This is just an example of a previous work. Be sure to refer to the assignment document and rubric for full details of the requirements for this report.

Smart Doorbell with Internet of Things Application

Define Product Interfaces

Table of Contents

1. Introduction

The primary purpose of the document is to provide a detailed description for the individual modules of the final design of this team's project. This modular design approach will provide a framework on which to follow when implementing components of the final design in a way that the project will come together in the end as one cohesive design. The individual modules will be analyzed and reported in detail to provide a complete overview of the project.

Given the limited time the team has to complete the project, it is integral that the modular approach is adopted to ensure individual team members simultaneously contribute to the design. Dividing the final design into tasks that can be divided into smaller modules will ensure that every stage of the design process is accounted for by every member of the team. In doing so, the varying schedules of each member can be accounted for and tasks can be completed independently to make sure the project continues to be on schedule.

In its current state, the project is in various stages of development. Some modules have been completed, whereas others are just beginning. By overlapping the more complex modules with the simpler ones, the team can ensure the project continues to steadily approach the final deadline on pace. More complex modules have been given more time to account for potential problems and the team has allowed for time to troubleshoot and fix these potential issues. These potential hurdles have been identified in this report and plans have been laid to ensure the team rectifies any issue in a timely manner.

2. Architectural Description

This section shall begin by introducing the block diagram, which illustrates the ways in which the doorbell can be interacted with and how the different modules of the doorbell system and application connect to each other. This section will also detail how each module in the system functions.

2.1 Block Diagram

The detailed block diagram for the Smart Doorbell can be seen in Appendix A. Modules that are implemented using hardware are outlined in blue, while modules that have been implemented using software are outlined in red.

2.2 Descriptions of Each Module

This section shall explain the purpose of each module included in the block diagram, which can be seen in Appendix A. This section shall also explain how this module will fulfill customer requirements, as well as how the module will be implemented into the project.

2.2.1 Arducam OV5647 5MP Camera

The camera module will be connected directly to the Raspberry Pi. It will provide 1080p 60fps video capturing and streaming for the doorbell, which is a requirement for most customers. The camera provides the basis for most of the most of the doorbell's security features, like facial recognition or video streaming. The video from the camera will be streamed to a web server, which can then be accessed by the homeowner.

2.2.2 HC-SR501 PIR Motion Sensors

There will be two HC-SR501 PIR motion sensors that will be placed on the face of the doorbell. The purpose of these devices is to provide a sensing range that is capable of detecting any approaching visitors from several meters away. This satisfies the customers' requirement for a doorbell with security features. The sensors will interact with the doorbell application by sending a text message to the user's phone through the application should a sensor detect someone approaching the door. The use of this hardware will provide the house with the ability to provide security, and the interaction between the sensors and the application via the alert system will provide the homeowner peace of mind in knowing that the doorbell will alert their phone of any potential threats.

2.2.3 The Physical Button (Amazon Dash)

The Amazon Dash button will serve as the traditional doorbell button. This fulfills the basic and original functionality of any doorbell, regardless of the IoT features included with the doorbell. This button will be implemented in hardware to allow for guests to ring the doorbell of the house in the same manner as any other doorbell. The additional benefit of this doorbell is that it can be activated wirelessly. This will allow for the team to implement the wireless ringing functionality of the application, which is discussed in further detail in the next section.

2.2.4 Wireless Doorbell from Application

A simple button on the application can be pressed to ring the homeowner's doorbell wirelessly. This satisfies the customers' desire to have a doorbell that can be interacted with through the application, as this will allow friends and family members to ring the doorbell when they want to quickly notify the homeowner that they have arrived. This button needs to be implemented in software so that the doorbell can be rung through the application.

2.2.5 Facial Recognition Feature

The facial recognition is a software written in Python that will run on the Raspberry Pi. Users have the ability to register faces on the software's database and assign a name for the face. The facial recognition software will run when the camera is initiated, and if it recognizes a familiar face, the program will send a message to the owner's phone of who is at the door or if a stranger is at the door. This feature satisfies the security customer requirement, and it will provide the owner with the information that a close friend or a family member is visiting.

2.2.6 Localization Feature

The localization feature serves to limit the range from which an individual can ring the doorbell from the application. In this case, the localization feature only allows the doorbell to be rung through the application if the phone is currently connected to the same Wi-Fi signal that the doorbell is running on. This was an implicit requirement of the wireless ringing capability. While customers wanted to have a doorbell that could be rung through the application, it was evident that the doorbell should not be able to be rung from several miles away, as this could allow friends to annoy the homeowner by constantly ringing their doorbell from anywhere. Since this feature is based off of which Wi-Fi network the phone is connected to, the Localization feature must be implemented using software.

2.2.7 Doorbell Chime

The doorbell chime serves the traditional purpose of a typical doorbell chime. When the doorbell is pressed, a sound is emitted inside the house to notify the homeowner that someone is at the door. This satisfies the customers' basic requirement for what a doorbell should do. This chime will be implemented in software so that the tone can be sent out as an audio output when the doorbell is rung. This chime will sound for both physical presses of the doorbell and wireless presses of the doorbell through the application.

2.2.8 Wi-Fi Module in Raspberry Pi

The Wi-Fi module is a hardware device in the Raspberry Pi itself that transmits the sensory data and doorbell presses to the user application. This chip allows for the connection between the hardware and software, which satisfies many of the customer requirements, such as the text alert features, the communication system, and the transmission of video from the camera to the web server. The chip itself is a physical component native to the Raspberry Pi, which was one of the many reasons why the Raspberry Pi was the microprocessor of choice for this project.

2.2.9 Text Notification from Raspberry Pi

Along with security, ease of use was one of the main requirements from the customers. The text notification sends a text message to the user's cell phone if a motion sensor detects someone approaching the door, or if someone rings the doorbell by pressing the doorbell or by wirelessly ringing the doorbell. This system not only monitors and alerts the homeowner if any of these aforementioned events occur, but this notification system also link the user to the web server that live streams the audio and video from their front door. Since this notification system needs to alert the user's phone based on the physical inputs from sensors and the doorbell, the text notification feature will be implemented through software.-

2.2.10 Web Server for Audio and Video

The web server allows the homeowner to view any current activity happening on their front porch. Furthermore, when a motion sensor detects someone approaching the door, the text notification sent to the phone will provide a link to this web server so the homeowner can see who is coming to their door in real time. This feature further fulfills the customer requirement for safety. The module allows the homeowner to view their front door at any time, and the link to this server is provided in the notification so the homeowner can quickly determine who is at their door. This feature will be implemented in software so the user can access the live video and audio feed of their front door from the phone application.

