

**Senior Design Project**

**Design of Low Cost Programmable IOT Platform for Home Automation**

ENGE476 Senior Design Project I

Department of Engineering and Aviation Sciences

University of Maryland, Eastern Shore

Samuel Adedeji

Project Advisor, Dr. Lei Zhang

Submitted

Mon. Day, Year

Special notice for formatting:

1. Don’t add any blank rows. Use spacing or page breaks.
2. Every figure or picture inserted in the text, must be accompanied with the caption and detailed description.
3. Keep the formatting consistent (learn and apply the MS ‘Styles’ in formatting).
4. The final report didn’t meet the basic formatting requirements won’t be accepted.

List of Contents

[List of Contents 3](#_Toc472281998)

[List of Figures 2](#_Toc472281999)

[List of Tables 2](#_Toc472282000)

[Abstract 4](#_Toc472282001)

[1. Introduction 5](#_Toc472282002)

[1.1 Backgound/Motivation 5](#_Toc472282003)

[1.2 Objective 6](#_Toc472282004)

[1.3 Design Requirements 7](#_Toc472282005)

[1.4 Design Constraints 7](#_Toc472282006)

[1.5 Design Methods 7](#_Toc472282007)

[2. Project Description 9](#_Toc472282008)

[2.1 System Description 9](#_Toc472282009)

[2.2 System Diagram 9](#_Toc472282010)

[2.3 System Functions 9](#_Toc472282011)

[3. Implementation Plan 11](#_Toc472282012)

[3.1 Tasks 11](#_Toc472282013)

[3.2 Team Organization 12](#_Toc472282014)

[3.2.1 Responsibility of Samuel Adedeji. 12](#_Toc472282015)

[3.3 Timeline/Milestones/Delivery Plan 12](#_Toc472282016)

[4. Implementation 14](#_Toc472282017)

[4.1 Implementation of Task 1 Communicate with weather underground application program interface (API) to access and extract weather information. 14](#_Toc472282018)

[4.2 Implementation of Task 2. 15](#_Toc472282019)

[5. Conclusion. 17](#_Toc472282020)

[Acknowledgement 18](#_Toc472282021)

[Appendix 19](#_Toc472282022)

[A. Component Specs 19](#_Toc472282023)

[1. Rapsberry Pi Specifications 19](#_Toc472282024)

[2. Specs of Arduino 19](#_Toc472282025)

[B. Source Code. 19](#_Toc472282026)

[1. Source Code for Weather Information Retrival 19](#_Toc472282027)

[2. Source Code of Robotic Arm 22](#_Toc472282028)

[REFERENCES 23](#_Toc472282029)

List of Figures

[Figure 1. Sensitivity to Number of Sensors 6](#_Toc472282056)

[Figure 2. Sensitivity to Climate Zones 6](#_Toc472282057)

[Figure 3 General System Diagram 9](#_Toc472282058)

[Figure 4 User login to the weather underground webpage. 14](#_Toc472282059)

[Figure 5 The terminal to the python integrated development program is used to output the retrieved weather information along with the status of the Led lights. 15](#_Toc472282060)

[Figure 6. The led lights turns on after the code has run. 16](#_Toc472282061)

List of Tables

[Table 1. Project Timeline and Delivery Plan 9](#_Toc471118144)

Abstract

By the end of the project, summarize the project into short text and put here.

1. Introduction

The purpose of this project is to develop a low-cost, programmable Internet of Things (**IoT**) platform using the raspberry pi microcomputer for home automation applications.

## Backgound/Motivation

The development of this low-cost, programmable raspberry pi internet of things (IoT) platform for home applications was inspired by the fact that there are many appliances in everyday user households that can be operated manually but users desire more autonomy when it comes to controlling their home appliances.

Being able to automate these appliances allows users to streamline daily activities as well as better manage living resources in their households. Some way homeowners are able to visualize their savings is through efficient water heating and space heating systems. These systems use intelligent heating control by automatically managing room temperature based on time, outside temperature, and presence. Developing this product for home automation applications requires one to have a good understanding of a what specifically makes up the IoT, what home automation is, and what are some of the current products available to perform the task of controlling devices in the house.

The IoT technological platform is currently in its infancy stage and growing quickly due to the many positive implications it has on improving human lives.

IoT is an umbrella term used to describe the interconnectivity of smart objects on a network that are able to communicate with each other, harvest information from the environment and interact with the physical world. The intricacy of common smart home networks will be discussed. As mentioned earlier a smart home network is a specific type of IoT network that functions in the structured environment of a residential homeowner. Just like any other IoT network it needs to contain the following components to function properly. These components are the sensors, actuators, people, services, and networks. Some common sensors used to collect information from the physical environment include a thermistor, microphone, humidity, temperature and motion sensors. Some actuators include LCD displays and speakers. In addition, these smart objects use existing internet-connected services to provide a means for information transfer, analytics, application and communication with the users and other smart objects on the network.

