Security Sensory Assistive Monitor (S.A.M.)

By: Martin Alava, William Cole, Nathaniel Furhang
Department of Engineering and Environmental Science
Mentor: (Jessica) Xin Jiang, Ph.D

ABSTRACT

Security S.A.M. is a security system built to provide a modular smart home experience. The user will be notified of the activation of entry sensors through changes in the Center Console lighting display patterns and the activation of the Haptic Notification Module. This system integrates cybersecurity and wireless communication standards while also offering user configurability in their own security system.

INTRODUCTION

Security Sensory Assistive Monitor (S.A.M.) is a senior engineering design project. It is designed to reduce the cost of smart home security and offer more customizable. Through our research, many smart security products offered by standard vendors have little customizable options and often require monthly subscriptions. Our project offers consumers an effective way to create their own security system without a monthly fee. This project provides the user with the inherent sensitivity of Smart Home solutions and programs was mitigated by allowing for automatic updates on HA to patch security flaws monthly. Another means to limit security breaches was to only allow the Center Console to connect to the user's private network.

WORKING PRINCIPLES

Security S.A.M. uses modular sensors to monitor the opening and closing of doors and other entrances. The block diagram in Figure 1 shows how the Sensors interact with the Haptic Notification Module.

Two types of sensors are used in the system. Specific Sensors are placed at important entrances, such as the front door, and are integrated with a camera, with high priority when notifying the Center Console. This sensor also takes and emails the user a photo of the entrance when it is opened. General Sensors do not include a camera and are placed at windows and other low priority locations. These Reed Switch Sensors act as publishers and send messages to the Center Console through Message Queuing Telemetry Transport (MQTT).

While on standby, the Center Console outputs a desired lighting pattern on the LED chain shining on a piece of frosted acrylic, creating the visual LED Panel shown in Figure 4. This is done through a Python script and the GPIO, with the brightness controlled by a light sensor also connected to the GPIO. When the Center Console receives an alert message from a Reed Switch Sensor, Node-RED has the Center Console start a Python Script saying which sensor was activated and for an alert lighting pattern, such as flashing red lights, to activate for a set amount of time. An alert message is also sent to the Haptic Notification Module, seen in Figure 5, to activate the connected haptic driver and motor for a set amount of time. This allows for one room to have the Center Console and another room to have the Haptic Notification Module, covering more space for the user.

PROJECT DETAILS AND METHODS

Security S.A.M. follows a process to deliver the user the necessary notifications when a sensor is activated. Figure 2 shows a Specific Sensor, consisting of the Raspberry Pi, camera, and reed switch. A Reed Switch Sensor activates when a reed switch pair connected to the door/window and the surrounding frame are disconnected. The reed switch is powered by its connection to the Raspberry Pi's General Purpose Input Output (GPIO) pins. A magnetic field occurs between the two sides of the switch, separating them stops the flow of electricity and activates a Python script. This script sends an alert message through MQTT to the subscribing Center Console that the entrance was either opened or closed. If the Reed Switch Sensor was a Specific Sensor, the Python script has the sensor take a photo of the opened entrance and send it to the user as an email.

Several programs and add-ons were used to control the Center Console's activities. The Home Assistant Supervised method installed the program through command line instead of as an operating system to maintain administrative control over the Raspberry Pi, which was necessary to control the connected LED Panel. The programming language “YAML Ain’t Markup Language” (YAML) was used to configure an HA add-on, Node-RED. This is a programming tool used to graphically tie together JavaScript functions’ inputs and outputs to include other subsystems necessary for accessing the sensors and outputs of Security S.A.M. The open-source program Mosquitto MQTT was installed and configured in the same way, allowing the Center Console to subscribe to the Reed Switch Sensors and publish to the Haptic Notification Module. Figure 3 shows Lovelace, a dashboard manager for HA, which allows for an end-user customizable Graphical User Interface (GUI) that can manipulate the functions available to them, such as disconnecting a sensor or stopping an alarm. Google Cast Text to Speech (TTS) allows the Center Console to verbally communicate the received alerts to the end-user through Node-RED and MQTT.

While on standby, the Center Console displays a lighting pattern on an LED chain shining on a piece of frosted acrylic, creating the visual LED Panel shown in Figure 4. This is done through a Python script and the GPIO, with the brightness controlled by a light sensor also connected to the GPIO. When the Center Console receives an alert message from a Reed Switch Sensor, Node-RED has the Center Console start a Python Script saying which sensor was activated and for an alert lighting pattern, such as flashing red lights, to activate for a set amount of time. An alert message is also sent to the Haptic Notification Module, seen in Figure 5, to activate the connected haptic driver and motor for a set amount of time. This allows for one room to have the Center Console and another room to have the Haptic Notification Module, covering more space for the user.

LIMITS, RISKS, AND MITIGATIONS

There were some cyber security concerns when designing this project. When trying to implement the camera into the Specific Sensor, an issue that occurred when trying to stream and take a photo of the camera feed, due to the limitations of the serial interface of the Raspberry Pi. The project moved forward with the Specific Sensor taking a photo of the entrance only when it was opened.

When designing the TTS feature of the Center Console, a WS2801 LED chain was specifically used so the GPIO could accommodate an connected speaker. Using Google Cast speakers also reduced which GPIO pins would have to be used.

CONCLUSION

Security S.A.M. was designed to offer a more aesthetically pleasing security system without a monthly fee. This project provides the user with a quick notification as to the opening of their door and/or other entry ways, with providing a picture and email for the main entrance. This project offers consumers an effective way to create their own security system, suited to their needs.

REFERENCES

Please scan to access reference list

Figure 1: Block Diagram of Security S.A.M.
Figure 2: Raspberry Pi Zero Equipped with Reed Switch and Camera
Figure 3: Lovelace Dashboard GUI for end user
Figure 4: LED Panel with Center Console
Figure 5: Haptic Feedback Motor