2.2.11 Communication Structure

The doorbell will support a two-way communication structure. The homeowner will have access to audio and video of the guest at the door through the phone application, while the guest will only have access to the audio from the homeowner's phone. The homeowner is able to receive audio and video from the phone by taking in the visuals from the doorbell's camera and the guest's voice from the doorbell's microphone. The homeowner's voice is then outputted through the doorbell's speaker. This connection between hardware is essential to fulfilling the customer's requirement to communicate with guests who are at the door. Since the communication needs to occur between the doorbell and the phone application, a combination of hardware and software features are used. Since the communication is initiated by the homeowner's phone, and since the communication process is possible through software, this block in the block diagram was considered more of a software module than a hardware module. As a result, its border is red in the block diagram, which can be seen in Appendix A.

3. Detailed Descriptions of Modules

This section shall provide a more detailed explanation as to how each module in the Smart Doorbell will function. Each section shall explain how the module of interest will operate, the inputs and outputs for the module, and an explanation of tests that will be performed to ensure proper module functioning. If necessary, calculations, experiments, and derivations for module operation will also be explained here.

3.1 Arducam OV5647 5MP Camera

The camera is capable of recording videos at 1080p 60fps and is compatible with the Raspberry Pi. A camera to give the customer information of what is outside of the door is a necessary feature in any smart doorbell. The camera will be connected directly to the Raspberry Pi via the camera port, and can be activated through the picamera Python library or using the opencv library method VideoCapture, which will require the drive bcm2835-v4l to be activated through the command “sudo modprobe bcm2835-v4l”.

The camera module is connected to the Raspberry Pi via a 15-way ribbon cable. The bcm2835-v4l driver is already installed on the newest Raspbian Stretch operating system. The facial recognition software after running will initiate the camera with the cv2.VideoCapture method. An LED will light up whenever the camera is turned on and working. This module will be tested for quality and smoothness, and will be tested concurrently with the facial recognition program.

3.2 HC-SR501 PIR Motion Sensors

The incorporation of two HC-SR501 PIR Motion Sensors will allow for detection of approaching guests or intruders from up to 7 meters away from the doorbell. Security was customers’ primary concern with a smart doorbell; homeowners wanted to be notified not only when someone rang their doorbell, but also when potential threats approached the front of the house. These motion sensors shall connect to the Raspberry Pi and output a 3.3 volt signal to the GPIO pin the device is connected to when motion is sensed. When this voltage is sent to the GPIO pin, the doorbell will send a notification to the user’s phone that someone is approaching the house.

This module does not take in any inputs besides a 5-volt power supply from the Raspberry Pi. When connected to the power supply, the sensors shall be under constant operation. Motion can be sensed by these modules through the use of infrared light. When a warm object, such as a human, protrudes into its line of sight, the increased heat presence is detected by the sensor. The sensor will then output a 3.3V signal to the connected GPIO pin, which will trigger the alert system to the phone. If no motion is detected, no voltage will be outputted by the sensor.

Each motion sensor requires a 5-volt input voltage for proper operation. Furthermore, each sensor draws approximately 65mA [1]. Ohm's law can be used to find the relative impedance of these devices. Ohm's law is stated below in Equation (1).

$$V = I*R \tag{1}$$

Mathematically manipulating this equation to solve for R, we find that R = V/I. Plugging in 5 volts for V and 65mA for I, the impedance of these PIR motion sensors is found to be approximately 77Ω.

The circuit schematic for these motion sensors was created using EasyEDA. Each motion sensor uses a 5V0 pin and GND pin from the Raspberry Pi 3B. Finally, each OUT pin on the HC-SR501 must be connected to a GPIO pin. These pins do not require a specific GPIO pin; however, the speaker setup does require GPIO pins 12 and 18 (more on this in Section 3.11). To prevent interfering with other component requirements, the GPIO pins for these sensors were connected to GPIO21 and GPIO20, as seen in Figure 1 below.

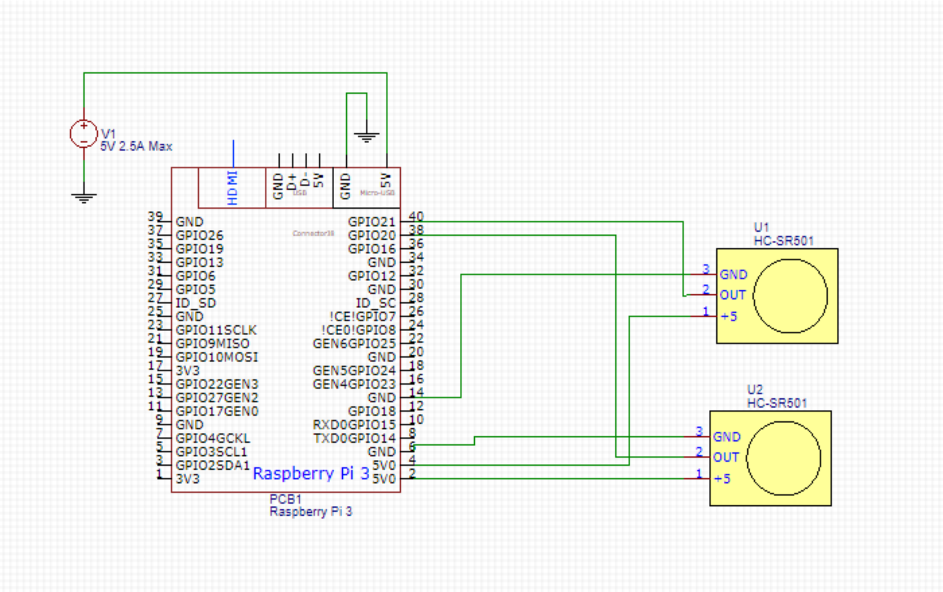


Figure 1 : Motion Sensors Circuit Schematic

Since these modules are fully assembled from a different manufacturer, a schematic of these sensors is included in Appendix B [1]. To ensure these modules are properly functioning, the motion sensor will be connected to the Raspberry Pi along with an LED. A Python script written to the Raspberry Pi will test the modules as follows: when a person stands in front of the motion sensor within a 7-foot, 100-degree range (as specified by its data sheets [1]), the LED should illuminate. When nothing is in front of the motion sensor, the LED should turn off. This will allow the team to understand if the module is functioning as desired. If the device incorrectly senses motion when none is present, or fails to detect motion when within the specified field of view, the sensitivity of the devices will be adjusted using the knobs on the front of the device.