There are currently a vast and varied amount of home automation products available to homeowners in the consumer electronics market. Many of these products obey the principles of IoT platforms. Programmable thermostats are commonplace devices used for home automation purposes. The basic idea is to control the HVAC equipment based on a setback schedule: the house is conditioned to a set point temperature when the occupants are typically active and floats to a more energy-efficient setback temperature when the occupants are typically away or asleep.

The charts below explain how energy is saved with the addition of the motion sensor in a home automation system that implements a smart thermostat.

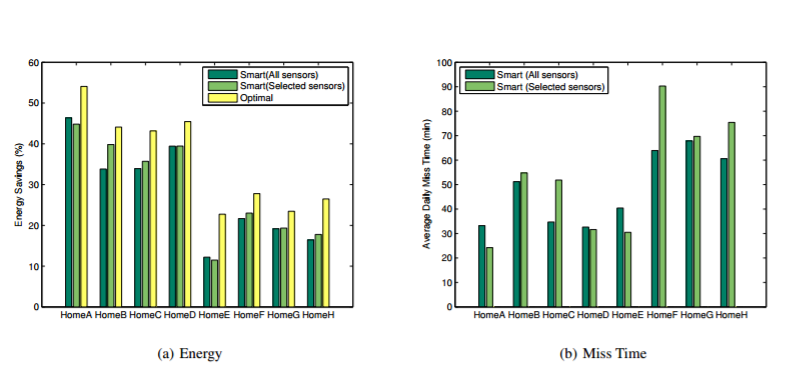


Figure 1. Sensitivity to Number of Sensors

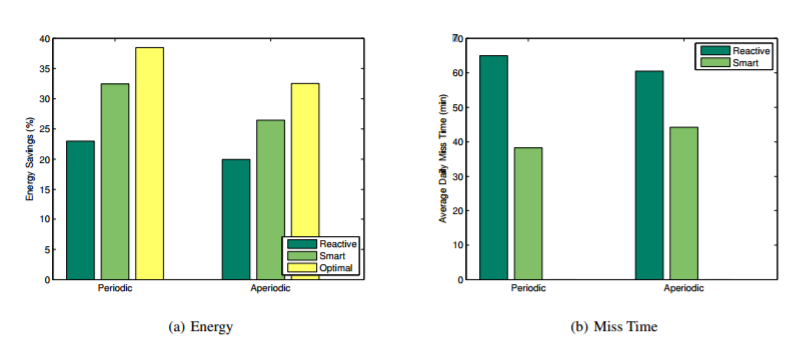


Figure 2. Sensitivity to Climate Zones

## Objective

From the perspective of the user, the goal of this project is to design and develop a low-cost personal home automation system using the raspberry pi microcomputer and an internet of things platform to manage resource consumption and optimize living conditions in the house. The home automation system will be equipped with multiple user interfaces that allow the user to interact with the raspberry pi, the main controller in the IOT system, in order to control home appliances. The user will be given the option to access the system locally or remotely through a mobile device that can connect to the internet. Using the mobile device, the user can make a request to the system to obtain information in order to make a decision to control an appliance in the household. For example, the user can control his or her sprinkler system in the house to control water usage based upon daily, weekly, or monthly weather information. In addition, the user also has the option of using an audio interface to communicate with the command center in the IoT system. Once again, the user can make a request to gather relevant information and use that information to control an appliance in the household. One aspect that makes this IoT system unique is that machine learning will be implemented to ensure that the system is able to understand and learn how one or more users of the system in the household use different household appliances. The IoT platform will be able to gather information on the user frequency and period when it comes to the different activities and interactions the user(s) has/have in the household on a daily weekly, or monthly basis. The algorithms used in the machine learning AI will be able to make adjustments automatedly or through positive and negative feedback from the user.

## Design Requirements

The design requirements for this project are stated as follows.

1. The system needs to be able to acquire and display user requested information from the internet.
2. The system needs to be able to control different home appliances based on acquired information from the internet
3. The system will be equipped with 2 human-machine interfaces that will be used to extract online information. User(s) will be able to request information by accessing the internet remotely or through a mobile device. Users will be able to request information by using speech recognition software to make online queries.
4. The system will be equipped with a graphical interface to display real-time information about the state of the system along with keeping a log of past system events.
5. The system will be equipped with machine learning algorithms that will be able to make automated home automation decisions by learning the behaviors of the user(s) using the system. User(s) will also have the option of overriding these machine learning algorithms as necessary.

## Design Constraints

The design constraints for this project are stated as followed:

* Access to wireless network required
* Data acquisition limited by services available online
* Online services may charge a fee to access information
* Computing capability of Raspberry pi limited
* The budget of this project should not exceed $150

## Design Methods

The design methods for this project are described as follows:

The first step for developing this project is to identify a problem that is affecting many people in their daily lives and research possible methods for implementing a solution. The second step is to use the systematic approach of the engineering design process to specify appropriate design requirements and constraints. The third step is to finalize an implementation plan and then gather project specific materials and resources such as the hardware and software that will be used to create the system. The fourth step is to develop a fully operational version of the system through the execution of the implementation plan by completing tasks and subtasks. The final step of developing this project is to modify and make any necessary adjustments to produce the final version of the product.

1. Project Description

## System Description

The system description for this project is defined as follows:

The brain and brawn of this IoT platform system for home automation applications is the raspberry pi microprocessor. The raspberry pi is the command center that is able to control most of the home appliances based on the information it receives from the multiple online services it accesses. Before the raspberry is able to output the proper or desired control voltage for the specific appliance in the household it needs to access additional information or usage information from the user or sensors interacting in the home environment. The raspberry pi is able to use three interfaces developed to access the internet and the information services available to allow the home automation activities to occur. The first interface created is mobile web interface. The second interface is an audio interface. The last interface is an LCD touchscreen which will be used as a graphical user interface to keep the user updated with the real-time information of the current state of the system as well as keep a log of past, present, and predicted future events. The brain part of the raspberry pi is that is able to “process” all/most of the information requests it receives from the user and use that pertinent information and to make a right decision of controlling a specific household appliance or communicating the proper, accurate system information back to the user through the user interface.

## System Diagram

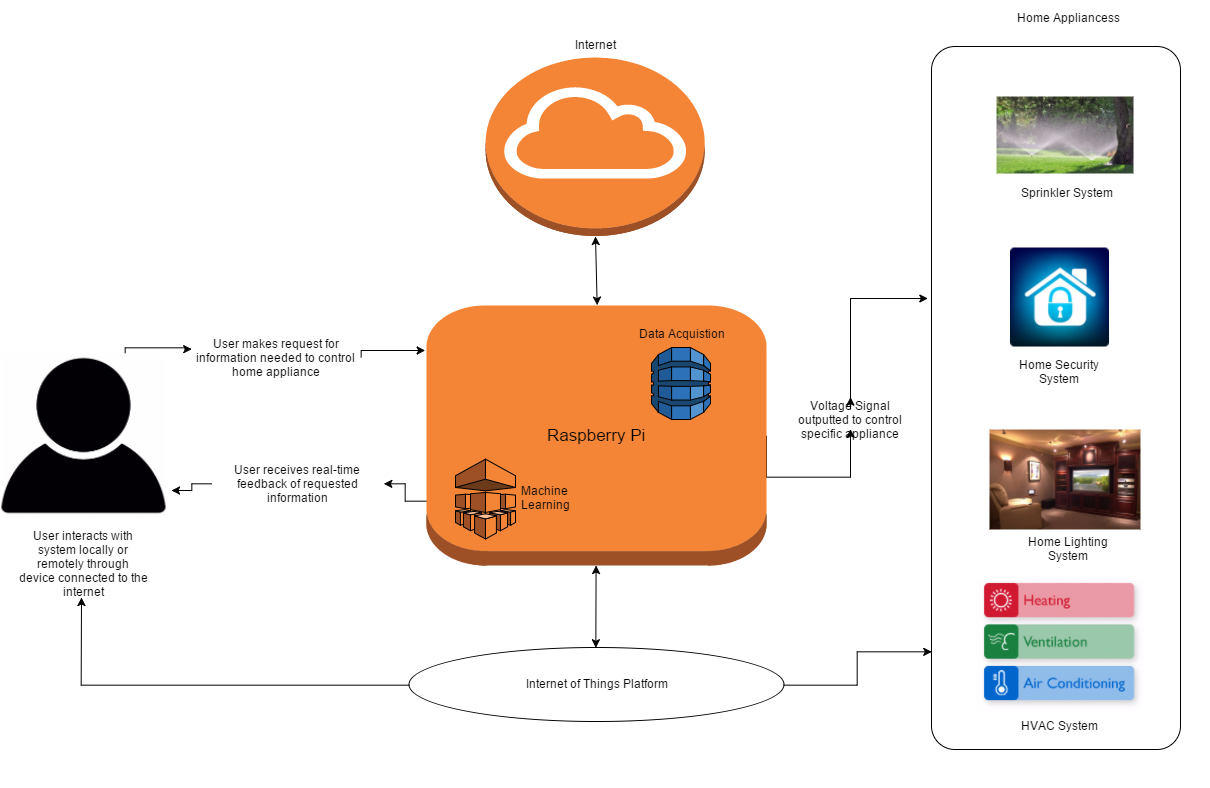


Figure 3 General System Diagram

## System Functions

Here you need to clearly define every function and every state of the system. Make you’re your state graph is complete. For example:

The system functions for this project are stated as follows:

1. Whenever the user (a household occupant) desires to control a household appliance he or she can make an input request to the IoT home automation system.
2. The user input request is processed using either the speech recognition, web/mobile, or LCD control panel interface available for the IoT home automation system.
3. Once an input request has been made by the user, software to software communication occurs between the user interface and the raspberry pi microcomputer to determine which action to perform in the IoT system.
4. After processing a user request, the raspberry pi microprocessor is programmed to perform one or all of these functions
   1. Data Acquisition
   2. Real-time System Status Monitoring and History Log
   3. Appliance Control
      1. Machine Learning Assisted
      2. User Input and Feedback
5. Based upon a user request the raspberry pi will be programmed to perform data acquisition. Data acquisition involves using an internet connected device to access a web service online such as weather underground website so as to retrieve specific information such as the weather forecast. Another example of the system processing a user input to acquire data involves being able to answer queries such as the following: who is the current president of the united states? Who manufactured this product? and another relevant, similar questions.
6. Based upon a user request the raspberry pi will be programmed to report real time system information to the user. Using the LCD display as the output display and the proper python script, the raspberry pi will be programmed to communicate the current and past readings of the sensors to the LCD display to allow the user to know current system conditions so he or she can make adjustments accordingly.
7. Based upon user request, the system will be able to control different home appliances. Control of appliances will be done through the usage of a GPIO to send an output voltage signal to either an electromagnetic relay or a voltage amplifier which will be used as direct signal to turn an appliance such as the Heating Ventilation and Air Conditioning unit on and set it to the desired temperature. (HVAC)

1. Implementation Plan

## Tasks

The tasks needed to be accomplished in the project are stated as follows:

* **Task 1. Communicate with weather underground application program interface (API) to access and extract weather information**
  + Subtask 1. Sign up for an account on api.wundeground.com in order to obtain an API key
  + Subtask 2. Write the appropriate python script that will communicate with the weather API and make a retrieval request
  + Subtask 3. Using the specialized python script, extract the desired information from the weather API
  + Subtask 4. Using the specialized python script, parse the weather information
  + Subtask 5. Using the specialized python script, display the weather information to the user.
* **Task 2. Demonstrate the ability to control an appliance based upon information retrieved from online**
  + Subtask 1. Setup the LED circuit that will be used to visualize the control of a household appliance
  + Subtask 2. Retrieve weather information
  + Subtask 3. Turn the LEDs on the circuit on and off.
* **Task 3. Setup 2.8’ TFT LCD touchscreen display to be used as a dashboard/control panel to update the user of the current state of the IoT platform**
  + Subtask 1. Interface the LCD touchscreen with the raspberry pi
  + Subtask 2. Create graphical interface to allow the user to make requests and control appliances
  + Subtask 3. Enable graphical interface to update the user about the current state of the system before and after a request is made.
* **Task 4. Configure a local client server for the remote configuration control of a led using node-red.**
  + Subtask 1. Determine the use of node red as the specific software programming tool interface for the led and future hardware.
  + Subtask 2. Download and run the proper command line prompts to enable node-RED as client server.
  + Subtask 3. Control the led light through the flow module workspace..
* **Task 5. Configure a speech recognition system user interface that will enable the user to process requests and communicate with the command center.**
  + Subtask 1. Setup and configure usb microphone to allow the recording of user speech input.
  + Subtask 2. Download Google speech to text (STT) API software engine.
  + Subtask 3. Develop multiple keywords that will used to process requests for online information.
  + Subtask 4. Download Espeak text to speech (TTS) API software engine

## Team Organization

All the task and subtasks specified in the last section are to be completed by Samuel Adedeji. The responsibilities of Samuel Adedeji are stated as follows.

### Responsibility of Samuel Adedeji.

Task 1,2,3,4

## Timeline/Milestones/Delivery Plan

Please prospect the timeline to deliver the results of each task/subtask.

Please schedule your project to no more than 22 weeks from now.

1. Project Timeline and Delivery Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Task | Comments | Responsible Personnel |
| Week 1 | Independent Research | Identified UAV Collision Avoidance as problem that the project development could provide a solution for | Samuel Adedeji |
| