Designing and Implementing an Affordable and Accessible Smart Home Based on Internet of Things

Urvi Joshi[1], Aaron Dills[1], Eric Biazo[1], Cameron Cook[1], Zesheng Chen[1], and Guoping Wang[2]

[1]Department of Computer Science, Indiana University - Purdue University Fort Wayne

[2]Department of Electrical and Computer Engineering, Indiana University - Purdue University Fort Wayne

Abstract

The purpose of this research is to build an affordable and accessible smart home system based on the technologies of Internet of things. Current smart home systems on the market are expensive and require compatible devices. Our goal is to investigate, design, and implement a cheaper system that can be used with appliances that people already have. Building an affordable and accessible smart home is important, because it allows people to control and monitor aspects of their homes, which makes lives easier and keeps homes more secure. In our system, a Raspberry Pi device is used to control various peripheral devices including lights, blinds, a fan, a temperature sensor, and a security camera, by executing various Python scripts. A user interacts with the system through a web interface. That is, a light or a fan can be turned on or off remotely, whereas the live video stream is sent to the user's smart phone, tablet, or computer, through a web browser. The final product is a system that, with some modifications, can be used with most lights, fans, and blinds already in many homes. This paper details the motivations, challenges, design, process, and conclusions of our research.

1. Introduction

Internet of Things or IoT is defined as "a network of items – each embedded with sensors – which are connected to the Internet" [1]. Smart homes are "those where household devices/home appliances could monitor and control remotely. When these household devices in smart homes connect with the internet using proper network architecture and standard protocols, the whole system can be called as Smart Home in IoT environment or IoT based Smart Homes" [2]. A smart home based on IoT is important, because it allows a user to remotely control aspects of his or her home from anywhere and at anytime, which makes life much easier. For example, you do not have to be at home to close your blinds or turn off your lights. Moreover, monitoring a live camera feed of your home helps keep your home more secure.

Popular smart home systems on the market today are Amazon Echo and Google Home. Amazon Echo costs $99.99, whereas Google Home costs $129.99. Both systems are voice-controlled and can make calls, play music, and control smart home devices [3, 4]. However, both these systems only work with compatible devices that are sold separately. For instance, in Google Home, the light that can be controlled by the system currently costs $69.94, and the Nest Thermostat that is used to control temperature costs $203.99. Moreover, Ring Video Doorbell, used as a security camera at a door, costs $168.99 [5]. It is obvious that with each device added, the price of the entire smart home system can exceed a few hundred dollars. For users who cannot afford the cost and just want simple control of devices they already own, Google Home and Amazon Echo may not be practical solutions.

Our system cuts the cost of buying compatible devices by interfacing with appliances that people already own. Our system can turn on and off lights and a fan, open and close blinds, and take pictures and videos from a security camera and email them to the user. The core module in our system (i.e., a Raspberry Pi [6]) costs only about $35 [7]. Currently, the system that we have built is not voice controlled and cannot make calls. However, it does contain the essential functionalities and serves as a starting point for future smart home systems.

From a high level, our project is designed as follows. A main microcontroller (i.e., Raspberry Pi) is set up as a website and provides web services to users. Meanwhile, this microcontroller connects to and interfaces with devices (i.e., sensors and actuators), such as lights, blinds, a fan, a temperature sensor, and a camera. A user can send a command to the microcontroller through a web browser in her or his smart phone, tablet, or computer, in order to turn on or off the lights, open or close the blinds, turn on or off the fan, or take a picture or video. Moreover, a live video stream is sent to the authorized users' web browser in real-time. The final product is a system that, with some modifications, can be used with most lights, fans, and blinds already in many homes.

In this paper, Section 2 briefly describes each component of the IoT architecture for our designed smart home, whereas Section 3 points out the main challenges. Section 4 details the implementation of the smart home. Finally, Section 5 discusses lessons learned, and Section 5 concludes this paper.

2. Design of IoT Smart Home

Figure 1 gives a pictorial view of the design of our system. A MCU platform or microcontroller used is a Raspberry Pi 3 Model B, which hosts a website and interfaces with sensors and actuators. In our smart home, sensors include a temperature sensor and a security camera, whereas actuators are lights, blinds, and a fan. To obtain the high quality live camera feed, we use another dedicated Raspberry Pi to collect camera data. Similarly, to control the fan, an Arduino [8] is applied to interface with it. The user application is a web browser in the user's smart phone, tablet, or computer. A user can view the data from sensors and control the actuators through the web browser. For example, when a user clicks a button (e.g., "Turn on Light") in the web page, the request is sent to the MCU (i.e., Raspberry Pi), and the MCU interfaces with an actuator (e.g., light) to carry out the user's request. The implementation and specific parts used are further detailed in Section 4.