3.3 The Physical Button (Amazon Dash)

The Amazon Dash is the physical button that guests can ring to trigger the doorbell chime. Ringing the doorbell will also send the text notification to the homeowner's phone that someone has rung their doorbell. The inclusion of this module allows the doorbell to serve its original purpose. When the doorbell is pressed, a sound will be emitted inside the house, which tells the homeowner that someone is at the door.

The Amazon Dash button can take inputs from physical button presses or from button presses over the application. Regardless of the way the doorbell button is pressed, this module will cause the doorbell chime to sound and the text alert to be sent from the Raspberry Pi to the homeowner's phone.

This module will be tested in two ways. First, the doorbell's response to a physical press will be tested. The team will press the doorbell to ensure that the chime sounds and the text alert is sent. Once the physical operation of the doorbell is found successful, the team will try to press the doorbell through the application. This test will be successful when the doorbell successfully causes the doorbell chime to sound and the text notification to be sent.

3.4 Wireless Doorbell from Application

The doorbell button on the application will improve ease of use for the guests, as well as satisfy an interest in the market for the ability to ring the doorbell through the application. By pressing the doorbell icon on the phone application while the phone is connected to the same Wi-Fi as the doorbell (for more information on this function, see Section 3.6, Localization Feature), the doorbell will emulate the same process as if the doorbell button were physically pressed. The chime will sound in the house, and the text notification will be sent to the homeowner's phone that someone rang the doorbell. This feature will allow for friends and family of the homeowner to ring the doorbell when they arrive at the house. One possible application for this feature could be for a carpooling group; using this feature, the driver could alert the homeowner that they have arrived without having to walk up to the door to ring the doorbell.

This module's only input is the button press from the application. Once a press has been registered by the button on the application, a command will be sent to the doorbell to cause the doorbell to chime. Along with this auditory output, the text notification will also be generated and sent to the homeowner's phone to alert them that the doorbell had been rung.

This module and the localization feature will be tested simultaneously. With the doorbell and the phone with the doorbell application in the same Wi-Fi region, the doorbell button will be pressed. Should the wireless doorbell button function as intended, the doorbell will chime, and the homeowner will receive an alert from their phone. Should either the doorbell or the text notification (or both) fail to perform their respective functions following this button press, the team will debug the system's code to determine and fix the issue. Once ensuring that the doorbell can be rung through the application when both are in the same wireless zone, the team will try to ring the doorbell when the phone has left the Wi-Fi zone. Pressing the doorbell should fail to generate both the chime and the text notification. Should either of these modules perform their intended functions, the team will investigate and debug the wireless connection issue.

3.5 Facial Recognition Feature

The facial recognition software is written in Python and is installed on the Raspberry Pi. It was written by and forked by with the authors' permission [2].

This software uses the Multi-task Cascaded Convolutional Networks (MTCNN) method for face detection, created by Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao [3]. This method find faces in three stages.

- First stage: Obtaining multiple windows in a frame that might be a face and combine windows that mostly overlap using non-maximum suppression (NMS)
- Second stage: Filter the windows found in the first stage, reject false candidates and use NMS again to receive only a few likely candidates for a face.
- Third stage: Identify faces from the windows remained from second stage, and output five facial landmark positions (two eyes, tip of nose, and two mouth corners) [3].

MTCNN method allows faces with different expressions and lighting to be detected, and also allows multiple faces to be detected in one frame efficiently.

The facial recognition architecture that this software uses is Facenet Inception Resnet V1, which utilize Google's Tensorflow library for training the models. In this specific implementation, the software will be using pre-trained models provided by David Sandberg [4]. This architecture and model have been tested for accuracy with Labeled Faces in the Wild (LFW), a database of face photographs designed for testing facial recognition, and have granted a 99.65 ± 0.252% accuracy [4].

There are two modes in the software: facial recognition mode and input mode. Facial recognition mode will initiate the camera recording and facial recognition, while input mode will allow user to register new faces on the database. The input mode store facial landmarks as coordinates in a text file, categorized into three types: left, right, and center; these facial landmarks will then be used for the facial recognition. Both modes can be initiated easily as a Python script on the Raspberry Pi; the facial recognition mode is run with the command "python3 main.py", the input mode is run with the command "python3 main.py --mode "input"". A demo of the program can be seen in Figure 2 below

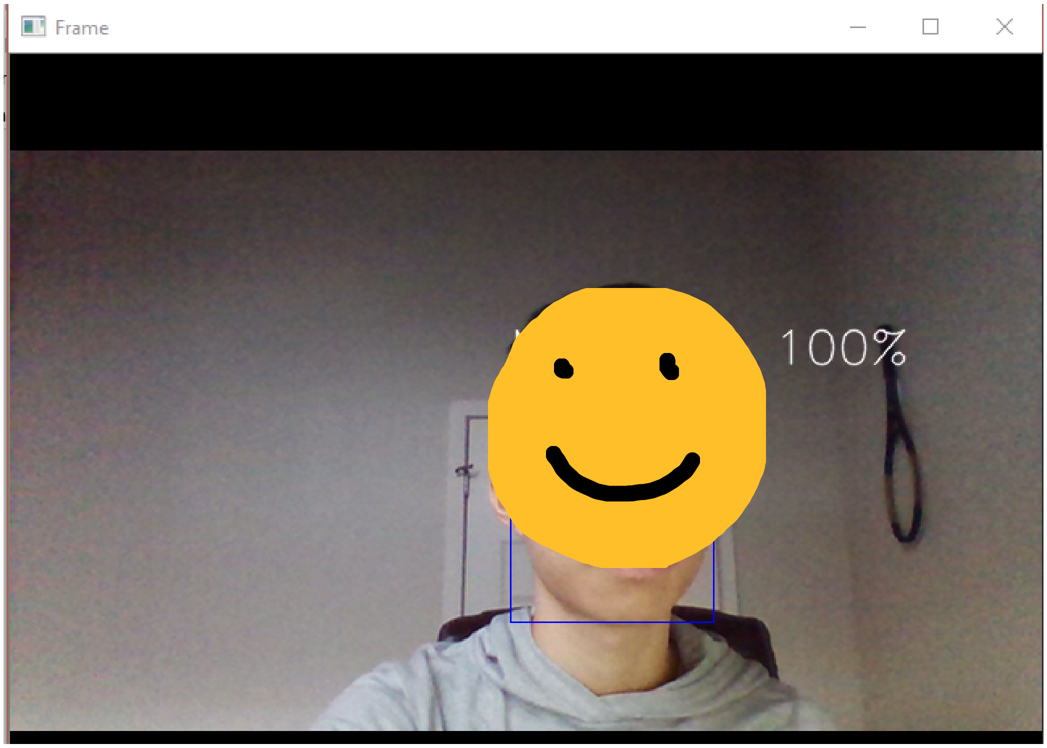


Figure 2: Facial Recognition Program Demo

The face detection will run after the camera starts recording, and after a face is detected on the screen, the program currently draws a box around the recognized face with the percentage of how similar the program thinks the face is to one of the faces stored inside the database, and the facial recognition feature will be initialized. If the face is more than 80% similar, the program will consider the face on the screen is similar enough to the one stored in the database. The Raspberry Pi will then send user a text notification that someone familiar is at the door. If a face is detected but not recognized as familiar, the Raspberry Pi will send the user a text notification that a stranger is at the door.

3.6 Localization Feature

The localization feature is a built-in feature of the wireless doorbell button on the application. This feature limits the range from where the doorbell can be wirelessly rung through the application. The intended "region of operation" for the wireless doorbell button will be the Wi-Fi zone of the house under which the doorbell is connected to. When the user's phone is connected to the same Wi-Fi as the doorbell, wireless ringing can occur; otherwise, the phone will be unable to ring the doorbell. This will satisfy the concerns that customers have about wireless ringing capabilities. With this localization feature in place, homeowners do not need to worry about annoying "false alarm" doorbell rings from several miles away.

Since this feature is a built-in functionality to the doorbell button on the application, this module does not take in any inputs. This module's output is based on whether or not the phone is connected to the same wireless network as the doorbell. If so, the wireless doorbell will successfully sound the doorbell chime and send the alert message to the homeowner's phone. If the phone is out of this Wi-Fi range, the button will produce no outputs.

As mentioned in the wireless doorbell application, the localization feature will be tested in two ways. First, the team will connect the phone to the same network as the doorbell. The team will then press the doorbell button on the app to see if the doorbell chime properly sounds and if the text notification is sent. If both of these actions occur, the team will move to another region in the Wi-Fi zone and perform another test. Testing will continue until it is evident that the doorbell can be rung from anywhere within the Wi-Fi zone. The team will then connect the phone to a different wireless network and attempt to ring the doorbell. If neither the chime or text notification transaction is triggered, the team will test another area off of the Wi-Fi grid to see if the same result occurs. Testing will be deemed successful when the team concludes that the doorbell cannot be rung when the phone is not connected to the same Wi-Fi as the doorbell. If either the doorbell chimes or the notification is sent, the team will remedy the undesired connection issue.

3.7 Doorbell Chime

The doorbell chime will perform the basic function of a typical doorbell; once the doorbell is pressed, the chime will sound in the house to tell the homeowner that someone is at the door. Without this basic functionality, the doorbell would fail to fulfill its original purpose.

The doorbell chime will take in the state of the doorbell press (pressed or not pressed) and output a sound inside the home if the doorbell button registers that the button was pressed. This button press can occur either through a physical press or through the wireless doorbell ringing functionality on the phone application.

The doorbell chime will be tested by ensuring that the chime properly rings when the doorbell button registers a press. When this button press is registered, the doorbell button should cause the doorbell chime to sound. When the doorbell button does not register a press, the chime should be silent.

3.8 Wi-Fi Module from Raspberry Pi

The Wi-Fi module is an embedded chip inside the Raspberry Pi 3B that will be used for communication between the doorbell and the phone application. The features that will use the Wi-Fi module include the text notification system, the two-way communication system, the wireless doorbell ringing feature, and the web server. Without this module, the doorbell would be unable to have any connection to the Internet.

The Wi-Fi chip will take in the commands written in code that state what the Raspberry Pi must do given the sensor readings. For example, if a motion sensor detects someone approaching the front door, the embedded code of the Raspberry Pi will require the Raspberry Pi's Wi-Fi chip to output the text alert to the homeowner's cell phone. Since there are several possible reasons as to why the Wi-Fi chip may be used, the Wi-Fi module's output varies. The notification system will output the text alert, the communication system will output the audio from both the phone to the doorbell and the audio and video from the doorbell to the phone. The wireless doorbell ringing feature will output a command to the doorbell button to notify the system that someone rang the doorbell, which will then prompt the chime to sound. Finally, the web server will continuously output the video from the camera and the audio from the doorbell's microphone.

Since the Wi-Fi chip is already included in the Raspberry Pi, the team will test the chip in parallel with testing every one of the aforementioned wireless transactions. Testing for the Wi-Fi module will complete once the team successfully completes testing for each of the modules that communicate between the doorbell and the phone application.

3.9 Text Notification from Raspberry Pi

The text notification system is essential in providing quick, simple alerts to the homeowner, notifying them that someone rang the doorbell or motion was detected by the HC-SR501 PIR sensors. This feature addresses the customers' main concern with security, as this system takes in the sensory inputs from the device, and warns the user of any potential threats to the household. This monitoring system provides users with peace of mind, knowing that the doorbell is patrolling the front entry to the house. This alert system is also essential because it increases the ease of use for the homeowner. The text alert not only notifies the homeowner of whether it detected motion or a doorbell press, but it also provides a link to the web server that live streams the audio and video from the camera and microphone built into the doorbell.

The text alert system is generated based on the information from the motion sensors and the doorbell. When either the motion sensors detect someone approaching or the doorbell is rung (either physically or through the application), the text alert system will generate a message to the homeowner, detailing which functionality occurred. For example, if the motion sensors are what cause the text notification to be sent, the alert will explain that motion was detected close to the home. Along with this notification, the alert system will also output a link to the web server, which allows the homeowner to quickly access the live video and audio feed from the camera and microphone. This will allow the homeowner to tell who is approaching the doorbell or who rang the doorbell. The text notification system will be tested to ensure that the notification is successfully and quickly sent to the homeowner's phone when the doorbell system detects a doorbell press or a PIR sensor detects motion within its field of view.

3.10 Web Server for Audio and Video

The Web Server is the final module that satisfies the security concerns for the customer. This server is continuously running, allowing homeowners to access the video and audio from their front door whenever they wish by simply clicking on a button in the application. This ability to view the front door whenever the homeowner wishes will make the homeowner confident that their front entrance is being protected and monitored at all times of the day.

This module takes in the sounds and noises from the USB microphone as well as the 1080p HD video stream from the Arducam camera. These two sources are displayed on the server. The server can be pulled up at any time through a button on the application. The server will also be linked in every text alert message sent to the homeowner’s phone so that the user can quickly see who is at their door.

A snippet of video from the livestreamed server can be seen in Figure 3 below. As evident from this image, the homeowner is able to clearly see and recognize whoever may approach the door. The camera module runs at 30 frames per second, which will allow for crisp visuals to be displayed at all times by the web server.

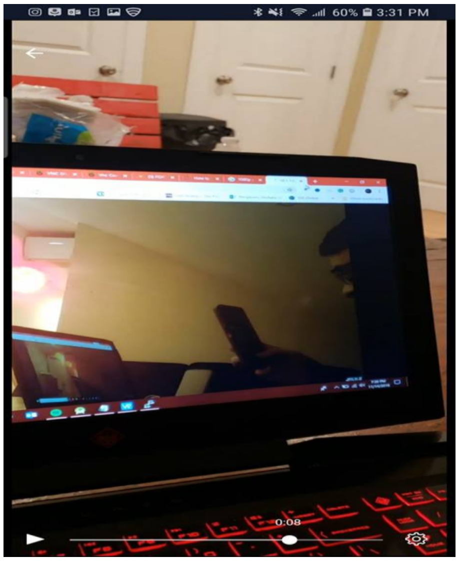


Figure 3: Video Livestream Demo

Testing for the web server has been underway by After writing the code necessary for the server, tested the visuals of the doorbell camera by running his scripts. To ensure that the video was running continuously and live streaming what the camera was seeing, began to move the camera module around his room. The web server verifies that the code and camera were properly functioning, as the web server displayed the new surroundings as the camera was adjusted. Once the microphone module arrives, will be able to ensure that the web server is also capable of outputting audio as expected.

3.11 Communication Structure

This module is responsible for satisfying the customers’ desire to communicate with guests, whether or not the homeowner is at home. This communication consists of hardware and software components, which will combine to create a two-way communication between the phone application and the doorbell itself. The phone application will feature a series of buttons that will allow the homeowner to enable doorbell audio and video, as well as the option to start a conversation between the guests at the door. On the doorbell, a microphone and speaker system will enable voice communication between the homeowner and guest. This feature will be useful in several applications, such as when a homeowner needs to tell a UPS worker where to leave a package that is being dropped off.

For the application, the necessary inputs include the audio from the microphone, as well as the video from the camera module. These two inputs will be outputted to the homeowners' application, creating a "Facetime"-like video conversation. The doorbell will take the audio from the phone's microphone and output these voices through the doorbell's speaker. Once the homeowner enables the video conversation from their phone, the two-way communication will begin. This will allow for the homeowner and guest to communicate with each other verbally. This functionality will be aided with the OpenTok communication system, which was once a competitor to the Facetime protocol.

The hardware setup for the communication structure can be seen in Figure 4 below. The microphone is an MI-305 USB microphone, which can be plugged into one of the four USB ports on the Raspberry Pi 3B. This is simply modeled as a Microphone USB on the circuit schematic, which is directly connected to the Raspberry Pi. The speaker system for audio output consists of a PAM8302A audio amplifier and a 40mm, 4Ω, 3 watt speaker. The pin layout for the PAM8302, as well as a typical application for this device in an audio setup, is listed in Appendix C [5]. To acquire analog audio output from the Raspberry Pi 3B, Channel 0 for the audio output must be used. Channel 0 can be accessed through GPIO12 and GPIO18 on the Raspberry Pi 3B. The positive input of the amplifier will be connected to GPIO12, and the negative terminal of the amplifier will be connected to GPIO18. The V_{DD} pin on the PAM8302A will be powered on a 3.3-volt source from a power pin on the Raspberry Pi. The amplifier's ground pin will be connected to one of the many GND pins on the Raspberry Pi. As listed by the specification sheets, the PAM8302A is capable of outputting between 0.86 watts and 1.25 watts to a 4Ω speaker when powered by 3.3 volts. The speaker, whose positive and negative inputs will be connected to pins 5 and 8 on the amplifier, respectively, is capable of a maximum of 3 watts. This verifies that the speaker and amplifier will be able to function without fear of damage. Using power equations, the range of voltages and currents that the speaker will operate on can be determined by manipulating the following equations:

$$P = V^2 / R \tag{2}$$

$$P = I^2 * R \tag{3}$$

Solving Equation (2) for voltage, it can be determined that the speaker will support voltages between the range of 1.85 volts and 2.24 volts. Solving Equation (3) for current, it can be determined that the current through the speaker will be between 463mA and 559mA.

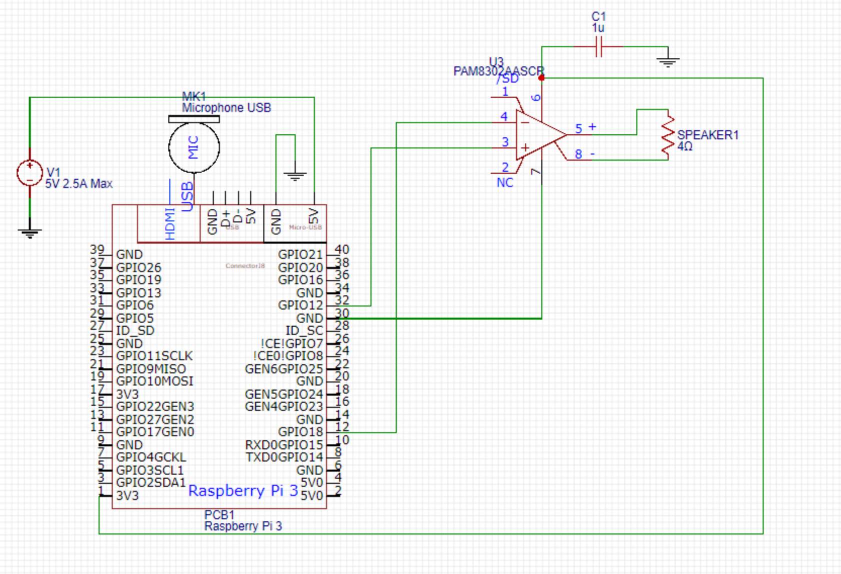


Figure 4: Communication Structure Circuit Schematic

The team has already tested the video streaming capabilities to ensure that the homeowner can see who is at the door. has shown that he is able to display live video feed from the camera module when he was designing the web server. To ensure that both sides of the audio work as intended, initial tests shall be performed on the speakers and microphone to ensure that audio can be outputted from the microphone to the phone the audio can be inputted from the phone to the speaker. Once the audio communication has successfully relayed the voices of both parties to the other, the audio and video systems will be combined to ensure that both systems can run simultaneously without issue.

The full schematic for the doorbell can be seen in Appendix D. The full schematic was segmented into two pieces to more clearly demonstrate how the hardware for the PIR motion sensors and the communication structures will be implemented.

4. System Integration

This section shall discuss how the hardware and software modules for the Smart Doorbell will be integrated and tested to ensure proper functioning in the final design. Finally, the system testing as a whole will be discussed, with explanations as to how to determine if the product functions properly

4.1 Arducam OV5647 Integration and Testing

The camera is connected to the Raspberry Pi through a 15-way ribbon cable. The camera will be tested for smoothness and ability to record video at 1080p, as well as being compatible with the facial recognition software. All of the criteria above have been tested with satisfactory result.

4.2 HC-SR501 Integration and Testing

To satisfy the primarily goal of the safety, both HC-SR501 sensors must successfully detect motion within the 100-degree, 7-meter field of view. When motion is detected, these modules will need to be capable of sending the 3.3-volt signal to the GPIO pin to trigger the text alert to the user's cellphone. Furthermore, these sensors need to be able to avoid "false alarms" and incorrect motion detection. These sensors must be calibrated properly to ensure that only valid motion of a person is being detected and sent to the application. As a result, these sensors need to not only function properly, but also need to properly connect to the notification system implemented in software.

4.3 Physical Doorbell Integration and Testing

The Amazon Dash doorbell needs to detect when a user has pushed the button physically or when the doorbell has been rung through the application. The team will test the functioning of the doorbell by ensuring that both physical and application-based doorbell buttons can cause the doorbell button to register a ring. From there, the button will need to cause the doorbell chime to sound inside the house. This will require the Amazon Dash to be integrated into both the doorbell button from the application, as well as the doorbell chime.

4.4 Wireless Doorbell from Application Integration and Testing

The wireless doorbell button must successfully “ring” the doorbell button when the phone is connected to the same Wi-Fi network as the doorbell. Furthermore, the doorbell button must fail to ring the doorbell when the button is not on the same Wi-Fi network. To test to ensure these requirements are met, the team will ring the doorbell with the application while the phone is both on and off of the same Wi-Fi as the doorbell. This experimentation will be successful when it is determined that the doorbell can only register a ring when the phone is connected to the same network as the doorbell. This requires the doorbell button on the app to be integrated with the doorbell button as well as the localization feature so that the doorbell only rings when the phone is within the restricted boundaries of the Wi-Fi region to which the doorbell is connected.

4.5 Facial Recognition Feature Integration and Testing

The facial recognition program must be tested to be able to run well on the Raspberry Pi 3B without too much performance issues. The program also must recognize face stored in the database with a reasonable accuracy, and not mistaking unfamiliar faces with familiar faces.

4.6 Localization Feature Integration and Testing

The localization feature must provide the necessary restrictive features to the wireless doorbell button to prevent people from being able to ring the house’s doorbell from anywhere. This module will need to ensure that the doorbell can only be rung when the phone from where the button is being pressed is currently connected to the same Wi-Fi network as the doorbell. This will require the localization feature to be connected to both the wireless doorbell button on the app, as well as the Amazon Dash button.

4.7 Doorbell Chime Integration and Testing

The doorbell chime must successfully produce a sound inside the home when the doorbell is rung, either physically or wirelessly through the application. This module is necessary for providing the basic feature of any doorbell, which is the ability to alert the homeowner that someone is at the door with a sound. The team will test the chime by ensuring that it properly rings when the doorbell button detects a button press. This chime must be integrated with the doorbell button so that the chime will sound inside the home whenever the doorbell button registers a button press.

4.8 Wi-Fi Module from Raspberry Pi Integration and Testing

The Wi-Fi chip from the Raspberry Pi must successfully transmit the proper output from the doorbell to the phone application based on the sensor readings from the doorbell. Furthermore, the doorbell must also transmit all audio from the phone to the doorbell's speaker during the two-way communication between the homeowner and the guest at the door. The team will test to ensure that the Wi-Fi module is working as desired by ensuring that all sensory readings from the doorbell send their respective outputs to the application. The team will also test the Wi-Fi module's ability to send information from the application to the doorbell while testing and debugging the two-way communication system.

4.9 Text Notification from Raspberry Pi Integration and Testing

The text notification system is the backbone for the monitoring and security features of the doorbell. This system must be very reliable and quickly send alerts to the user's phone whenever a doorbell press occurs or when a motion sensor detects someone approaching the front door. The system will be tested by triggering these three different inputs to determine if the notification is quickly and successfully sent to the user's phone. This module will need to be integrated with the motion sensors and the doorbell button so that the system can read and react to the outputs of these two devices. Furthermore, the web server must be linked to each alert, so the team will test to ensure that the notification includes a valid link to the livestream footage from the camera.

4.10 Web Server for Audio and Video Integration and Testing

The web server will need to be operational and durable to provide continuous, 24/7 video and audio of the front door. This module will be tested by connecting the camera and microphone to the Raspberry Pi and ensuring that both video and audio recordings are properly emulated to the server. Furthermore, the team will need to test that the link to the web server is always provided in the notification system so that the user can quickly access footage from their front door. This will require the web server to be integrated with the microphone and camera modules from the doorbell, as well as having a link that is embedded in the text notification module.

4.11 Communication Structure Integration and Testing

The communication structure must provide reliable two-way audio to both parties, as well as video of the front door to the phone application. The communication structure will be tested by ensuring that the phone application is capable of receiving both clear video and audio from the doorbell’s camera and microphone. Furthermore, the team will test to ensure that the speaker is able to properly output any audio from the phone’s microphone and output these signals to the speaker so that the guest can clearly hear the homeowner from the doorbell. This system will be the most integrated module in the system. The guest-to-homeowner side of the communication structure must connect the microphone and camera inputs to the web server and phone speakers so the homeowner can clearly see and hear the guest. For the homeowner-to-guest side of the communication, the phone’s microphone must send the captured audio to the doorbell, where the amplifier and speaker of the doorbell can audibly output the homeowner’s voice to the guest at the door.

