Smart Door Security System

Adam Stefanik, Moisess Rodriguez,
Reham Hammad, Zachary Jackovich

University of Central Florida
Department of Electrical and Computer Engineering
Orlando, Florida 32826, USA

***Abstract* – Our project's objective is to design and build an innovative smart door security system that combines easy accessibility with highly advanced and secure features. The user will be able to approach and unlock their home's door hands-free, while still maintaining the highest level of security available today. The system leverages true Two-Factor Authentication, including the use of high-level facial recognition software to determine the identity of who is at the door, and high-level speech-to-text software to confirm the pin number only known by the homeowner. The group selected this project because we believed that we could improve the smart door features that already exist on the market today, as well as challenge ourselves by designing, building and testing a project that put our Electrical and Computer Engineering skills to the test, all of which will better prepare us for our future careers in the field.**

I. Introduction

As people continue to incorporate advanced technology in their homes, smart locks were one of the technologies that people desired to have in their homes. For the past decade, this field has been noticeably improving by combining different technologies into one bigger design. Our Smart Door Security System uses some of these technologies such as facial recognition and speech-to-text software. Both facial recognition and speech-to-text have a designated purpose, which combined allow our system to be extremely secure. Facial recognition verifies who is at the door, and speech-to-text allows the user a hands-free way to enter a pin number, of which that only the homeowner should know. When combined, the facial recognition and pin number entry create a true Two-Factor Authentication system that is extremely secure and very difficult to crack. Without the knowledge of the pin number and having the correct facial identity, the door will not open. And with the option to either enter a pin number on a physical keypad, or verbally into a microphone, leveraging advanced speech-to-text software, the homeowner has the option to unlock their home's door completely hands-free, which is useful in numerous situations such as having a both hands full of groceries. Overall, the system is designed to be very secure but still extremely accessible, if not more accessible than your standard deadbolt and key.

II. System Concept

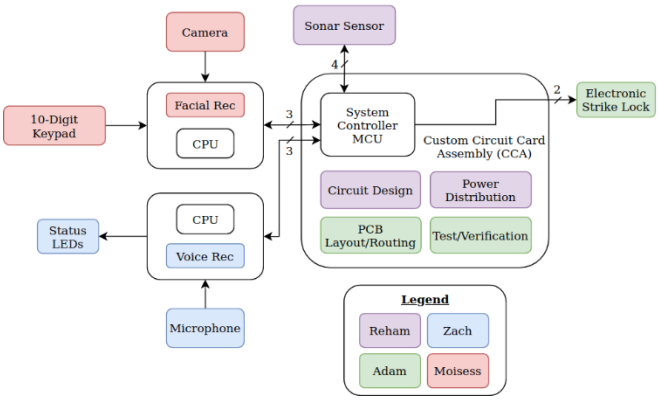

Figure 1. Smart Door Security System Block Diagram

The block diagram in Figure 1 describes the high-level architecture of our Smart Door Security System and assigns the lead team member responsible for each part of the system. Although we had a lead team member

assigned to each part of the system, we all contributed and supported each other in many ways in order to be successful and create a reliable system. In short, the system will be comprised of a custom circuit card assembly which will interface the 2 high-level computers to the sonar sensor and electric strike lock. The custom circuit card assembly will also be designed to distribute the power rails needed throughout the rest of the system. With each high-level computer, we have attached the peripherals that are required to meet the needs of the system. For the computer running facial recognition, we have the camera connected through USB, as well as the 10-Digit Keypad for physical pin number entry connected with GPIOs. For the computer running the speech-to-text software, we have a microphone connected through the USB port as well as LED's that will glow different colors based on the status of the system's authentication.

The project specifications, objectives, and electrical standards were used to derive our system hardware concept. Having a list of requirements to achieve provided us with a clear goal to accomplish. Our project's main objectives were for the system to be reliable, affordable, and easy to use. After that, marketing requirements were specified and based on this information, engineering design requirements were outlined, all of this shown in the House of Quality below in Table 1.

Column #		1	2	3	4	5	6
Direction of Improvement	Polarity	▲	▲	▲	▲	▲	▲
Customer Requirements (Explicit and Implicit) \ Engineering Requirements		Facial Recognition	Voice Recognition	Pin Entry	LED Status Lights	Notifications	Sonar
Cost	▼	●	●	▽	▽	●	▽
Reliability	▲	●	○	●	▽	●	○
Modularity	▲	▽	▽	○	○	○	●
Useability	▲	●	●	●	●	●	▽
Accuracy	▲	●	○	○	▽	▽	○
Security	▲	●	●	○	▽	●	▽
Target		95%	95%	95%	95%	95%	95%

Table 1. House of Quality

III. System Components

Our system integrates many components, which work together to provide the homeowner with a convenient yet secure way to access their home. Each of the system's components for the Smart Door Security System will be discussed in this following section.

A. Microcontroller

To select a microcontroller for this project, we searched for an affordable model that we had some experience with, provides a large number of I/O pins for flexibility and backup design options, and has plenty of online support. After careful consideration, our group selected the MSP430FR6989 by Texas Instruments. It's a very low power integrated circuit with a 16-bit RISC architecture and has up to a 16 MHz clock speed. The 100 pin package includes 2 channels for either I2C, SPI, UART and a plethora of other GPIO functions, and with a price tag of only $7 per unit, this microcontroller is an extremely affordable choice. Considering we also had extensive experience with it from our previous Embedded Systems course, this microcontroller was an easy choice.

B. Raspberry Pi 4

For the high-powered software that our system requires, we needed a high-powered computer in a small form factor, and the Raspberry Pi 4 is the perfect choice for this component. With 8GB of RAM, there is plenty of memory to run a high-powered and computation intensive software algorithm such as facial recognition or speech-to-text recognition. On top of that and in order to lighten the load even more from the Raspberry Pi, we decided to use 2 Raspberry Pi's, and let each high-level software program run on its own individual Raspberry Pi 4 computer. This would allow all resources to

be utilized at their full potential to ensure the system is fast and can recognize and authorize the homeowner quickly, aligning with the easy accessibility goal of our system. Finally, the Raspberry Pi 4 has 40 GPIO pins allowing almost unlimited accessibility with other peripherals that we include in our system, which are all detailed below.

C. Ultrasonic Sensor

The ultrasonic sensor was our go-to choice for initiating the start of our software. By leveraging the low power mode of the MSP430FR6989 microcontroller, we can run at a very low power, until someone is sensed approaching the door. At the point when someone is sensed approaching the door, the system's microcontroller will trigger the system to turn on and begin running the authentication of facial recognition and pin number entry. We chose to go with the HC-SR04 ultrasonic sensor since we had previous experience with this sensor, and we had this component already.

D. Camera

A standard USB webcam's camera is needed for our system to run advanced facial recognition software. This camera is connected to the Raspberry Pi 4 through USB which is easily detected as input automatically thanks to the Raspberry Pi 4 being such a high-level and high-powered computer.

E. Microphone

A standard USB microphone is needed for our system to allow verbal pin number entry, as a mode of easy accessibility. Like the webcam, the USB microphone is easily detected and selected as the audio input for the Raspberry Pi 4. This microphone is used as the input for the system's speech-to-text method of user authentication.

F. LED Lights

Basic LED strip lights that connect to the Raspberry Pi 4 via USB were used to indicate status of the system, throughout the authentication process. When someone is detected at the door via the sonar sensor, and the advanced recognition software begins running, the LED status lights will shine white. When the first of 2 authentication methods are verified, the LED status lights will shine yellow. Finally, when the homeowner is authenticated the LED status lights will shine green. If by chance someone besides an authorized homeowner tries to enter and is declined access, the LED status lights will shine red which indicate not authorized.

G. Electric Strike

An electric strike lock was utilized for the lock mechanism. This was decided for a few reasons, most importantly being we could implement this lock on the outer frame of the door, and easily connect it directly to our system on the outside door frame via power and ground wires. When we want to unlock the door, we pulse a GPIO from the MSP430 system controller on our circuit card assembly which temporarily cuts power to the strike lock, rendering the door "unlocked", until power is restored to the lock which automatically re-locks the door. This mechanism is easy and reliable and is perfect for our system. Our system requirement was that the lock will be as strong or stronger than a conventional deadbolt, chich has ~600lbs of holding force. We chose an electric strike lock with 2200lbs of holding force, providing a significant improvement over our design requirement.

H. 10-Digit Keypad

A 10-Digit Keypad is used for physical pin entry, should the homeowner not want to verbally enter their pin. This keypad is connected to the Raspberry Pi 4 using 8 GPIO pins, but because there are many GPIO pins available this is not an issue. The keypad was

easily incorporated as an input method nested inside the existing python program that we had running.

I. LCD Screen

Although not pictured on our system's block diagram, an LCD screen was implemented successfully. This began as a stretch goal that we would aim to implement if everything went smoothly, and fortunately we were able to successfully meet this stretch goal. The 16x2 LCD screen connects to the Raspberry Pi via I2C and provides fantastic visual prompts as the user is detected and goes through the authentication process. Finally, when the homeowner is detected, a nice "Welcome Home, *Homeowner's Name*" message is printed.

IV. Hardware Detail

The MSP430FR6989 on our PCB acts as a system controller for the Raspberry Pis. It controls an ultrasonic sensor, and when a user is within 1 meter of the door, it sends a signal to the Raspberry Pis to get them to start the high-power code for facial recognition and speech-to-text. When the Raspberry Pis each determine that a user has been authenticated, they inform the MSP430. Using two-factor authentication, the MSP430 makes the decision to apply power to the fail-secure lock to unlock it. Lock interface is done with a daughter board which contains a 3.3V relay and optocoupler. This allows the 3.3V MSP430 to control the 12V electric strike lock.

The hardware implementation of the Smart Door Security System involved a high degree of connectivity between the MSP430 system controller and sensors/Raspberry Pis. Rather than implement an existing protocol for communication between the Raspberry Pis and the MSP430, our group found discrete connections for each function to be better suited. A wire is dedicated between the MSP430 and Raspberry Pis to wake them up and start the facial recognition and speech to text. A wire is then dedicated to each authentication method: facial recognition, speech to text, and PIN entry. These methods each have a dedicated wire connecting them to the MSP430. When a GPIO pin on the Raspberry Pi is set high for an authentication mechanism, that indicates that a user has been authenticated. We found that this single-wire scheme better fit the needs of our project than existing schemes.

Due to an error in the initial design of the lock interface circuit, we procured a daughter board to interface between a MSP430 pin header and the electric strike lock. This daughter board contains an optocoupler and relay so that we can unlock the 12V lock from the 3.3V MSP430 output.

There were two other major PCB reworks, which will be detailed in this section. The first was straightforward: removing and replacing 10V rated capacitors for 12V rated capacitors due to a procurement issue with our board house. The next issue had to do with fine-tuning our power supplies after testing them in-circuit with a load. A resistor was added to reduce the voltage supplied by the 3.3V power supply, and a resistor was added in parallel with another in the 5V supply to increase the voltage to around 5.1V. This was necessary because the 5V power supply was decreasing voltage under load and under-powering the Raspberry Pis.

The hardware details not mentioned here are described in III. System Components and Figure 1. Smart Door Security System Block Diagram.

V. Power Detail

Through power calculations, we approximated that our system would require

7A at 12V. To allow for a safety factor, we set our input power requirement to be 12V, 10A. We elected to use 12V as our input despite most of the system being 5V, because our Electric Strike lock operates at 12 volts. To provide our design with the needed power we used a 12V 10A power supply plugged to a wall outlet as our input voltage source. After that, two buck converters were used to step down the voltage first to 5V and then to 3.3V to supply each circuit component with its appropriate voltage level.

Using such voltage converting circuits allowed us to satisfy the power specifications of our design components, while also meeting safety standards by handling an input voltage of 12 volts without causing electrical shock or fire hazards.

We tested this circuit by applying the input power from our regulated 12V, 10A power supply. We tested our circuit first with no load and next with a load. Every voltage level on the PCB was probed to ensure it was within the requirements of the Electric Strike Lock, Raspberry Pis, Ultrasonic Sensor, and MSP430FR6989. We made sure that there was not excessive ripple voltage and that our measured voltage at each level was close to as designed.

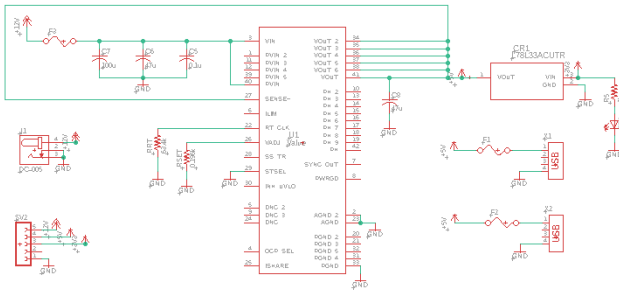

Figure 2. Power Circuit for Smart Door Security System

VI. Software Detail

As mentioned, to develop our system we incorporated two Raspberry pi's as well as a microcontroller. Each Raspberry pi, and the msp430 microcontroller handle specific tasks. For instance, "Raspberry pi number one" is responsible for running the facial recognition software. Raspberry pi number one will also control the LED lights. "Raspberry pi number two" is responsible for running the speech recognition software. In addition to this, Raspberry pi number two will also control a keypad where users will be able to enter in a four-digit pin, when prompted.

The main door control, or microcontroller will be the device which starts the entire system. It will be responsible for detecting users with the ultrasonic sensor connected to it, sending/receiving data to the pi's, and most importantly locking/unlocking the door. Each of these tasks will be explained in the following sections.

A. Facial Recognition

The facial recognition software will begin running on Raspberry pi number one as soon as a high voltage signal is received from the msp430 microcontroller. Once a user is sensed from the ultrasonic sensor at a certain distance, the microcontroller will send a high voltage signal to the pi.

More specifically, the code in the raspberry pi will have an infinite while loop which will be checking a specific GPIO pin. If the GPIO pin is set to high, then it will call the facial recognition function. The facial recognition application will be running in real time and detecting users through a web camera. This function will also control the LED status lights. Once the function is called, the lights will be initialized to white. If the pi recognizes a user's pin entry but not face, then the lights will turn yellow. If the pi recognizes a user's face, but not a pin, then it

will turn blue. If both pin and face are recognized the then lights will turn green.

Raspberry pi number one will receive a high voltage signal from Raspberry pi number two once an authorized pin is received. If a user's face and pin are both authorized, then a high voltage signal will be sent back to the microcontroller. The function will end, and the application will return to the infinite while loop, which will then continue to wait for a high voltage signal from the pi. If the facial recognition function is called but a user is not recognized within two minutes then the function will end, and the process will repeat. This will save power if the system was triggered by accident.

For example, if there was an instance where a person was delivering a package to the door. If this "Two-minute rule" was not there, then the function will keep running until a user is authorized. This will consume more power than the function not being called. Once the function is running, LED lights are on, and a video stream is being analyzed through the web camera from the facial recognition model.

The facial recognition model is able to determine authorized users because it was trained by analyzing multiple images of a specific person. If the model recognizes a user that it was not trained with, then that user is labeled as "unknown" and the door will not unlock.

The facial recognition function was written in python. Python allows us to import many helpful libraries. The library we used to implement facial recognition was OpenCv. We also used a library called imutils which allowed us to use a video stream from our webcam.

Within the application there are three main components which allow this facial recognition software to run. There is a python file which will train the model. This file will locate the desired user, by finding a folder that has the users name. So, in order to train multiple users, additional folders would have to be created with their names, and then a line of code would have to be change, to indicate the users name. Once the model is trained then an encoding.pickle file will be created. This file will have necessary data for recognizing faces. Finally, the detection method that the facial recognition software uses is Histogram of Oriented Gradients (HOG).

B. Speech-To-Text Software

The speech-to-text software was decided upon after countless hours of research and investigation. The initial design was going to use Voice Recognition or "Speaker Recognition" which identifies who is actually speaking, as opposed to speech-to-text, which just translates any audio input to written text. But after many weeks of trying to correctly implement voice recognition, we decided to switch our design to use speech-to-text, which would both be easier to implement and more accurate and secure in the end. This design change was perfect as we could now truly leverage Two-Factor Authentication, by verifying "something someone is" (facial recognition) and "something someone knows" (speech-to-text or physical pin entry).

The speech-to-text code is written based on a Python library called SpeechRecognition 3.8.1. The SpeechRecognition package leverages the *Recognizer* class which makes it easy to recognize speech from an audio input such as microphone. Luckily the SpeechRecognition library makes it easy to get up and running, including setting up to capture and retrieve the audio input from a microphone, which normally takes a lot of

work to build scripts to set up audio input from microphones.

The SpeechReocognition library is also very flexible and works with many existing API's. SpeechRecognition can act as a wrapper for several popular speech API's such as Google Web Speech API. Thankfully this allowed us to easily build and implement our speech-to-text python program, which prompts the user to say "Keypad or Speech" to select which method of pin number entry they would like to use when they approach the door. After the user chooses which method to enter their pin number, and then enters it either verbally with our implementation of speech-to-text or using the keypad, the program sets a GPIO to the MSP430 system controller "high" to signal that the homeowner's pin number was authenticated.