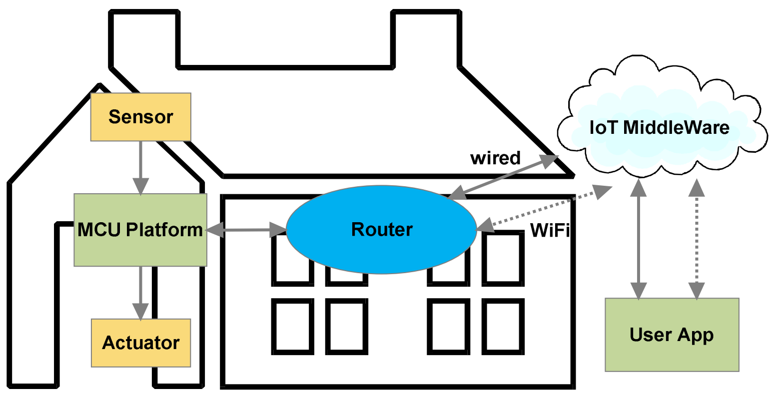

Figure 1. Design of Smart Home

3. Challenges

As we were building the system and adding more functionality, we found the three main challenges:

1) Availability of online resources. IoT is a relatively new research area. Online materials on connecting the MCU with different actuators are limited, especially for blinds and fans. For example, it is difficult to find a motor that is strong enough to pull up the blinds but not so strong to break the Raspberry Pi. Moreover, not many documents can be found to apply the infrared bulb to control the fan. As a result, we took much trial and error in our research.
2) Limitations of the Raspberry Pi. Originally, we wanted to use a single Raspberry Pi to collect all the data from the sensors and control all the actuators, in order to reduce the total cost. We found that although the Raspberry Pi was able to perform the jobs, the response time was beyond users' tolerance. As a result, we applied a separate Raspberry Pi to collect data from camera, which made the website become much more responsive.
3) Security. The IoT framework is well known for the vulnerability to security attacks [9]. In our designed smart home system, we have to consider how our system can be less vulnerable to different security issues, such as password protection.

4. Implementation of IoT Smart Home

In this section, we detail the implementation of each component of our designed system, including a website, prototype, lights, blinds, a fan, a camera, and a temperature sensor.

4.1 Website

The MCU (i.e., Raspberry Pi) hosts a website, which is implemented using HTML, CSS/Bootstrap, JavaScript, PHP, and MySQL [10, 11]. The website uses three tables in the MySQL database. One table is for storing user information, one for current settings, and another for temperature readings.

When accessing the website for the first time, a user is taken to a registration page where they enter their name, email, and password, as shown in Figure 2. This information is stored into the database table for user information. The password in the database is encrypted for enhancing the security. For every subsequent visit to the website, the user is taken to the login page where they enter their username and password, as shown in Figure 3. The username and password are checked against the database.

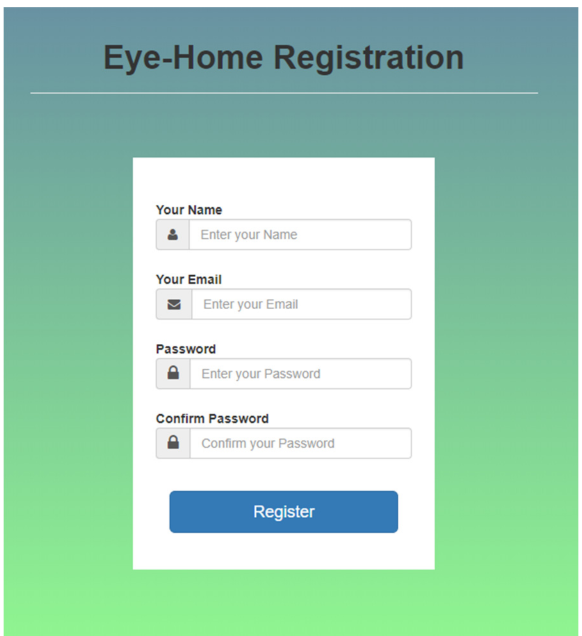

Figure 2. Register Page

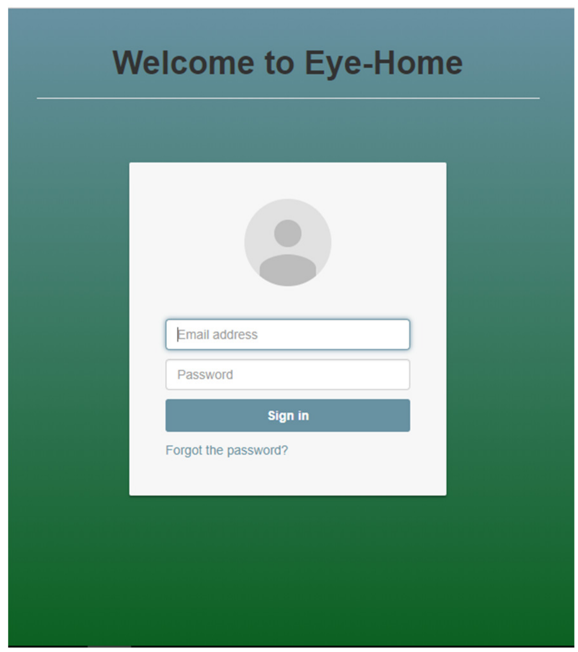

Figure 3. Login Page

If all the information is correct, the user is taken to their dashboard, which is shown in Figure 4. Through this dashboard, a user can turn on and off lights and a fan, open and close blinds, see a live stream of the camera, read the value of current temperature, and take pictures and videos. Specifically, the current temperature reading is updated every five seconds with the value from a temperature sensor connected to the Raspberry Pi. All temperature values are also stored in the database for future reference. The current status of the light is shown in the dashboard and is stored in the setting table in database. The live video stream can be viewed and recorded. Moreover, the buttons are designed as an input form that, when clicked, send requests to the PHP web server and trigger the server to run Python scripts to respond to the user's requests. The logout link logs out the user and redirects to the login page.

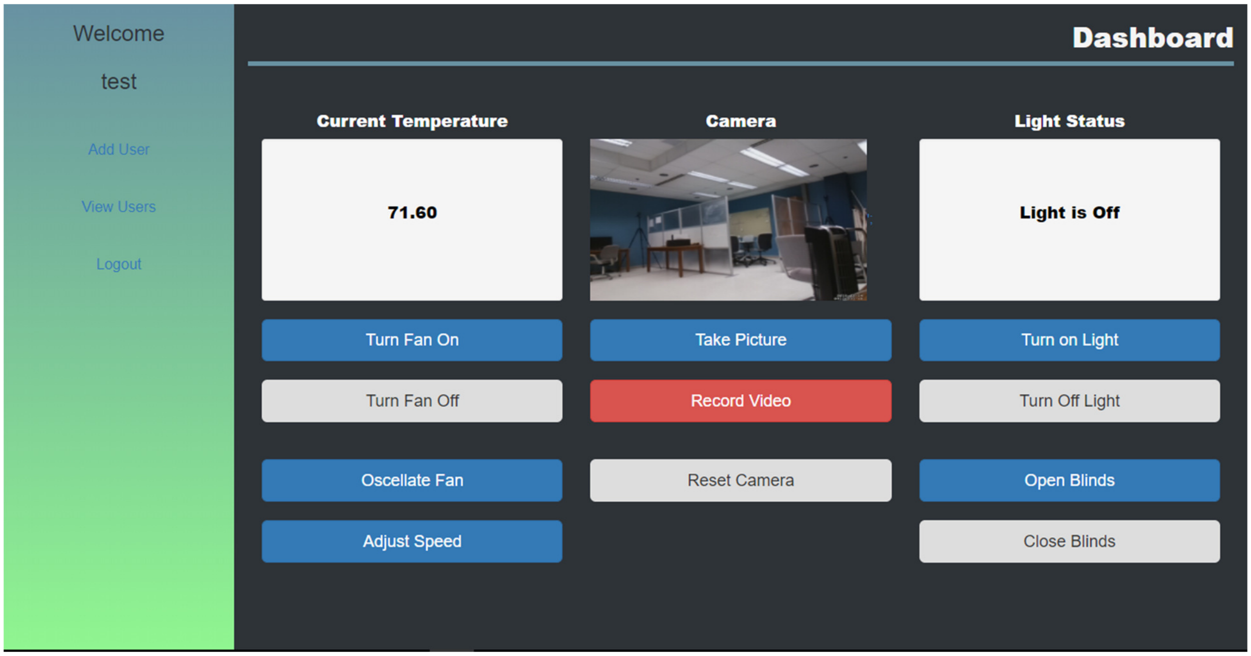

Figure 4. Dashboard

Further, an admin user has the option to add a new user through the dashboard. Figure 5 shows the page for the admin to add a new user. Here we separate the admin user from other users so that the system is more secure. That is, "Add User" option in the dashboard is only visible to the admin user. Moreover, an admin user can also view the information of all the other users, as shown in Figure 6.

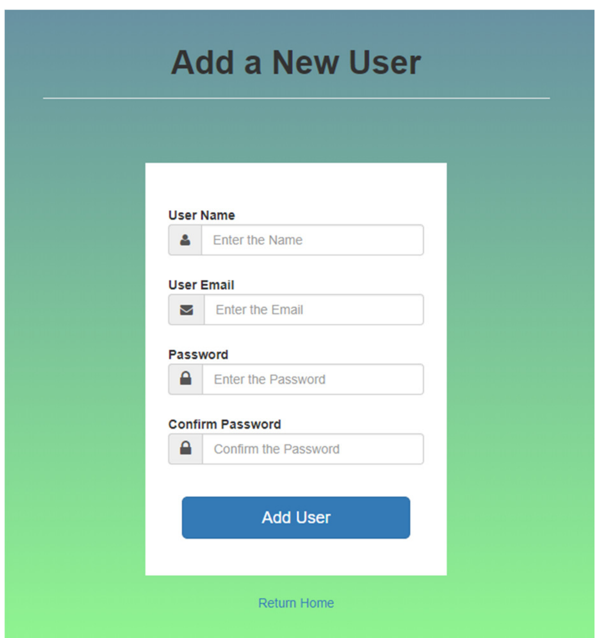

Figure 5. Add New User Page

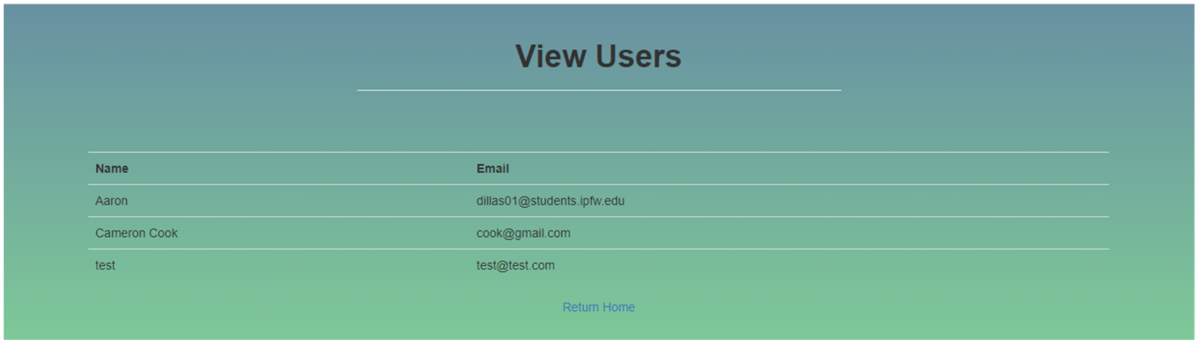

Figure 6. View Users Page

4.2 Prototype

Figure 7 shows the overall setup of our IoT smart home prototype. Specifically, a Raspberry Pi 3 Model B connects to lights, blinds, a fan through an Arduino Uno [8], a camera through anther Raspberry Pi 3 Model B, and a temperature sensor. Figure 8 demonstrates the pin connections on the Raspberry Pi. The lights apply an 8 channel 5V relay [12] and Supernight 5M/16.4 Ft SMD 3528 RGB 300 LED Color Changing Kit lights [13]. The blinds use a TowerPro SG-5010 servo motor [14]. The fan is controlled through an Optek OPV332 infrared light bulb [15] connected to the Arduino Uno. The camera is a picamera [16]. The temperature sensor is a DHT22 temperature and humidity sensor [17]. The total cost of the entire prototype is about $238.50.

Figure 7. Overall Setup of IoT Smart Home

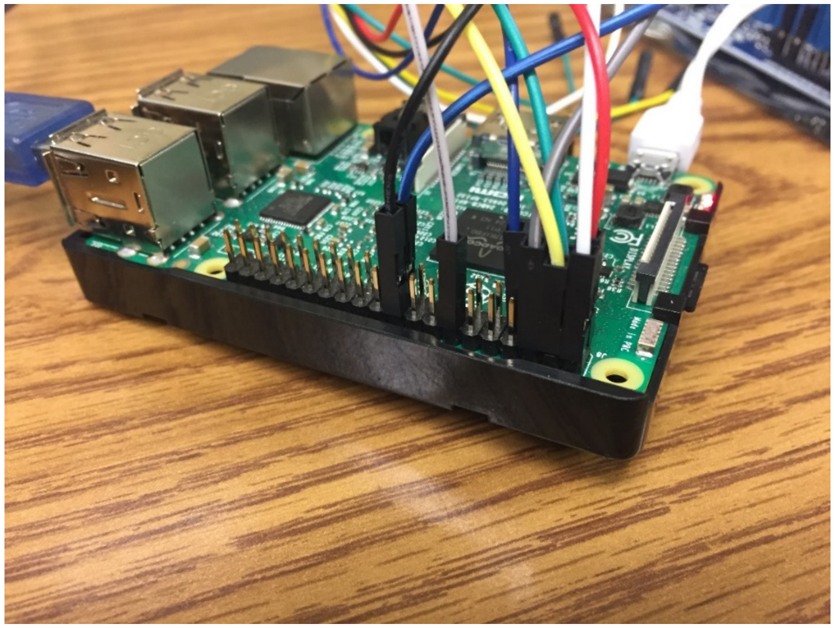

Figure 8. Pin Connections for Entire Smart Home

4.3 Lights

The light control system is implemented using the string lights and the eight channel relay [12], as shown in Figure 9. The Raspberry Pi is connected to the light through the relay, as shown in Figure 10. Here we do not connect directly the Raspberry Pi to the lights, because the Raspberry Pi cannot provide enough power to lights. We create the hardware connection by referring to the online tutorial [18]. Specifically, the VCC, IN2, and GND pins on the relay are connected to pins 2, 19, and 6 on the Raspberry Pi, respectively. The lights are turned on or off by running one of two Python scripts in the Raspberry Pi. When the button "Turn On Light" in the user's web browser is clicked, the script to turn on the lights is executed. Similarly, when the button "Turn Off Light" is clicked, the other script is executed.