5. Conclusion

In conclusion, the project is on pace to be completed on time with all modules completed. Currently, the camera and Raspberry Pi are completely functional and can run the facial recognition software, albeit very slow, but functional nonetheless. In addition to this, the web server is completely functional and accessible from any device with a web browser on the same Wi-Fi. The phone application is still in development, but has a user interface and the appropriate SDKs imported. Hardware components, including the Amazon Dash button, microphone, speaker, and audio amplifier have shipped and are currently being implemented. The amplifier has been soldered to the appropriate components and audio testing is currently being performed.

Implementing the Wi-Fi button on the phone application could potentially hinder the schedule of the project. Also, implementing the OpenTok SDK with the web server in place could pose to be an issue. The team may have to implement a different web server and look into alternative methods of implementing a communication protocol. Implementing audio through the GPIO pins could also prove to be fairly difficult, and could put the project off schedule. The last potential hindrance is getting all of the modules to work harmoniously in the final project. The team has full confidence that all modules can be implemented individually, however it may prove to be difficult to get all modules to work together.

In the event that any of these modules prove to be difficult, the team has a plan of attack to get the project back on schedule. If the application proves to be a problem, developing a web application may be more feasible given the time restraints. If the web server does not work, then implementing a server more compatible with the OpenTok SDK may be a better option. If the audio is an issue, using the 3.5” audio jack for audio output may be an alternative worth looking into. Lastly, if all features work, but not together, it may be more feasible to run a demo showing all modules working instead of spending hours automating the modules together. Ideally, none of these potential problems will arise, and the team will be able to make their vision a reality. Regardless, this team has planned accordingly the best possible way to implement every module of this project within the scope of each member’s individual abilities and will deliver a functional, polished product that performs all the functions outlined in this report.

6. References

[1] D. Vu, “A Tensorflow implementation of Facial Recognition in Python,” (2017). GitHub repository, https://github.com/vudung45/FaceRec

[2] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, 2016.

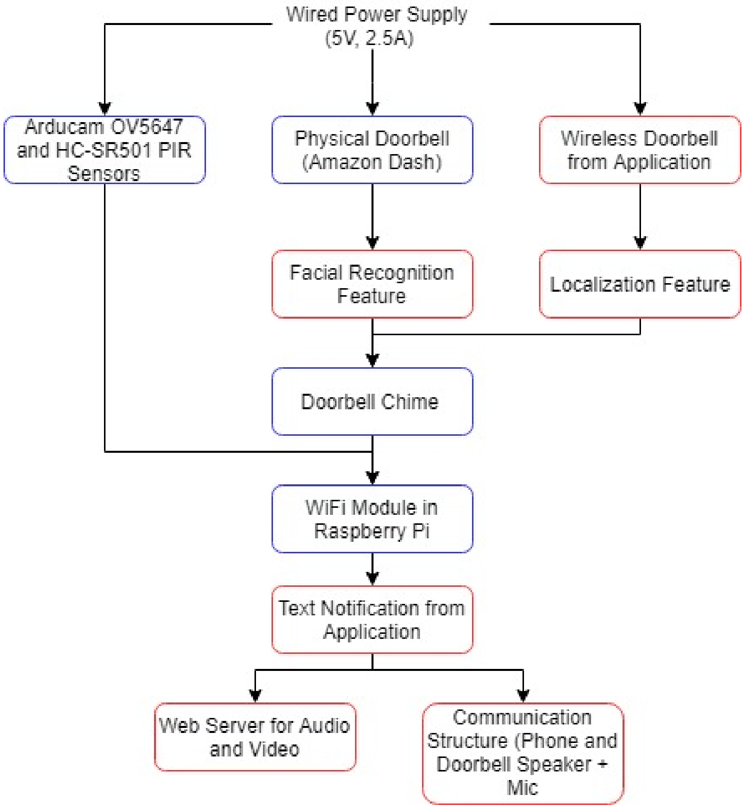
[3] D. Sandberg, “Face Recognition using Tensorflow,” (2017). GitHub repository, https://github.com/davidsandberg/facenet

[4] Oiyagai, “HC-SR501 PIR Motion Detector,” Component Datasheet, (Nov. 14, 2018), pp.1-2, https://www.mpja.com/download/31227sc.pdf

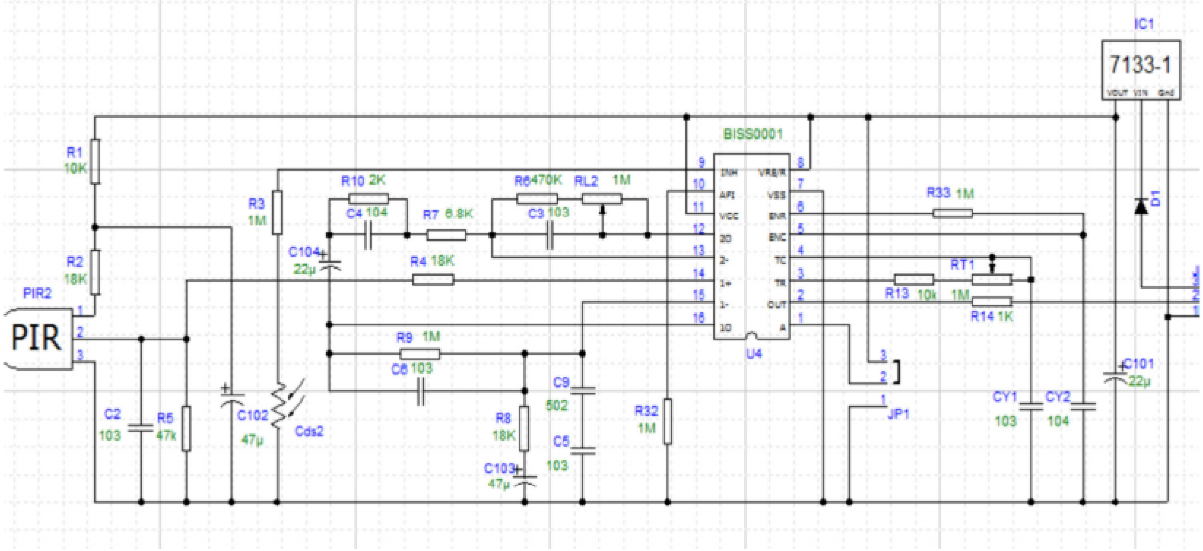
[5] Diodes Inc, “PAM8302A 2.5W Filterless Class-D Mono Audio Amplifier,” Component Datasheet, (Nov. 14, 2018), pp.1-2. https://www.diodes.com/assets/Datasheets/PAM8302A.pdf

