smartGuard

Bridges, Wilson

Alzate, Santiago

Florida Atlantic University

School of Engineering

Boca Raton, FL, 33431 USA

[Wbridges2014@fau.edu](mailto:Wbridges2014@fau.edu)

[Salzate2016@fau.edu](mailto:Salzate2016@fau.edu)

Professor: Dr. Alhalabi

**Abstract:** A complete smart, low cost and easy to use smart home door bell system was crated with a Raspberry Pi, Google SDK, Twilio API, and a web-based application. The two-way communication was achieved thru Wi-Fi.

**Index Terms:** The following abbreviations can be found throughout this document: API, UX, SDK, RP3

API: Application Programming Interface

UX: User Experience

SDK: Software Development Kit

RP3: Raspberry Pi 3

# **1 INTRODUCTION**

## **1.1 Description**

Our idea came from wanting to incorporate different kinds of technology solutions that are currently in the market into one. We did extensive research on smart doorbell systems as a form of home security and found many doorbells that have a lot of smart capabilities expect one with facial recognition, voice control and ability to unlock and lock door remotely. Most products in the market give you the ability to get alert of motion on your door, and allow you to speak and see the person at your door.

Using Google Assist, we were able to integrate voice commends to our project. This would allow for our doorbell to have a communication with guest as they’re on the door. As the person speak to Google Assist, a picture is then taken of the guest and texted to owner. We were able to achieve this by using a third-party micro service called Twilio. This text message also gives owner a link to view live feed of the person over the web and allow owner to lock or unlock their door.

We hope to inspire others by making Our project completely open sourced and will be available on Github soon. This will allow for others to make improvements to the project and grow the vision that we have. Also, we understand that new products and services are coming out every day and we want future generations to add those services to our product to make it better. We are also going to continue working on this product and move our authentication process from a local network based to a cloud-based solution. This would allow for diversification of connectivity from any network.

## **2 Experimental Details**

Our experimental process was done using a divide and conquer process. Santiago worked on the Raspberry end to make sure there was communication between Raspberry and WIFI and I made sure our Web Application was able to detect Live Feed. Our backend consisted of Python Script to run Google Assist, Listen for Trigger to turn on LED, Scripts for Voice Recognition commands and facial recognition.

Port forwarding was set up to allow access to live stream from anywhere outside of the Local Network

Our middleware was Google Cloud Services for Google Assist and Twilio REST API for sending Text message to owner that someone was at their door.

We used a specific code word to trigger the Text Message, we set it up that when the Guest says “Hello,” Twilio would then send the text message alerting owner that someone was their door.

Raspberry Camera was set up to turn on once someone approached door using motion and would wait 5 seconds before snapping a picture of the guest

Text message had link to web application that would allow owner to open link and see live feed as well as give the owner ability to lock or unlock door completely remotely

**Google:** After the MicroSD is flashed with the Raspbian image, it’s not needed to install the google libraries since there’re included in the image. To start testing the google assistant, it’s required to generate the google APIs. To do so, we have to go to <https://console.cloud.google.com/> and login with a google account. Then, we must enable the APIs and setup an authorization method. After that, we can generate a json and download it into the pi. The assemble instruction also includes step-by-step instructions on how to setup google APIs.

**Twilio:** we are using Twilio APIs to send text messages from the Pi. To setup the API, you have to create a Twilio account and choose a new phone number. They charge $0.08 per SMS and $1 per month per Twilio phone number, but they give $15 free to new customer. Once you have your account, we can see your ID and authorization token that you need to include in your code. Then, you have to install the Twilio library for python in your Pi**: “sudo pip3 install twilio**.” <https://www.twilio.com/docs/sms/tutorials/how-to-send-sms-messages-python>

**Picamera library:** To trigger the camera from python, we have to install the “Picamera” library: “**sudo apt-get install python-picamera python3-picamera”**

**2.1 Hardware and Communication Details**

The hardware we used was a Raspberry Pi 3 which included a microphone and WIFI capabilities, Speaker, integrated Raspberry camera and a button LED. All our communication was done thru WIFI, live feed using TCP/IP and Port Forwarding and our Twilio SMS is a REST API service done thru a microservice which was accessed using a Twilio Rest Library in our Raspberry Pi using Python.

**Pi as a web server:** To connect our web app with our Pi, we have to setup our Pi as a web server. To do so, we have to install apache: “**sudo apt-get install apache2 -y.”**

Once the web server is working, we can transfer our web app files to /var/www/html. Then, we can open the web app from any device connected to the same network typing the Pi IP.

**WiringPi:** To control the LED from our web app using PHP, we need the library “wiringPi” and git-core: **sudo apt-get install git-core**, **git clone git://git.drogon.net/wiringPi,** and then to build “wiringPi”**: cd wiringPi, ./build.**



1. **The AIY voice kit:** the AIY voice kit was designed by google and it let lets you build your own natural language processor and connect it to the Google Assistant or Cloud Speech-to-Text service, allowing you to ask questions and issue voice commands to your programs.

