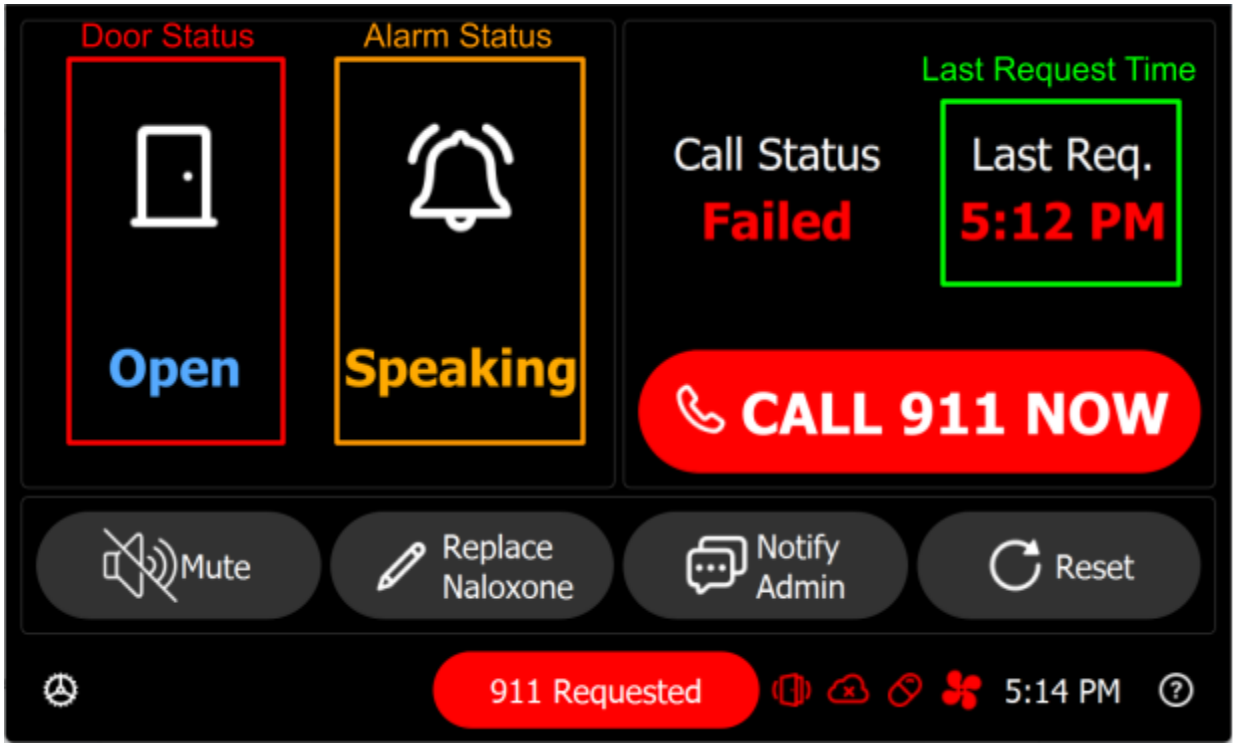

Figure UM5: Door Open Page UI.

Pushbutton Function Summary

Note that not all pushbuttons will be shown on the door open screen at the same time. The UI will adjust itself so that the user can get access to the most needed function in a simplified manner.

Pushbutton	Function
Stop	Stop the countdown timer so that the emergency phone call will not be automatically called.
CALL 911 NOW	Call the emergency number immediately. Pressing this button will automatically set the countdown timer to timeout.
Mute	Stop the alarm sound. You cannot enable the alarm again unless the next emergency call request failed.
Replace Naloxone	The settings page will appear so the user can enter new information about the replaced naloxone.

Notify Admin	The admin will receive an SMS saying that paramedics have arrived.
Reset	Reset the device to the default state. Always close the door before pressing this button.

Table UM4: Door Open Screen Pushbuttons.

On Door Open

The door open screen will be shown when the door is opened, and the door sensor is enabled. It will first show a countdown timer. Within the countdown period, the user can stop the countdown to disable emergency phone calls by pressing the “Stop.” They can also shut down the door to go back to the previous screen when opening it accidentally.

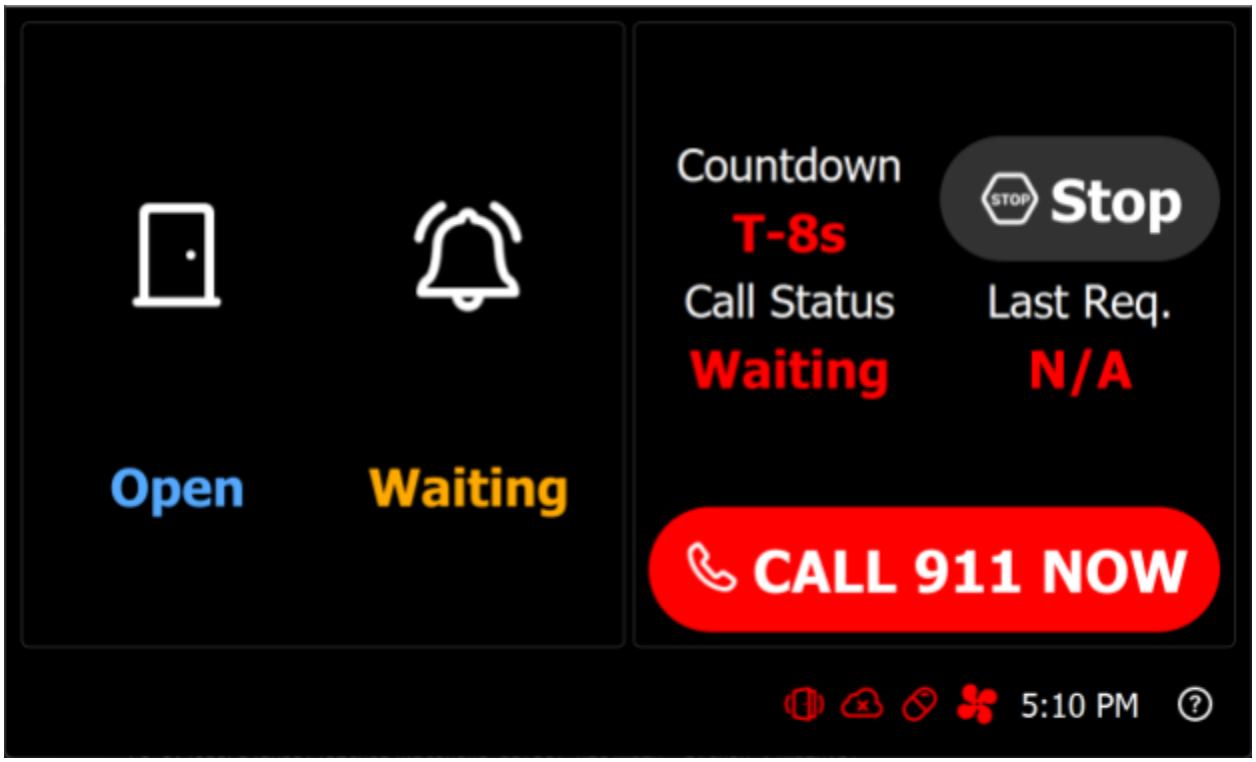

Figure UM6: Door Open Page with Countdown Timer.

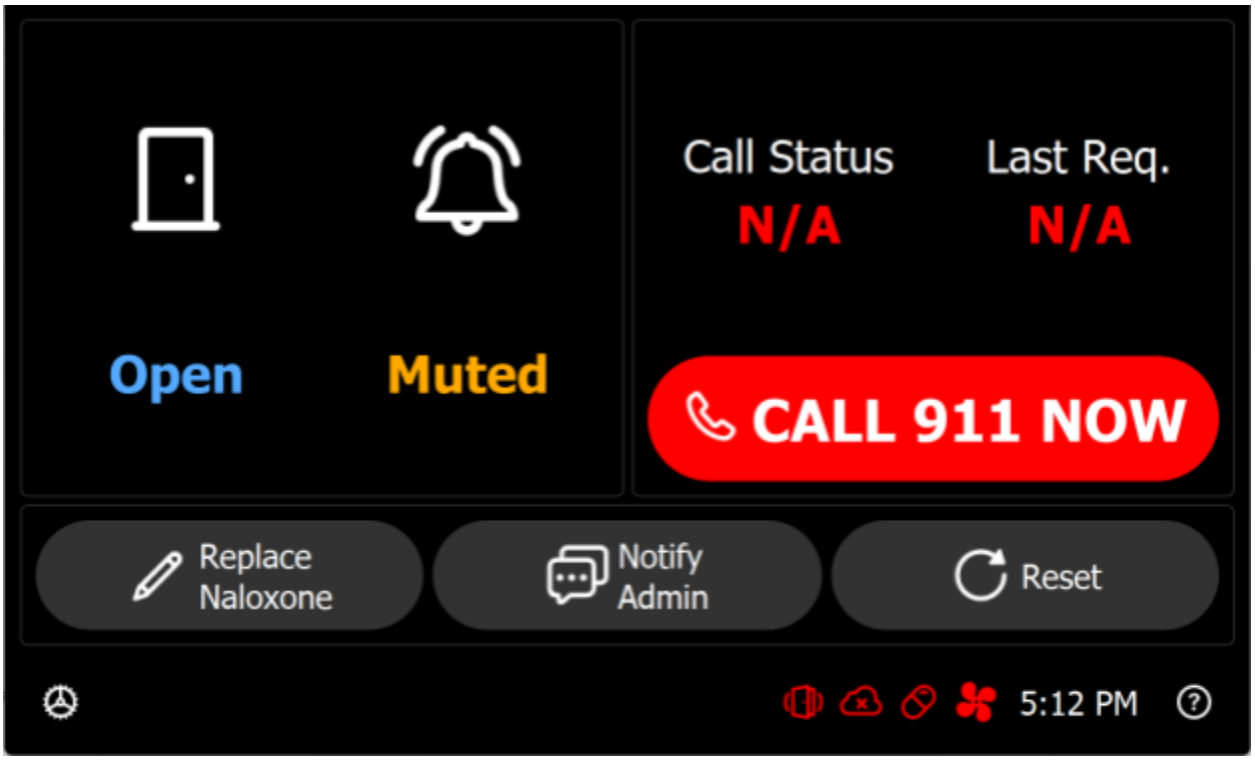

Figure UM7: Door Open Page after Pressing Stop.

After pressing the “Stop” button, the user must manually reset the device. They need to ensure that the door is closed before resetting. Otherwise, the device will not be reset and a warning will be shown in the taskbar. They can also observe the text under the door icon to check whether the door is closed or not.

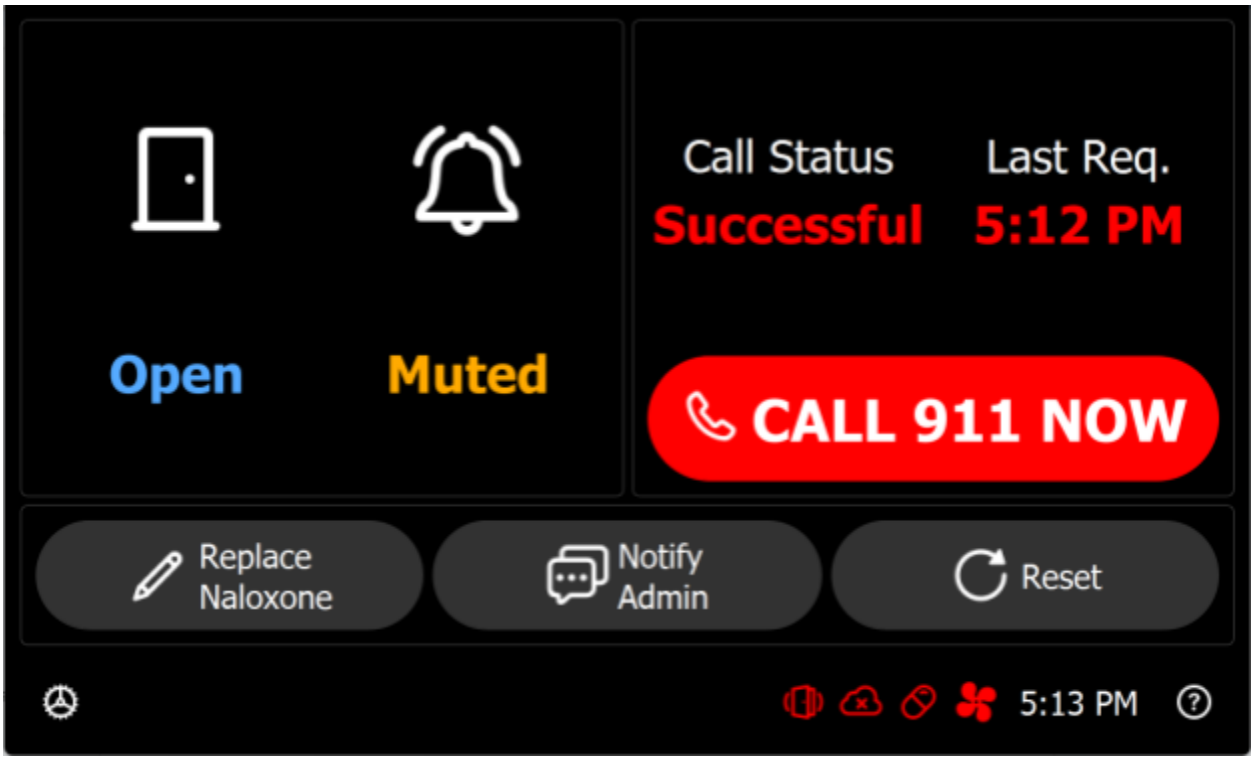

Figure UM8: Door Open Page after Successful Emergency Call.

If there is indeed an emergency, the user can press the “CALL 911 NOW” to call the emergency immediately or wait for 10 seconds before the device calls 911 by itself. If the 911 is placed successfully, the alarm will not be played by the device to minimize disruption.

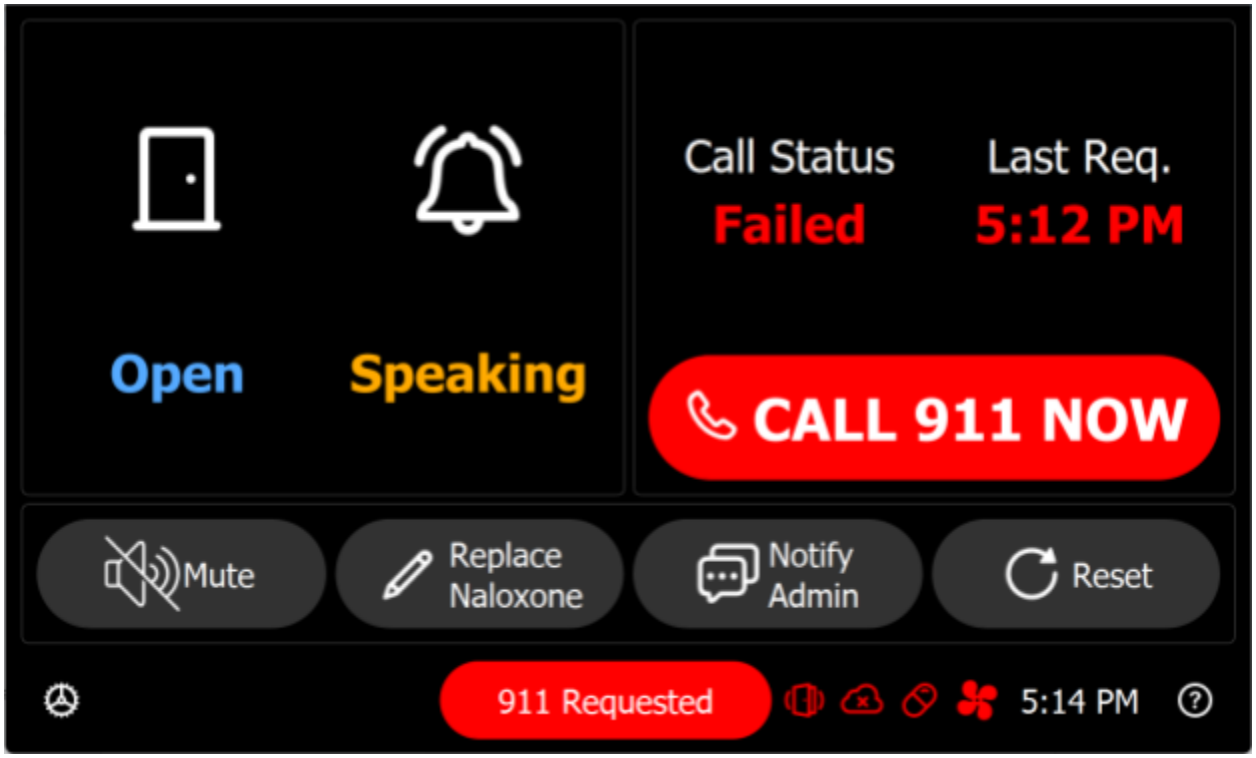

Figure UM9: Door Open Page After Unsuccessful Emergency Call.

If the emergency call cannot be placed successfully, the device will play the alarm file generated when setting up. The user can press “Mute” to stop the alarm if they want to.

Passcode Screen

Both the administrator and the building manager can enter their passcodes on the passcode screen to access the various settings that are available to them. Paramedics can use the "Notify Admin" button to send an SMS to the admin so that the admin can come and replace the Naloxone on their own if the admin has restricted access to only the admin. Paramedics can enter their phone number into the designated field to retrieve the Naloxone passcode on their phone if the administrator grants them access. Using the "Forgot Passcode" function, the administrator can also get their passcode by sending an SMS to their registered phone number. This screen with a passcode helps ensure that only authorized individuals have access to the sensitive safety kit settings.

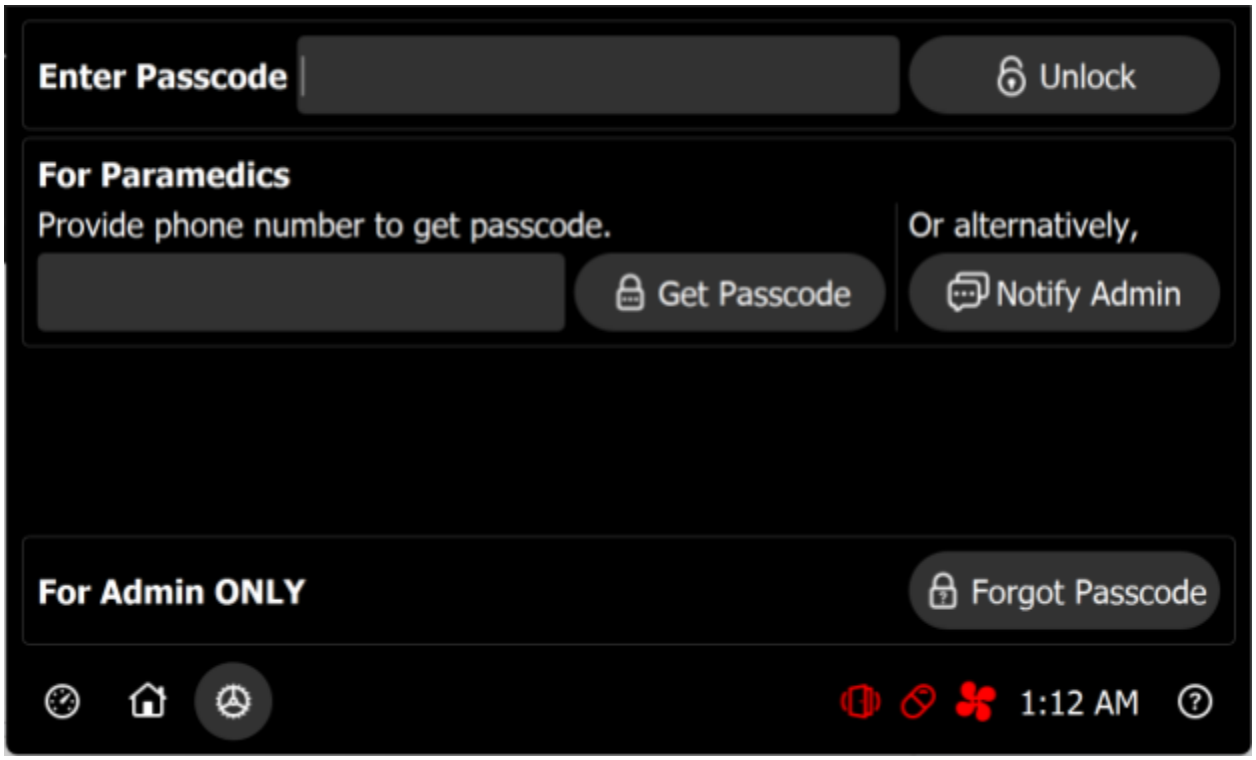

Figure UM10: Passcode Screen.

Settings

This section includes detailed information about all available settings to customize the user experience.

Security Settings

The Security page of the Naloxone safety kit allows you to secure the device's settings by locking or unlocking them. By default, all settings are locked, and you need to enter an admin passcode to unlock them.

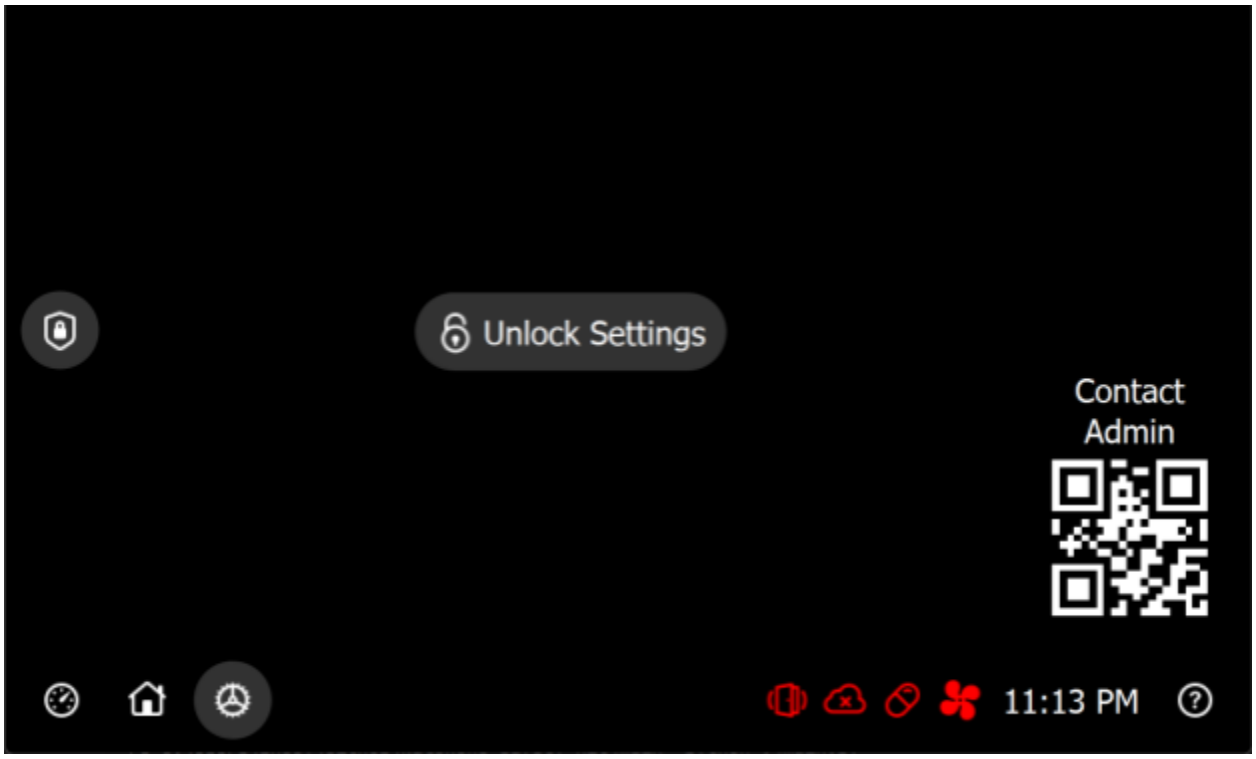

Figure UM11: Security Settings.

Unlock Settings

To unlock the settings, navigate to the Security page and press the "Unlock Settings" button. You will be taken to the Enter Passcode page, where you can enter the admin passcode. If no passcode is set, pressing the "Unlock Settings" button will automatically unlock all settings. Once the settings are unlocked, a white unlock icon will appear in the bottom right corner of the screen.

Lock Settings

To lock the settings on your device, navigate to the Security page and press the "Lock Settings" button. If a passcode is set, you will need to enter it again to access the settings. Alternatively, the settings will be automatically locked when you leave the settings pages. By locking the

settings, you can prevent unauthorized access to your device and ensure the safety of the Naloxone safety kit.

Contact Admin QR Code

Similar to the QR code on the home screen, the user can get the admin phone number by scanning the QR code.

Naloxone Settings

The Naloxone page on the Naloxone safety kit allows you to update information about the Naloxone nasal spray stored in the device. By using the controls on this page, you can set a new maximum temperature for storing the Naloxone and update the expiration date of the Naloxone using the calendar widget.

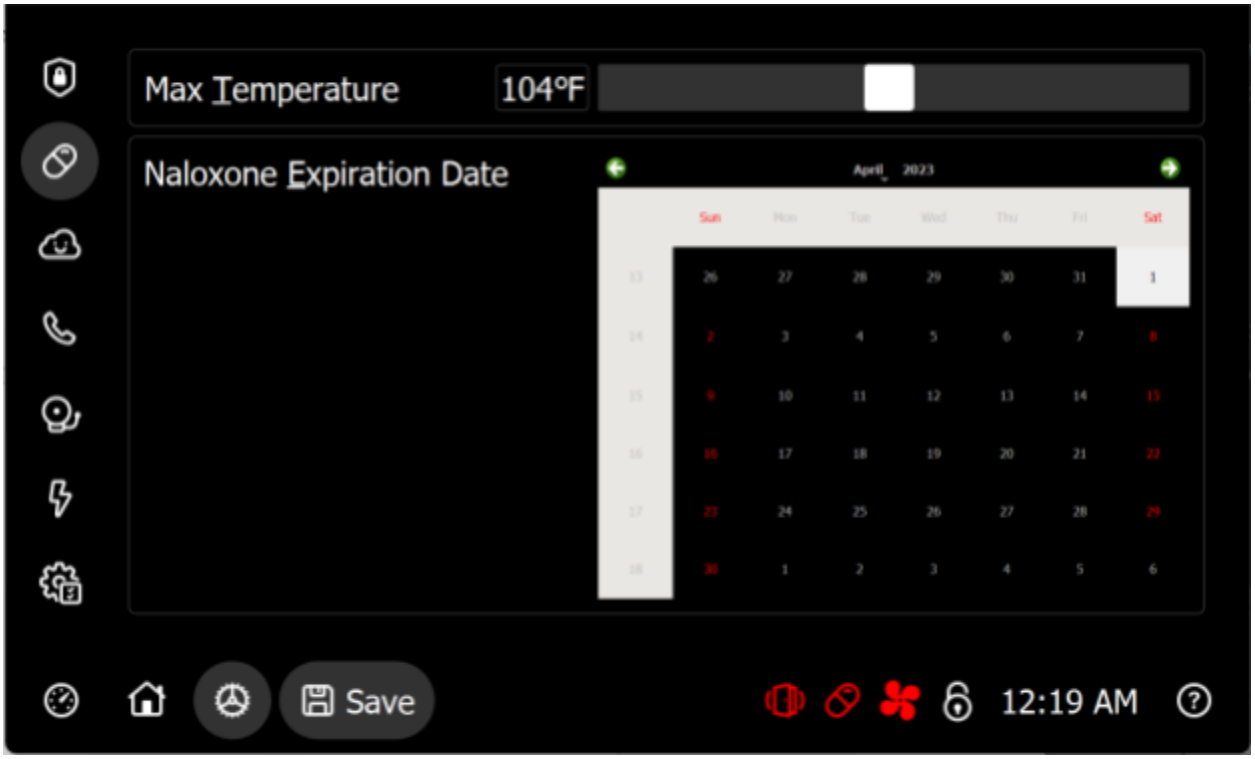

Figure UM12: Naloxone Settings.

Max Temperature

The Naloxone nasal spray should be stored at a maximum temperature of 104 degrees Fahrenheit to ensure its effectiveness. You can adjust this temperature by using the slider control on the Naloxone page. The current temperature setting will be displayed on the adjacent label. If the maximum temperature is reached, the system will display a red pill icon in the taskbar as well as a persistent notification. To ensure the proper functioning of the Naloxone safety kit, it is recommended to enable SMS reporting so that the admin can be notified in the event of overheating.

Naloxone Expiration Date

The Naloxone page also allows you to view and update the expiration date of the Naloxone. Once the Naloxone has expired, the system will display a red pill icon in the bottom right corner of the screen. It is recommended to enable SMS reporting so that the admin can be notified in the event of an expired Naloxone.

Twilio Settings

The Twilio page allows you to configure all the settings related to the Twilio service. Twilio is a cloud communications platform that enables you to send and receive text messages and phone calls. To use the Twilio service with your Naloxone safety kit, you will need to enter your Twilio virtual phone number, Twilio Account SID, and Auth Token. These settings can be found in the Twilio online console.

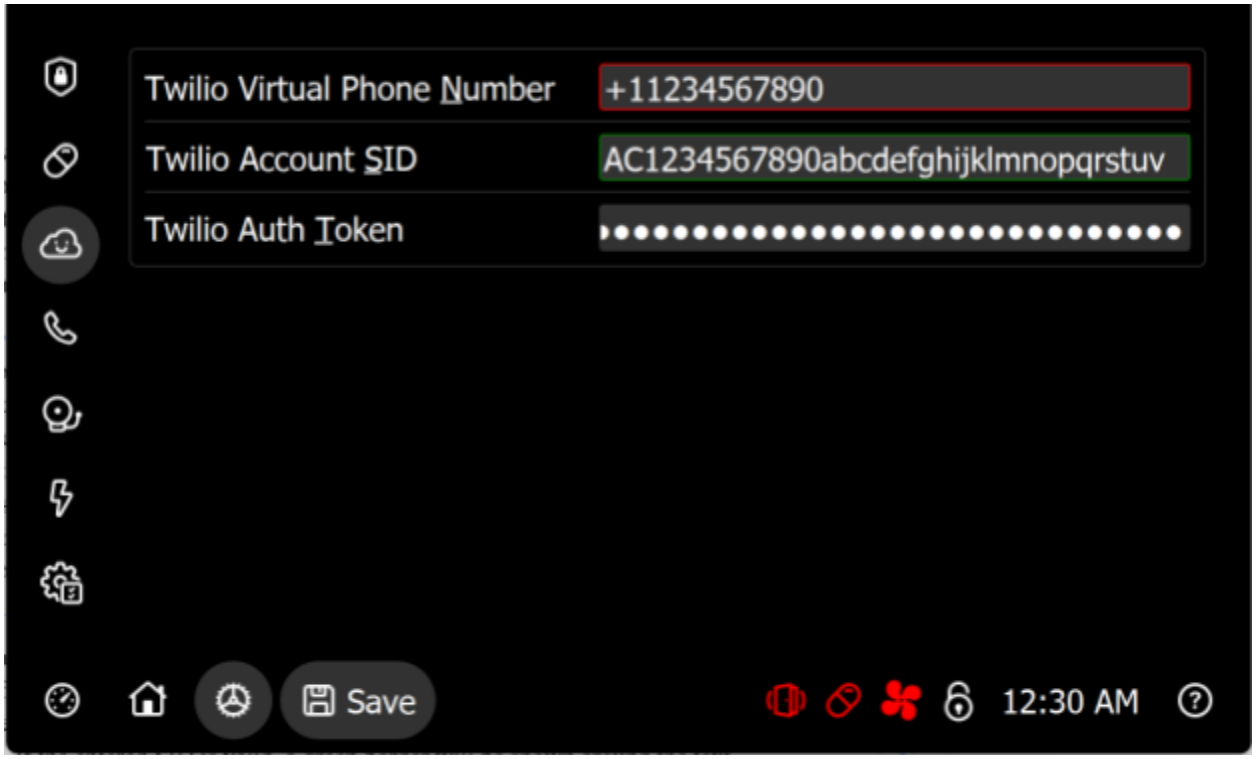

Figure UM13: Twilio Settings.

Twilio Virtual Phone Number

This is the leased phone number from Twilio that will be used to make and receive phone calls and text messages. You can enter the virtual phone number in the text box provided. Please make sure to include the area code (+1 for the U.S.) when entering the virtual phone number. The system will automatically check the validity of the phone number. A green border will be shown around the text box if the entered phone number is valid. If it is invalid, the frame will be red. Please note that the system does not prevent invalid phone numbers from being saved, but this will cause phone call requests to fail.

Twilio Account SID

The Account SID is a 34-character string identifier for your Twilio account, and it always starts with “AC”. You can find your Twilio Account SID in the Twilio online console. Make sure to include the “AC” when entering the SID. A validator will run in the background to check the

entered Account SID. If the entered SID is valid, a green border will be shown around the text box. If it is invalid, the border will be red. The system does not prevent an invalid Twilio account SID from being used.

Twilio Auth Token

The Auth Token is a security token that acts as a password for your Twilio account when sending requests. You can find your Twilio Auth Token in the Twilio online console. You will need to enter this token to access your account using Twilio's APIs. Please make sure to enter the token exactly as it appears in the Twilio online console.

Emergency Call Settings

The Emergency Call page allows you to set the emergency call destination, address, and message that will be used when making an emergency call.