Appendix

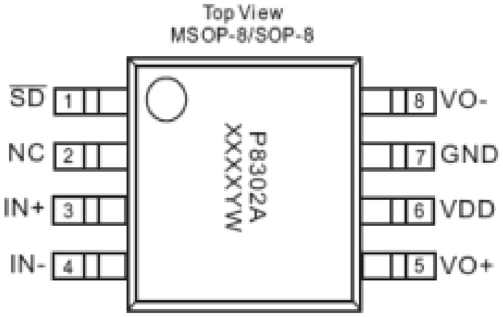
Appendix A: Block Diagram of the Smart Doorbell

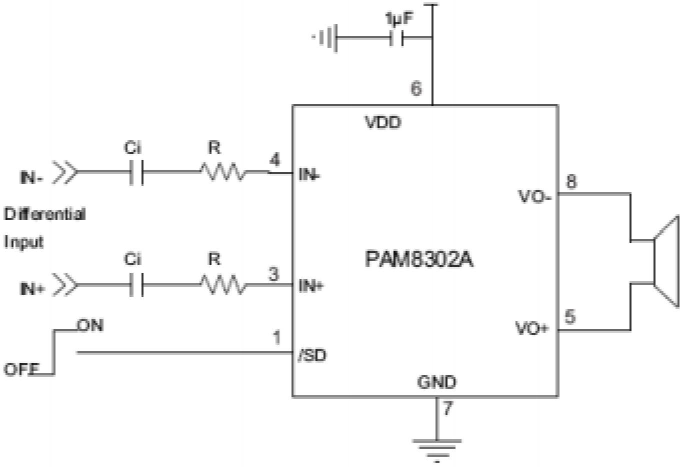

Wired Power Supply (5V, 2.5A)
Arducam OV5647 and HC-SR501 PIR Sensors
Physical Doorbell (Amazon Dash)
Wireless Doorbell from Application
Facial Recognition Feature
Localization Feature
Doorbell Chime
WiFi Module in Raspberry Pi
Text Notification from Application
Web Server for Audio and Video
Communication Structure (Phone and Doorbell Speaker + Mic

Appendix B: Circuit Schematic for the HC-SR501 PIR Motion Sensor

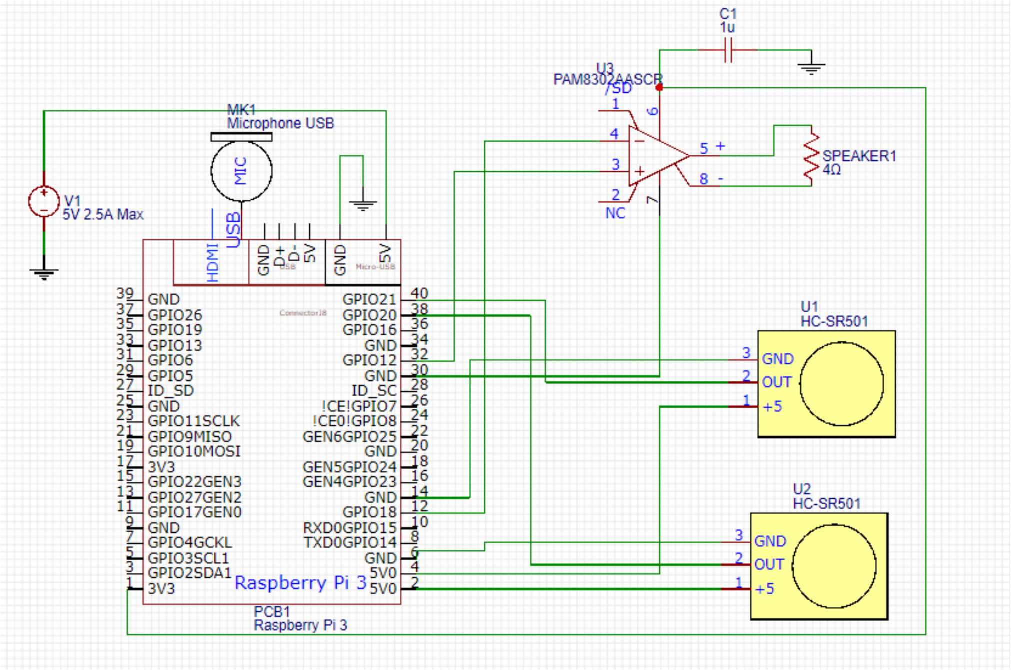


Appendix C: PAM8302A Audio Amplifier Pinout and Typical Audio Application





Appendix D: The Full Hardware Schematic


C1
1u
U3
PAM8302AASCR
/SD
1
6
4
3
2
5 +
8 -
7
NC
SPEAKER1
4Ω
MK1
Microphone USB
MIC
USB
V1
5V 2.5A Max
HDMI
GND
D+
D-
5V
GND
5V
Micro-USB
39 GND
37 GPIO26
35 GPIO19
33 GPIO13
31 GPIO6
29 GPIO5
27 ID_SD
25 GND
23 GPIO11SCLK
21 GPIO9MISO
19 GPIO10MOSI
17 3V3
15 GPIO22GEN3
13 GPIO27GEN2
11 GPIO17GEN0
9 GND
7 GPIO4GCKL
5 GPIO3SCL1
3 GPIO2SDA1
1 3V3
Connector28
GPIO21 40
GPIO20 38
GPIO16 36
GND 34
GPIO12 32
GND 30
ID_SC 28
!CE!GPIO7 26
!CE0!GPIO8 24
GEN6GPIO25 22
GND 20
GEN5GPIO24 18
GEN4GPIO23 16
GND 14
GPIO18 12
RXD0GPIO15 10
TXD0GPIO14 8
GND 6
5V0 4
5V0 2
Raspberry Pi 3
PCB1
Raspberry Pi 3
U1
HC-SR501
3 GND
2 OUT
1 +5
U2
HC-SR501
3 GND
2 OUT
1 +5