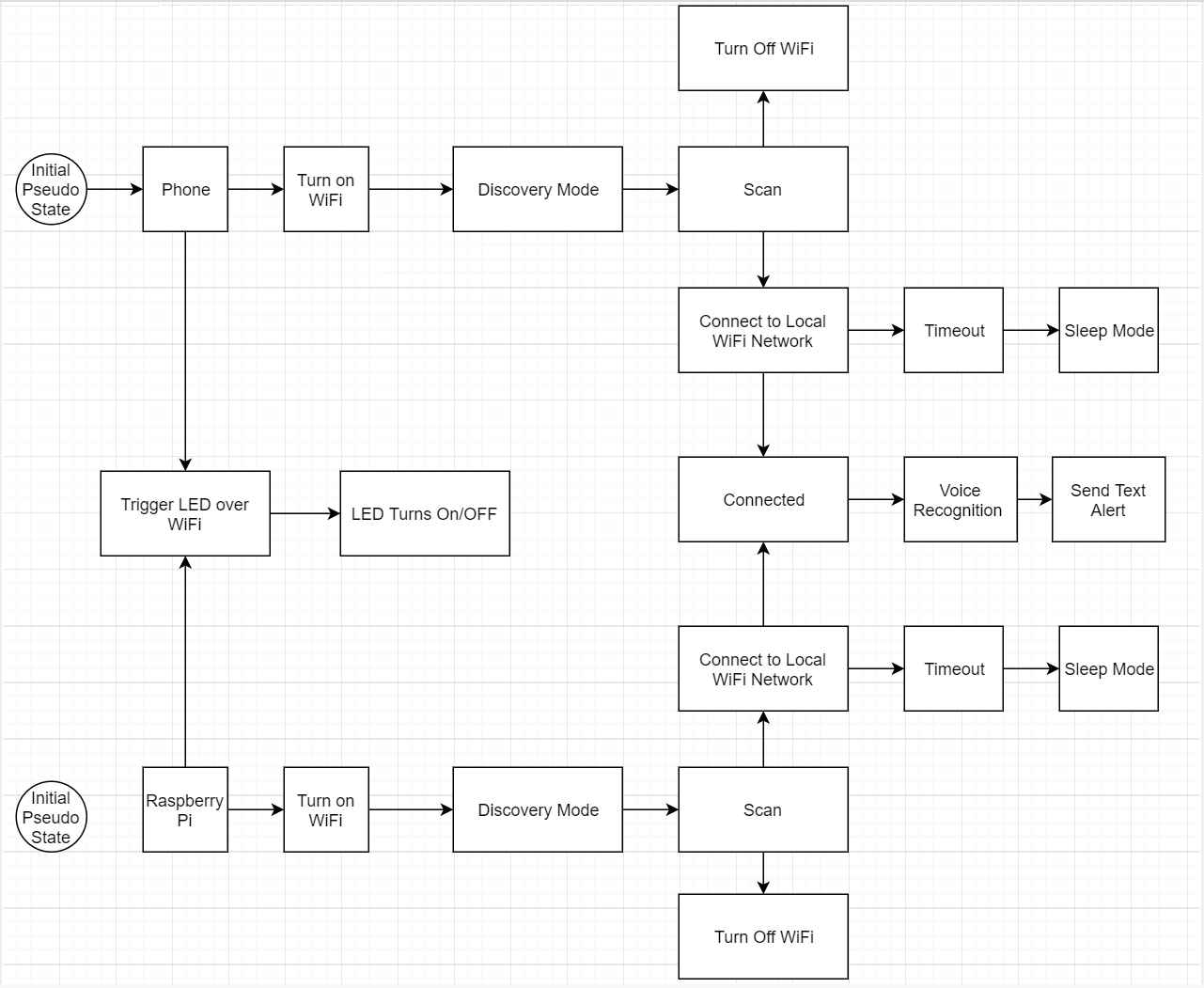
**Parts:**

* Voice HAT accessory board
* Voice HAT microphone board
* Plastic standoffs
* 3” speaker (wires attached)
* Arcade-style push button (Also a LED)
* 4-wire button cable
* 5-wire daughter board cable
* External cardboard box
* Internal cardboard frame
* Raspberry Pi 3 (not Included in the kit)
* SD card (not Included in the kit)
* Pi camera (not Included in the kit)

## **3 Results**

Our demo was a success, we demonstrated in class the ability to have a two-way communication over WIFI between Raspberry and Application. The Raspberry Successfully processed voice command to trigger a text message, motion to detect someone at door and turn on camera. Algorithm successfully waited 5 seconds to take picture and text to owner. Owner successfully received text, was able to open text and view live feed and picture, as well as triggering LED to turn on and off to mimic unlocking and locking door

**3.1 State Diagram**

****

## **Conclusions**

We demonstrated through extensive research that our product is unique as it incorporates different technologies all packed into one small device thru the raspberry pi. Our code is simple enough that it doesn’t need a lot of compiling power and the Microprocessor doesn’t have enough cores to run a heavy program. We created a product that incorporated different technologies as well as creating a simple solution that will give home owners control of their home from anywhere as long as system is connected thru WIFI. Also, visual connection of their entrance from anywhere and ability to lock and unlock door. Lastly, we also wanted to this project to be low cost, smart and simple enough for anyone to use. We were able to build this whole smart solution for under $100. One thing we learned was that streaming over the web wasn’t as easy as we thought and that knowing about Ports was very integral to being able to successfully build smartGuard.

**4.1 Final Product**

****

# **Acknowledgements**

The Authors wish to thank the anonymous reviewers for their valuable suggestions

**References**

[1] <https://www.raspberrypi.org/>

[2] <https://voice.google.com/u/0/about>

[3] <https://www.twilio.com/>

[4] <https://github.com/google/aiyprojects-raspbian/releases>