C. MSP430 System Controller

The MSP430 acts as the System Controller since the microcontroller is designed to be the interface between the high-level software programs and the lock and sonar sensor, hence the "brains" of the system. The raspberry pi's run the highly complex computations for facial and speech-to-text but without the System Controller to interface the sonar sensor and the lock to the pi's there would not be a complete system.

We designed main.c that runs on the MSP430 to leverage the low power mode and monitor the sonar sensor's distance measurement, where a measurement less than 1m will trigger an interrupt captured through Timer A Channel 0. By using the low power mode and the sonar sensor's echo pin to trigger an interrupt, our system will save on power when someone is not within sensing range of the door.

The rest of the logic for our system is handled in the Interrupt Service Routine as well, where we are monitoring the input GPIO pins from both raspberry pi's – which are set "high" whenever either of the high-level software programs have their authentication methods, facial recognition or pin number entry. When both the facial recognition and the pin number entry authentication methods are verified by the pi's and set their GPIO pins "high", this will satisfy the logic within the ISR which then calls a function called "unlock_door()" which just sets the output GPIO that is wired to the relay "high" for 2 minutes, allowing the homeowner to enter their home. After those 2 minutes are up, the door will auto-relock and the program will go back to low power mode, and resume progress until another echo pin signal triggers an interrupt.

VII. Board Design

Before starting the design process, we have first studied the PCB standards. Since our design is a dedicated electronic service it falls under the general electronic products, the used standard to design such product is IPS-2221. Autodesk Eagle was the software we choose to design our PCB schematic and layout. Using Eagle was convenient to us because it allows the user to create a circuit schematic using the exact circuit parts chosen by the designing engineer. Figure. Display an image of our fully assembled printed circuit board.

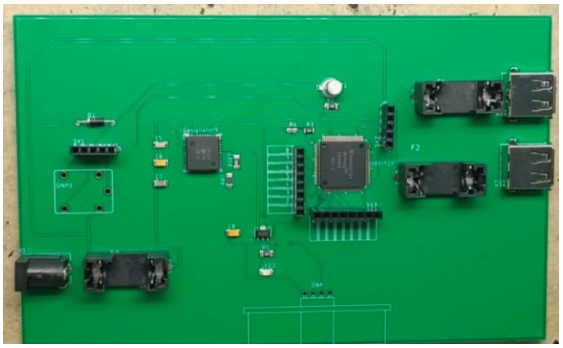

Figure 3. Smart Door Security System Circuit Card Assembly

IX. Conclusion

In summary, we have designed, built, and tested the Smart Door Security System which incorporates advanced facial recognition and speech-to-text software giving the system a user friendly, accessible, and extremely secure door lock system. With true Two-Factor Authentication, you must meet the biometric facial features of the homeowner as well as have knowledge of a corresponding pin number in order for the system's lock to open. This Two-Factor Authentication has become industry standard for security of systems in the field and puts the Smart Door Security System among the most secure in the market today. And with the hands-free door unlock capability that is leveraged by speech-to-text pin entry, the homeowner has even greater accessibility that any standard deadbolt lock that must be unlocked using a physical key or physical pin number entry. With the advanced security features, and ease of accessibility, there truly is nothing like the Smart Door Security System out there today.

The "Smart Door Security System" Engineers:

Adam Stefanik will graduate with his Bachelor's degree in Electrical Engineering and a minor in Mathematics. He will be attending the University of Florida for his Master's Degree in Electrical and Computer Engineering starting in the Fall of 2021. He is interested in Analog Circuits and Optics and will be working as an engineer at a defense contractor after his UCF graduation.

Moisess Rodriguez will graduate with his Bachelor's degree in Computer Engineering. He will pursue his Master's degree in Electrical and Computer Engineering at the University of Florida beginning in Fall 2021. His area of study will be in Artificial Intelligence, and will be working full time as a software engineer for Northrop Grumman after graduating from UCF.

Reham Hammad is a senior graduating in August 2021 with a bachelor's degree in Electrical Engineering. She got accepted in the Electrical Engineering master program at the university of Central Florida and she is looking to pursue a carrier in the RF field.

Zach Jackovich will graduate with his Bachelor's Degree in Computer Engineering and is interested in Hardware & Systems Security. He will begin his Masters Degree in Electrical and Computer Engineering from the University of Florida in the Fall of 2021. He has also accepted an offer to work for Lockheed Martin as an ASIC & FPGA Design Engineer after graduation from UCF.