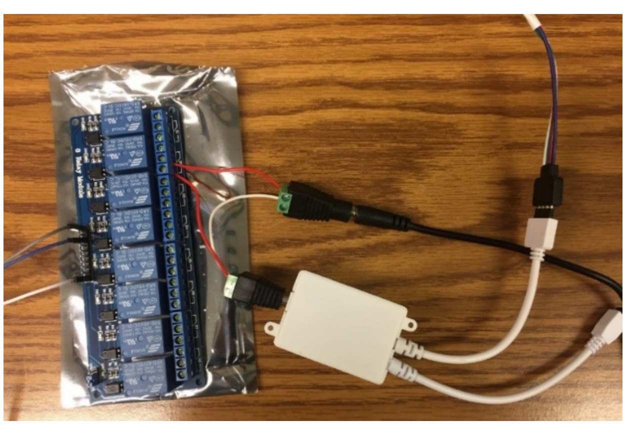

Figure 9. Relay Connections From Relay to Lights

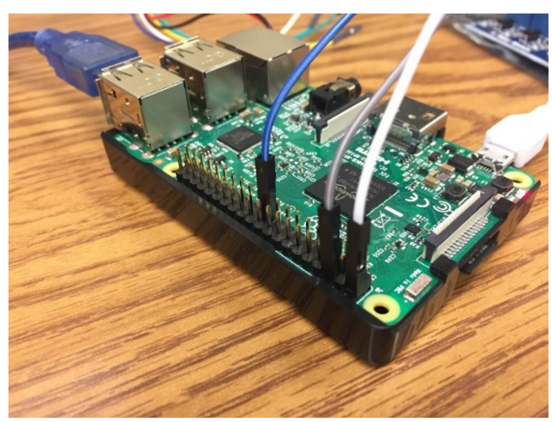

Figure 10. Connections From Relay to Raspberry Pi

4.4 Blinds

The blinds control system is implemented using the servo motor connected to the Raspberry Pi, as shown in Figure 11. The servo motor [14] has been modified to spin forwards or backwards continuously, referring to the online tutorials [19, 20, 21]. Specifically, the power (red wire), control (yellow wire), and ground (brown wire) pins on the servo motor are connected to pins 4, 3, and 14 on the Raspberry Pi, respectively. In our prototype, we use a small bottle with two strings tied to it to attach to the servo motor. The strings are to be connected to the chords of the blinds. Moreover, the servo motor is mounted to a wooden frame, which is built from one 3’x3”x0.5” wooden plank. The frame has two 12”x3” pieces on the outside, two 7”x3” pieces on the interior, and four 1.5”x3” pieces holding the motor and the bottle. The wooden frame is necessary to support the servo motor and provides enough stability to allow the motor to pull up the blinds. Similar to the control of lights, the servo motor is controlled by running two Python scripts in Raspberry Pi. One script spins the motor forward to pull up the blinds, whereas the other script is for spinning the motor backwards and pulling down the blinds. The script for pulling up (or down) the blinds is executed when a user clicks the “Open Blinds” (or “Close Blinds”) button in the dashboard page.

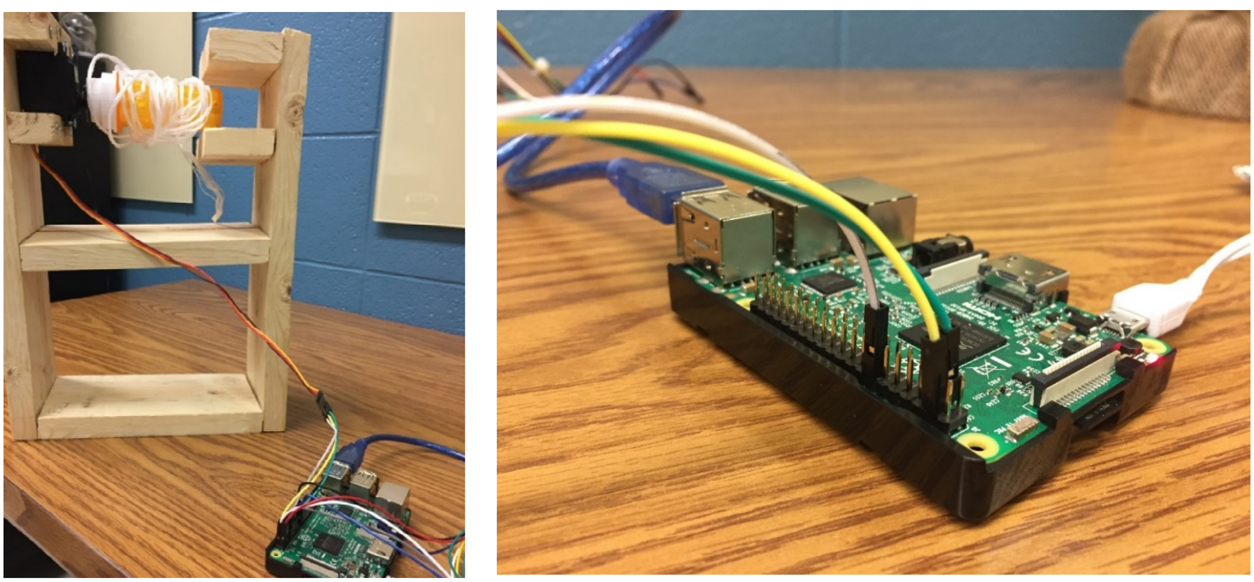

Figure 11. Blinds Frame and Servo Motor Setup and Connection with Raspberry Pi

4.5 Fan

The raspberry pi uses an infrared light bulb [15] that emits light at different frequencies to communicate with the fan, as shown in Figure 12. Specifically, the bulb is connected to the GND and 13 pins on an Arduino Uno. Moreover, the Arduino communicates with the Raspberry Pi through a serial port. Our innovation for controlling the fan is to use the frequency of the bulb lighting to determine the operation that the fan should execute. Specifically, the fan used has four different operations: On, Off, Speed, and Rotate. The Raspberry Pi contains four Python scripts, one for each of the fan operations. When a user clicks a button in the dashboard page for controlling the fan, the request is sent to the web server in Raspberry Pi that runs a Python script to control the frequency of the infrared bulb, which in return instructs the operation of the fan.

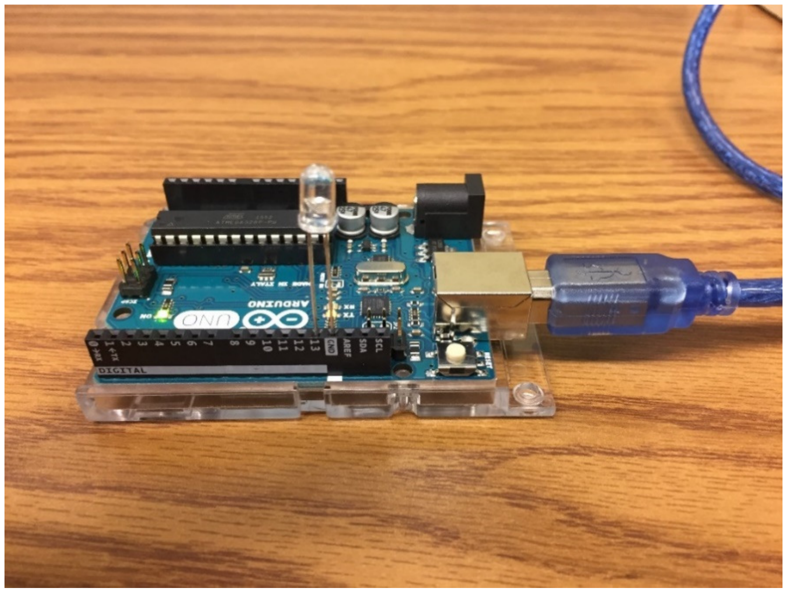

Figure 12. Fan Connection

4.6 Camera

The camera is connected to a separate Raspberry Pi. As shown in Figure 13, a picamera [16] is used for live video streams and pictures. We use the built-in library “Picamera” in Python [22] to obtain live video streams and pictures from the picamera. The video sent from the pi is in the H264 format.

The main Raspberry Pi establishes a connection with the remote Raspberry Pi attached with the picamera, using a Python library “Paramiko” [23, 24, 25]. Specifically, such a library provides a way to create an SSH client in the main Raspberry Pi. That is, the main Raspberry Pi uses the Paramiko library to establish the SSH connection with the remote Raspberry Pi and then sends a command to the remote machine to run a Python script that takes a picture or video, or emails the picture or video taken.

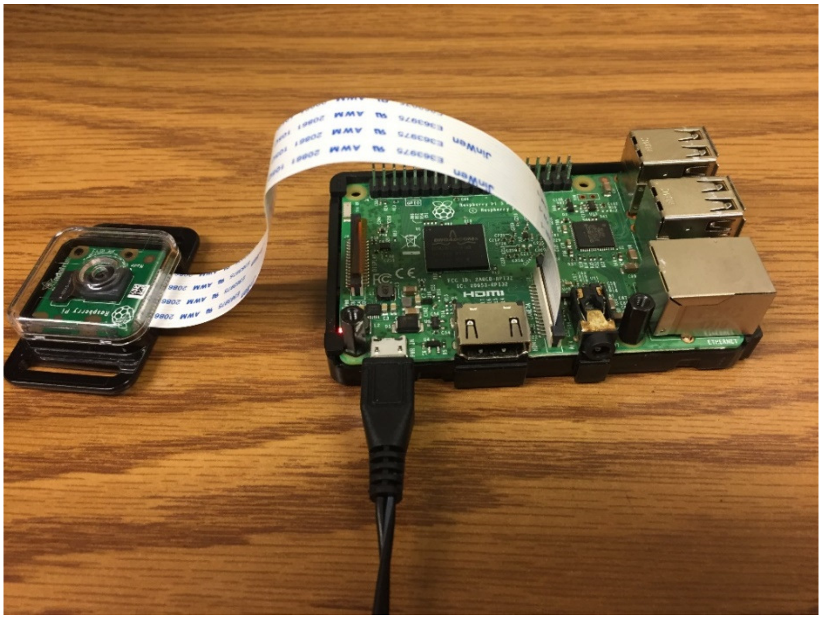

Figure 13. Camera

4.7 Temperature Sensor

The temperature sensor is a DHT22 temperature sensor [17]. As shown in Figure 14, the sensor has been soldered to a board with a distance sensor as well. The temperature sensor only needs 3.3V of power. As a result, the power (red wire), ground (black wire), input (blue wire) pins from the temperature sensor are connected to pins 1, 20, and 7 in the Raspberry Pi, respectively.

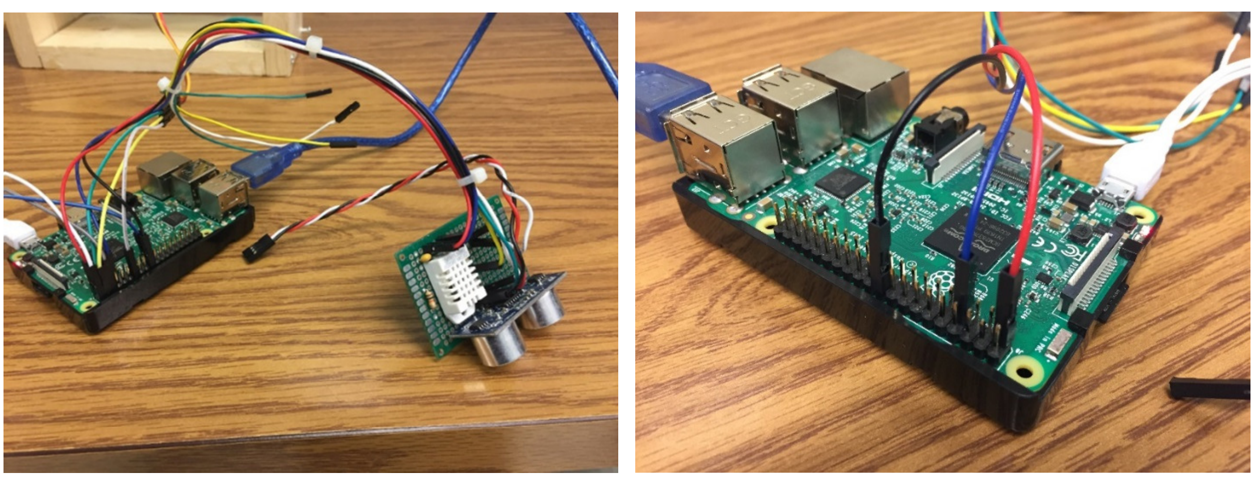

Figure 14. Connection From a Temperature Sensor to Raspberry Pi

5. Lessons Learned

The entire project was a learning opportunity. As a group, we learned about IoT technologies and how they can be used in a smart home system. We became to appreciate the versatility and understand the limitations of a Raspberry Pi. We acquired knowledge of how to connect a Raspberry Pi to lights and blinds to control them, how to use infrared to operate a fan, and how to communicate information between two Raspberry Pis. In addition, we gained many web design skills like HTML, CSS, JavaScript, PHP, and MySQL for IoT applications.

6. Conclusions

In this project, our main contribution is a prototype of an affordable and accessible smart home system. Our prototype, with a few modifications, can control lights, blinds, a fan, and a camera in any home. The prototype can be used as a basis for more advanced smart home systems.

For the future works, we plan to enhance our smart home system by considering more devices in the home, such as an oven, a stove, a garage door, and an air conditioner. We will also consolidate our designed system by introducing the methods from machine learning [26]. That is, the smart home system is able to automatically turn on or off connected devices based on the information in the home environment. A simple example is that a fan is automatically turned on when the temperature is above a certain value.

ACKNOWLEDGMENTS

This work was supported in part by the 2017 IPFW IRSC Collaborative Research Grants.

References:

[1] R. Minerva, A. Biru, and D. Rotondi, "Towards a Definition of the Internet of Things (IoT)," *IEEE Internet Initiative*, May 2015. [Online] Available: http://iot.ieee.org/definition.html (March/2018 accessed).

[2] P. Gaikwad, J. Gabhane, and S. Golait, "A survey based on Smart Homes system using Internet-of-Things," 2015 International Conference on Computation of Power, Energy Information and Communication (ICCPEIC'15), April 2015, Chennai, India.

[3] Echo (2nd Generation) [Online]. Available: https://www.amazon.com/dp/B07456BG8N?tag=googhydr-20&hvadid=245069637046&hvpos=1t1&hvnetw=g&hvrand=2523177145212576366&hvpone=&hvptwo=&hvqmt=e&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9016356&hvtargid=kwd-295921611050&ref=pd_sl_2g7cb1h5ze_e (March/2018 accessed).

[4] Google Home [Online]. Available: https://store.google.com/us/product/meet_google_home (March/2018 accessed).

[5] The Best Smart Home Devices That Work with Google Home [Online]. Available: https://www.tomsguide.com/us/pictures-story/1075-google-home-compatible-devices.html#s16 (March/2018 accessed).

[6] Raspberry Pi [Online]. Available: https://www.raspberrypi.org/ (March/2018 accessed).

[7] Raspberry Pi 3 on Sale Now at $35 [Online]. Available: https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/ (March/2018 accessed).

[8] Arduino [Online]. Available: https://www.arduino.cc/ (March/2018 accessed).

[9] IoT Security Foundation [Online]. Available: https://www.iotsecurityfoundation.org/ (March/2018 accessed).

[10] Front-End Web Development, Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Front-end_web_development (March/2018 accessed).

[11] XAMPP [Online]. Available: https://www.apachefriends.org/index.html (March/2018 accessed).

[12] Elegoo 8 Channel DC 5V Relay Module [Online]. Available: https://www.amazon.com/gp/product/B01HCFJC0Y/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1 (March/2018 accessed).

[13] SUPERNIGHT (TM) IP65 Waterproof 5M/16.4 Ft RGB SMD 3528 LED Color Changing 300 LED Flexible Strip Light [Online]. Available: https://www.amazon.com/dp/B00BP0TF5A/ref=asc_df_B00BP0TF5A5388316/?tag=hyprod-20&creative=395033&creativeASIN=B00BP0TF5A&linkCode=df0&hvadid=232740426404&hvpos=1o1&hvnetw=g&hvrand=14758942366558414328&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9016345&hvtargid=pla-391735213894 (March/2018 accessed).

[14] TowerPro SG-5010 Servo Motor [Online]. Available: https://www.adafruit.com/product/155?gclid=Cj0KCQiA5t7UBRDaARIsAOreQthm39hewT8Fde3F_ah5kJdlxqLGl_n7yg-kdV2rjQs3bbJqPCog5JIaAhFJEALw_wcB (March/2018 accessed).

[15] Optek OPV332 Infrared Light Bulb [Online]. Available: https://www.mouser.com/productdetail/optek-tt-electronics/opv332?qs=sGAEpiMZZMsn8wIhgY8aVaWSORibvsBn7Aopk9g6rX4%3D (March/2018 accessed).

[16] Raspberry Pi Camera Module V2 [Online]. Available: https://www.amazon.com/Raspberry-Pi-Camera-Module-Megapixel/dp/B01ER2SKFS (March/2018 accessed).

[17] DHT Temperature-Humidity Sensor [Online]. Available: https://www.adafruit.com/product/385 (March/2018 accessed).

[18] Raspberry Pi: How to Turn on a Light With Your Phone [Online]. Available: https://www.youtube.com/watch?v=WpM1aq4B8-A (March/2018 accessed).

[19] How to Make Servo 360° Degree Continuous Servo Rotation Mod Method 2 [Online]. Available: https://www.youtube.com/watch?v=K3m6uxwxQnw (March/2018 accessed).

[20] Servo Motor Control With Raspberry Pi [Online]. Available: http://www.instructables.com/id/Servo-Motor-Control-With-Raspberry-Pi/ (March/2018 accessed)

[21] Claimed to Be the Cheapest IoT DIY [Online]. Available: https://electronicsofthings.com/iot-projects/make-build-iot/claimed-cheapest-iot-diy/ (March/2018 accessed).

[22] Getting Started with Picamera [Online]. Available: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera (March/2018 accessed).

[23] Paramiko [Online]. Available: http://www.paramiko.org/ (March/2018 accessed).

[24] Python and SSH: Sending Commands Over SSH Using Paramiko [Online]. Available: https://daanlenaerts.com/blog/2016/01/02/python-and-ssh-sending-commands-over-ssh-using-paramiko/ (March/2018 accessed).

[25] Use Python to Build a Raspberry Pi - Powered Home Security Camera [Online]. Available: https://medium.com/@andyorr/use-python-to-build-a-raspberry-pi-powered-home-security-camera-for-50-84ab7e344e2d (March/2018 accessed).

[26] Machine Learning, Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning (March/2018 accessed).