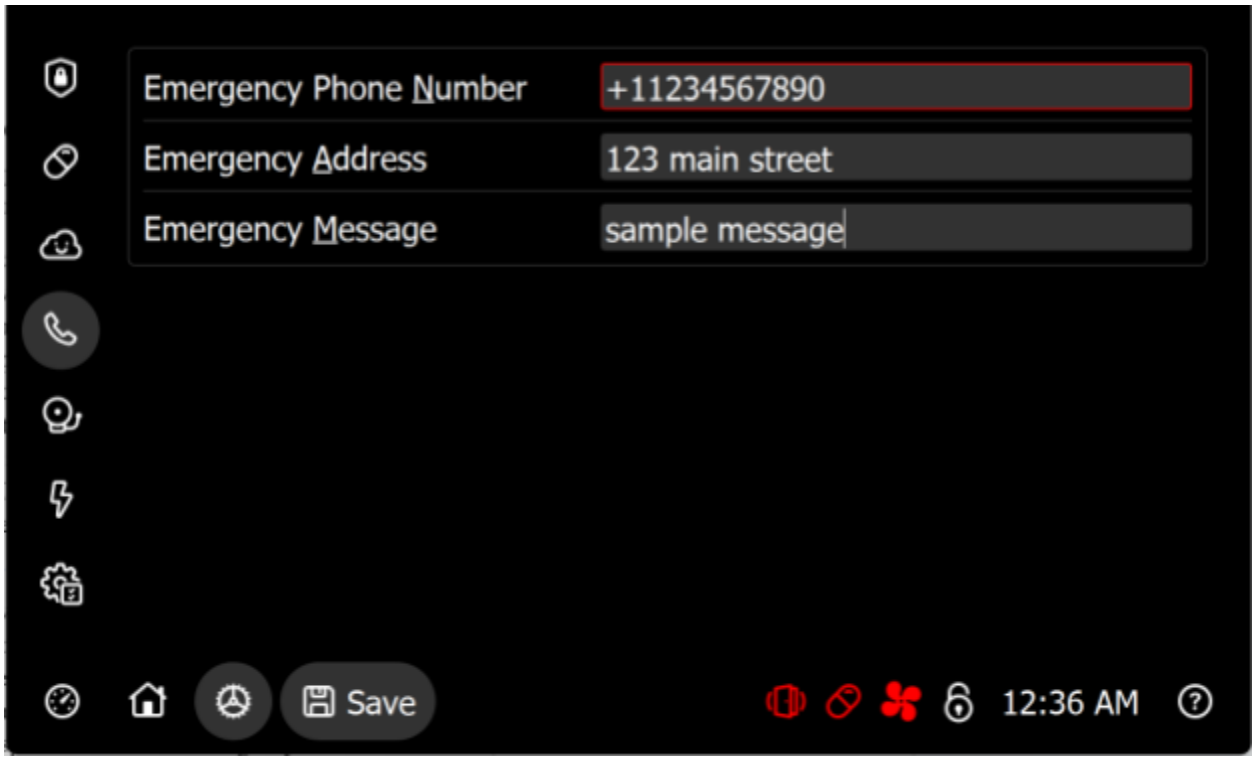

Figure UM14: Emergency Call Settings.

Emergency Phone Number

This setting allows you to set the destination for the emergency call. In most cases, it should be set to 911 to call emergency medical services in the event of an overdose incident. You can also set it to another phone number, such as the phone number of a local hospital or public safety department. It is important to make sure to include the area code when entering the phone number if you are not using 911 as the destination. The system will automatically check the validity of the phone number. A green border will be shown around the text box if the entered phone number is valid. If it is invalid, the border will be red. Please note that the system does not prevent invalid phone numbers from being saved, but this will cause phone call requests to fail.

Emergency Address

This setting allows you to set the installation address of the Naloxone safety kit. Make sure to be specific about the address, including the room and floor number and street name. **It is important to note that after entering the emergency address, you must also update it in your Twilio account. Failure to do so will result in high costs on your bill and may delay the response**

time of the emergency service. For details, visit [Emergency Calling for Programmable Voice | Twilio](#).

Emergency Message

This setting allows you to set a specific message that you may want paramedics to be aware of immediately, such as the route to the address. This information will be sent to the emergency service when making phone calls. Keeping this message brief and relevant to the emergency situation is essential.

Alarm Settings

The Alarm page is where you can set the alarm settings that will be used when it is impossible to make emergency phone calls.

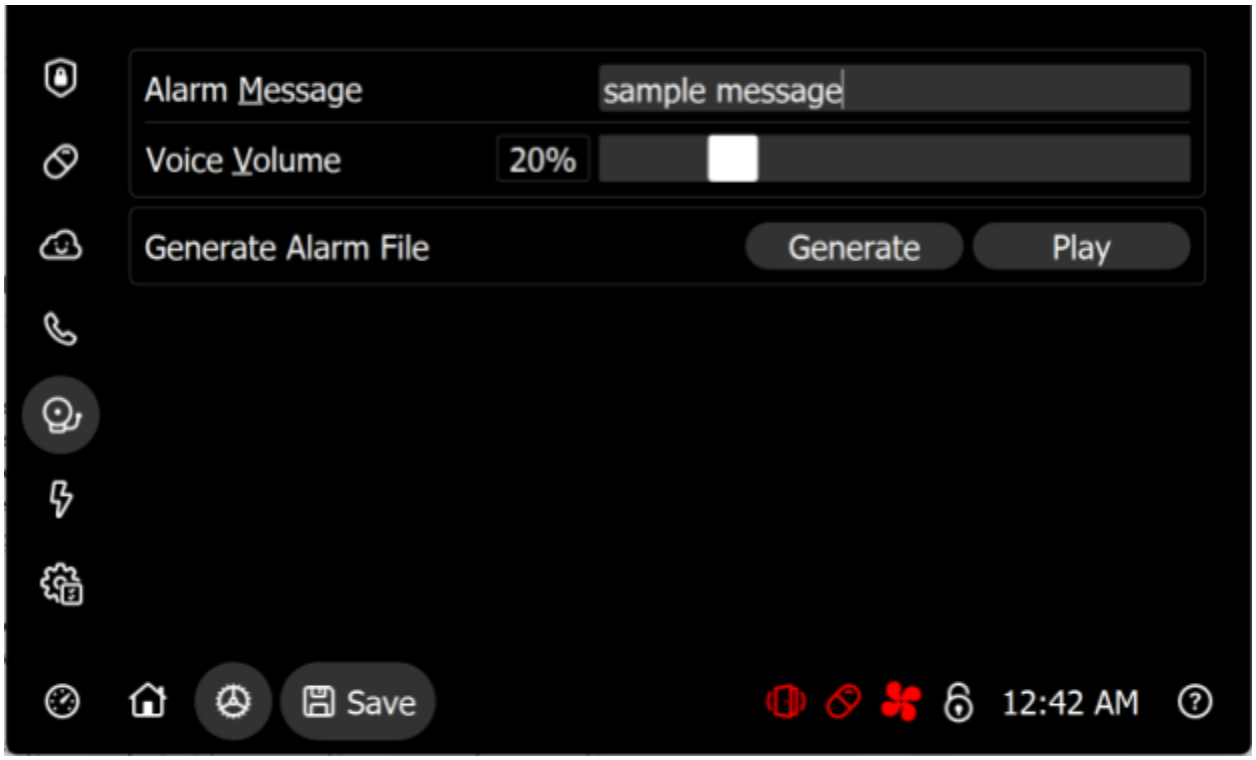

Figure UM15: Alarm Settings.

Alarm Message

The message will be spoken loudly by the system in the event of an emergency. It is important to use meaningful words in the message, such as “someone has overdosed,” so that people have a better understanding of the incident. The message will be passed to the Google text-to-speech engine, which will generate an mp3 file. This file will be played as the alarm message. It is recommended to keep the message clear and easy to understand.

Voice Volume

You can adjust the volume of the alarm using the slider. The best volume setting should make the alarm message clear without being too loud or too quiet. The volume of the USB speaker cannot be adjusted using the slider.

Generating File

Before the alarm message can be played, you need to generate the mp3 file using the Generate button. This function requires an active Internet connection. Once you have generated the file, it will be stored on the device for future use.

Testing

You can test the current alarm settings using the Play button. It is recommended to start with a lower volume setting before increasing the volume. Remember to save the settings after testing, as changes will not be automatically saved.

Power Management Settings

The Power page provides control over the power consumption of the device and cooling options.

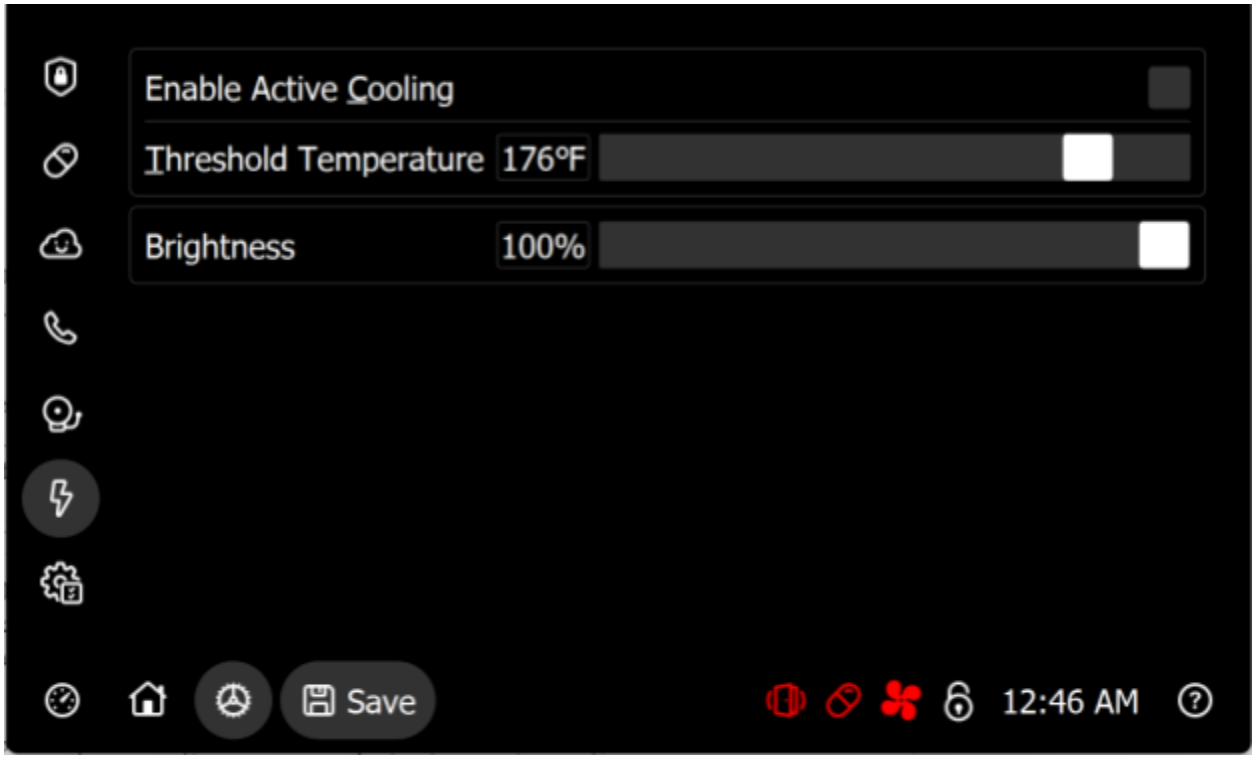

Figure UM16: Power Management Settings.

Enable Active Cooling

This option allows you to enable or disable the active cooling feature of the device. Active cooling involves using a fan to remove hot air from the device quickly. When this option is enabled, the cooling fan will turn on as needed. If it is disabled, the fan will always be off.

Threshold Temperature

This is the minimum temperature at which the cooling fan will turn on. By default, the threshold temperature is set to 176 degrees Fahrenheit. This value is recommended to minimize noise while still ensuring adequate cooling. A linear relationship between temperature and fan speed will be used to regulate the fan.

Brightness

This option allows you to adjust the brightness of the display. By default, the brightness is set to the system default level. You can use the slider to increase or decrease the brightness to your preference. You need to use the “Save” button to make the change persistent.

Admin Settings

The Admin page is designed for advanced users who need to control the behavior of the device and access advanced settings. These settings should be adjusted only by the admin after reading the user manual thoroughly.

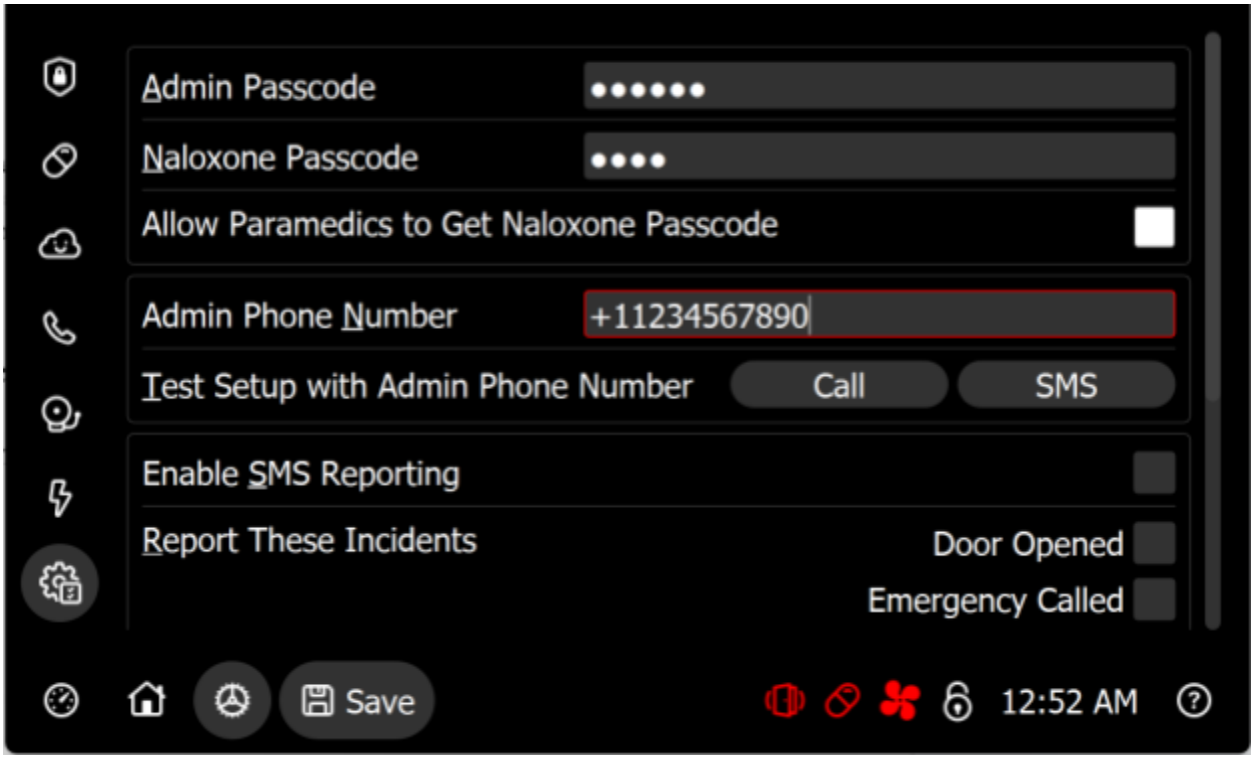

Figure UM17: Admin Settings 1.

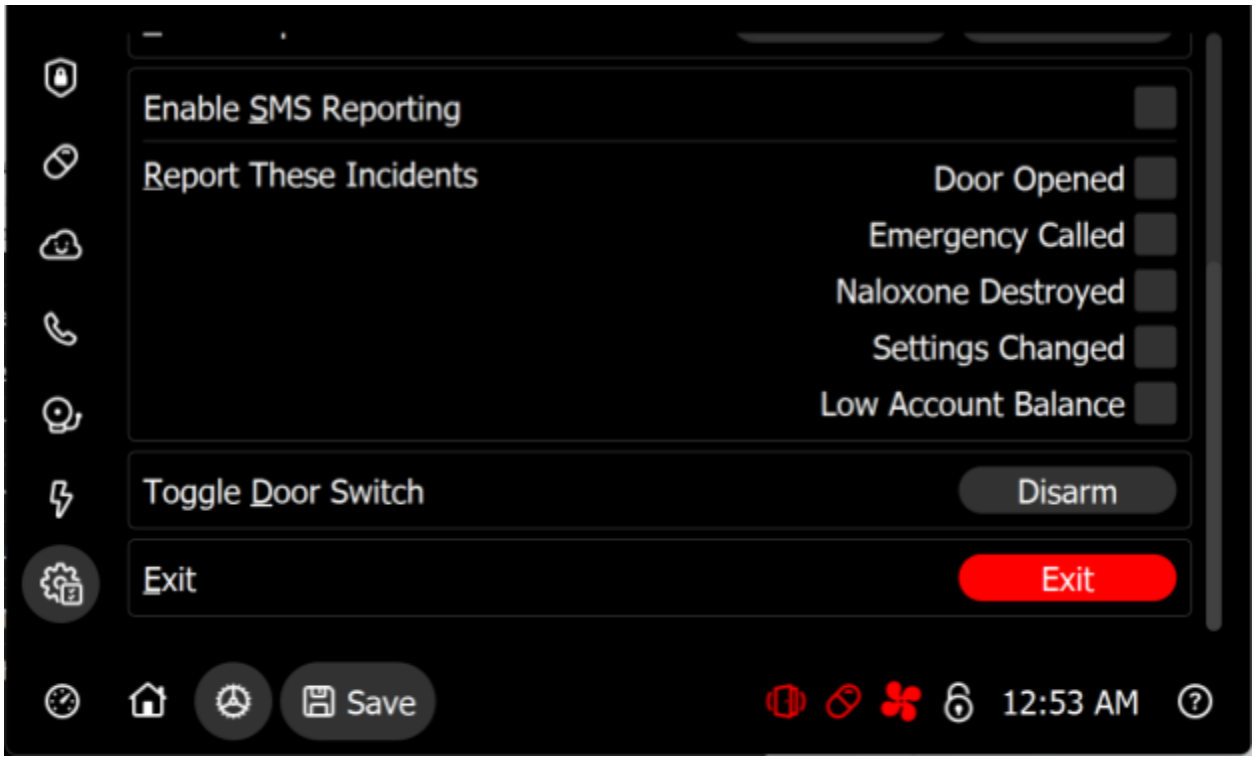

Figure UM18: Admin Settings 2.

Admin Passcode

This is the passcode used to unlock all setting sections on the device. If you leave this field empty, then anyone will be able to access the settings, which may be convenient in certain situations. However, if you want to ensure that only authorized personnel can make changes to the device, then you should create a strong admin passcode.

To create an admin passcode, simply enter your desired passcode into the “Admin Passcode” field and then click the “Save” button.

Naloxone Passcode

The Naloxone passcode is a separate passcode that is used to unlock the Naloxone settings of the device. This can be useful if you want to restrict access to all settings but still allow authorized personnel to make changes to the Naloxone settings. To set a Naloxone passcode, enter your desired passcode into the "Naloxone Passcode" field and then click the "Save" button.

Allow Paramedics to Get Naloxone Passcode

When this option is enabled, paramedics who arrive on the scene of an overdose can retrieve the Naloxone passcode by entering their phone number into the passcode page.

The phone numbers of the paramedics will be sent to the admin for additional support. If this option is disabled, paramedics can only notify the admin via SMS. Please note that the passcode retrieval feature requires a balance on your Twilio account.

Admin Phone Number

This is the phone number of the admin, which can be used to receive SMS updates on the status of the device. To set the admin phone number, simply enter the phone number into the “Admin Phone Number” field and then click the “Save” button. The system will automatically check the validity of the phone number. A green border will be shown around the text box if the entered phone number is valid. If it is invalid, the border will be red. Please note that the system does not prevent invalid phone numbers from being saved, but this will cause phone calls or SMS requests to fail.

Test Setup with Admin Phone Number

You can test the emergency call during the setup process by pressing the “Call” or “SMS” button. This will send a phone call or an SMS to the admin with the message that the emergency service will hear. This can be a valuable way to ensure that the emergency calling feature is working correctly before deploying the device.

Enable SMS Reporting

By enabling this option, the admin can receive status updates on the device via SMS. This can be useful for keeping track of the device’s status when you are not physically present. Some SMS notifications, such as the passcode retrieval request by paramedics, cannot be disabled. You can find the details in Management.

Door Opened
By enabling this option, the device will automatically send an SMS to the admin phone number when the door is opened and the door sensor is armed. The SMS will be sent as soon as the door has been opened.

Emergency Called
By enabling this option, the device will automatically send an SMS to the admin phone number when the emergency phone number has been called. The result of the calling will be sent to the admin as well so that the admin can come to help if the emergency call fails.

Naloxone Destroyed
By enabling this option, the device will send a daily SMS to the admin phone number if the Naloxone has been destroyed so that the admin can come to replace the destroyed Naloxone.

Settings Changed
By enabling this option, the device will send an SMS to the admin phone number if the settings are changed.

Low Account Balance
By enabling this option, the device will send a daily SMS to the admin phone number if the Twilio account balance is less than $5. The admin should refill the account balance as soon as possible since the daily SMS will be sent daily if the account balance is lower than $5.

Toggle Door Switch

This feature allows the admin to disable the door sensor when modifying the device. After pressing the “Disarm” button, the device will ignore signals sent by the door sensor. A red door sensor icon will be displayed in the taskbar. The device can also be reset with the door sensor disabled on the door open page. Entering the settings page on the initial setup will automatically disarm the door switch. Remember to turn on the door sensor after completing any modifications by pressing the “Arm” button.

Exit

By pressing this button, the program will exit to the desktop. Any unsaved changes will be lost when you leave the program. If you have made any changes you want to keep, click the "Save" button before leaving.

Core Functionalities

The Naloxone safety kit provides several core functionalities to help prevent and respond to overdose incidents. The most important functionality is the emergency call feature, which allows the automatic calling of emergency medical services in the event of an overdose. The kit also allows users to store Naloxone nasal spray at an appropriate temperature, with the option to set a maximum temperature and expiration date.

The kit also provides a secure passcode system to restrict access to specific settings and features, allowing for greater control and safety. Other core functionalities include a customizable alarm message and volume, power management options such as active cooling and brightness control, and the ability to generate and play audio files during network outages. By providing these core functionalities, the Naloxone safety kit offers a comprehensive solution for preventing and responding to overdose incidents.

Emergency Call

Emergency calling is a critical function of the Naloxone safety kit, designed to quickly alert emergency services in the event of an overdose incident. When the door is opened, and the door sensor is armed, the door open page will show automatically and wait for 10 seconds before calling the emergency number. This is to give the user enough time to stop the countdown in case of a false alarm or accidental door opening.

Suppose the emergency number cannot be reached due to poor network coverage or technical difficulties. In that case, an alarm will be played via the built-in speakers to attract the attention of nearby individuals. The volume of the alarm can be adjusted on the Alarm page of the settings. The alarm message can also be customized to provide more information about the emergency incident.

It is important to note that the emergency calling function requires a valid Twilio phone number to be set in the Emergency Call settings page. By default, the phone number is set to 911, but it can be changed to any other phone number for testing purposes. It is also essential to keep the device connected to a reliable network to ensure that the emergency call can be made promptly.

If the user accidentally triggers the emergency calling function, there is no way to cancel the phone call. The user should contact the admin immediately by pressing the “Notify Admin” pushbutton on the door open page.

Naloxone Monitoring

The Naloxone monitoring function of the safety kit is a critical feature that helps ensure the effectiveness of the naloxone nasal spray stored in the kit. The DHT22 temperature sensor

constantly monitors the temperature of Naloxone. The admin or building manager can set the expiration date on the Naloxone settings page. It is important to note that once the Naloxone has expired, it will no longer be effective in reversing an overdose.

If the Naloxone has been destroyed, a red persistent notification will be shown in the taskbar. Additionally, if the admin has opted to receive notifications, they will receive an SMS alert about the destroyed Naloxone. However, it is important to remember that there is no way to prevent people from using expired or damaged Naloxone. Therefore, it is crucial for the admin to regularly monitor the Naloxone safety kit and ensure that it is stored properly to maximize its effectiveness.

In summary, the Naloxone monitoring function of the safety kit is a crucial feature that helps ensure the effectiveness of the Naloxone nasal spray stored in the kit. By regularly monitoring the expiration date and temperature of Naloxone, users can help maximize its effectiveness in reversing an overdose. Additionally, by enabling SMS reporting and regularly checking the Naloxone safety kit, users can help ensure that any issues related to Naloxone are addressed promptly.

Security

The device's security is of utmost importance to ensure that only authorized individuals have access to the settings and functions of the safety kit. The Security page allows you to lock or unlock the settings on your device, providing an extra layer of protection against unauthorized access. Press the “Unlock Settings” button to unlock the settings and enter the admin passcode on the “Enter Passcode” page.

It is recommended to set a secure admin passcode to prevent unauthorized access to the safety kit. This passcode can be used to unlock all setting sections, and if left empty, the settings will be accessible to everyone. You can also set a different Naloxone passcode for the Naloxone settings to provide quick access to paramedics when replacing the Naloxone while keeping most settings secret.

Additionally, the safety kit comes equipped with a door sensor that will automatically trigger an emergency call in the event that the door is opened. When the door is opened, and the door sensor is armed, the Door Open page will show automatically and wait for 10 seconds before calling the emergency number. If it is impossible to call the emergency number, an alarm will be played via the built-in speakers. This feature ensures that any attempt to tamper with the safety kit will trigger an immediate response.

Finally, it is worth noting that enabling the SMS Report feature on the Admin page will allow the admin to receive status updates of the device via SMS. This feature can be useful in detecting

any attempts to tamper with the safety kit and notifying the admin if the Naloxone is destroyed or expired. The safety kit is designed with security in mind, ensuring that only authorized individuals have access to the settings and functions of the device.

Alarm

The administrator can set an alarm message that will be played through the built-in speakers with the Naloxone Safety Kit when the Twilio service is unavailable. This feature can be used to notify people in the area about the emergency. The administrator has the ability to modify the alarm message to include any necessary information. The feature converts text to speech using Google Text-to-Speech (TTS), which is compatible with numerous languages. The volume slider can be used to adjust the alarm message's volume so that it can be heard even in noisy environments.

Management

The Naloxone Safety Kit provides a variety of reporting options to make device management simpler for administrators. The daily report of the destroyed naloxone is one of these features. When naloxone is destroyed either by overheating or expiration, the device will send a daily SMS to the admin directly so that the admin can come to replace the destroyed naloxone. Moreover, the admin can choose to get a notification about the low account balance so that they can refill the Twilio account balance to ensure that the emergency call can be made successfully. Both of these events will only send one SMS per day to minimize the communication cost.

Some critical events, such as door open events and emergency calling events, will be reported immediately if the admin chooses to enable the SMS reporting feature. The admin will get a SMS in the event of an emergency so that the admin can come and help resetting the Naloxone safety kit. When a tampering incident occurs, such as when someone enters the wrong passcode, the administrator can also receive SMS notifications.

Please keep in mind that the SMS notifications feature is dependent on the Twilio service. To avoid service interruptions, the administrator ought to make certain that their Twilio account has sufficient funds. Also note that the SMS reporting feature can be disabled completely in the admin settings if the admin decides to do so.

Appendix A: Troubleshooting

This appendix shows some troubleshooting steps to some common software problems. Also, you should follow these steps as well to solve general problems:

- Try to restart the device to see whether the problem disappears,
- Check the connection between sensors and Raspberry Pi to rule out hardware connection issues,
- Check the Twilio account balance. The SMS and calling will not work with insufficient account balance.

System Menu Bar Shows in GUI

You can re-hide the menu bar by pressing anywhere in the GUI. If you want to prevent the menu bar from showing in GUI, you can right click on the menu bar and select Panel Settings. Then, go to the Advanced tab and tick Automatic Hiding and set size when minimized to 0 pixels. However, you will need to use the mouse to interact with the menu bar afterwards.

Persistent Notification “Naloxone Destroyed” Shows in Taskbar

You need to replace the naloxone immediately. This notification means that the naloxone is ineffective to reverse the overdose because it is either overheated or expired. After replacing the naloxone, you need to enter the new information of the naloxone in the settings.

CPU Fan Not Working

First, you need to check whether the active cooling is enabled in the settings. You also need to ensure that the CPU temperature is higher than the threshold temperature in the settings. To avoid conflict PWM signal, you need to disable the Fan in the Raspberry Pi Configuration.

GUI Does Not Respond Immediately to Switch Changes

This is normal. The IO thread in the software is designed to be run every second to avoid lock up while maintaining a high sample rate. The GUI should respond in less than 1 second after the switch state changes.

Cannot connect to the Internet

Ensure that the Wifi is connected correctly. The Raspberry Pi is unable to connect to the eduroam network by default, you need to use the wpa_supplicant following the manual from your organization.

Emergency Call Not Working

First, do not try to test the device with the emergency phone number set to 911. You need to first change the emergency phone number to your personal phone number. You do not need to add area code before 911 when setting the emergency phone number. Also, make sure that the provided phone number is correct and start with the area code. Also ensure that your Twilio account has sufficient balance.

Naloxone Temperature Not Displaying

First check the wiring to the temperature sensor. If the wiring is correct, wait for around 15 seconds since the software will try to read from the temperature sensor every 10 seconds. A new temperature data should be available when the software reads from the temperature sensor again.

Alarm Not Working

First, you need to press “Generate” after entering the alarm message in the settings so that gTTS will be used to generate the alarm file needed. Also, when pressing “Generate”, ensure that the network is working properly. You can also observe the taskbar since a notification will be displayed when the file is generated. You can also press the “Play” button in the settings to play the file once. The alarm will only be played when the Twilio emergency phone call failed.

Appendix B: References and Resources

- Software Source Code: zyl120/Naloxone_Safety_Kit: Repository for Meng ECE design project (github.com)
- PyQt5 - Show virtual keyboard: https://stackoverflow.com/questions/62473386/pyqt5-show-virtual-keyboard
- Install QtVirtualkeyboard in raspberry-pi?: https://stackoverflow.com/questions/63719347/install-qtvirtualkeyboard-in-raspberry-pi
- The Twilio Python Helper Library: https://www.twilio.com/docs/libraries/python
- Connect Raspberry Pi to eduroam: https://inrg.soe.ucsc.edu/howto-connect-raspberry-to-eduroam/
- Resources for icons: https://icons8.com

Appendix C: License Information

Software License

GNU Lesser General Public License
Version 3, 29 June 2007

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions
As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions
If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files
The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the object code with a copy of the GNU GPL and this license document.
4. Combined Works
You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries
You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
6. Revised Versions of the GNU Lesser General Public License
The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you

may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Icon Sources

Icons on the GUI come from https://icons8.com.

Appendix D: Naloxone Replacement Manual For Paramedics

Print this manual and stick it on the door of the naloxone safety kit for reference.

Naloxone Replacement Procedure

1. Enter your phone number in the box provided, including your area code. If you have a US phone number, please format it as "+1XXXXXXXXXX".
2. Press the "Get Passcode" button. You will receive the passcode on your phone via text message.
3. Enter the passcode in the top box on the screen and press the "Unlock" button to gain access to the Naloxone expiration date settings.
4. Update the expiration date for Naloxone by using the calendar widget or entering the date manually.
5. Also change the maximum storing temperature if needed. You should be able to find the maximum storing temperature at the box of the naloxone.
6. After entering the new expiration date, press the "Save" button to save the changes.
7. Once you have saved the changes, press the "Home" button to exit the settings page.

Notifying Admin Procedure

Press the "Notify Admin" button so that the admin can know that an overdose incident has occurred.

What information will be collected?

Your phone number will be sent to the admin so that they can help you replace naloxone and reset the device afterwards. Other personal information will not be collected.

Appendix E: safety_kit.conf Template

You can choose to edit the configuration file *safety_kit.conf* on a personal computer and then copy the file into the directory containing *main.py*. Below is a template for the configuration file. Remember to replace the fields with the correct information.

```
[twilio]
twilio_sid = AC123456
twilio_token = abcdefg
twilio_phone_number = +11234567890

[emergency_info]
emergency_phone_number = +11234567890
emergency_address = 123 Main Street
emergency_message = Sample Emergency Message

[naloxone_info]
naloxone_expiration_date = Sat Apr 1 2023
absolute_maximum_temperature = 104

[admin]
passcode = 123456
naloxone_passcode = 0000
admin_phone_number = +11234567890
enable_sms = False
report_door_opened = False
report_emergency_called = False
report_naloxone_destroyed = False
report_settings_changed = False
report_low_account_balance = False
allow_paramedics = True

[power_management]
enable_active_cooling = False
threshold_temperature = 176
brightness = 100

[alarm]
alarm_message = Sample Alarm Message
voice_volume = 20
```

Appendix F: Systemd Unit File

You can use the following file to start the software automatically at boot. Follow the instructions in the section Autostart.

```
[Unit]
Description=Safety Kit
After=multi-user.target

[Service]
User=username
Group=usergroup
Environment=DISPLAY=:0
Environment=XAUTHORITY=/home/username/.Xauthority
Environment=XDG_RUNTIME_DIR=/run/user/1000
Environment=QT_QPA_PLATFORMTHEME=qt5ct
WorkingDirectory=/home/username/Documents/Naloxone_Safety_Kit/main
Restart=on-failure
RestartSec=5s
ExecStart=/usr/bin/python3 main.py

[Install]
WantedBy=graphical.target
```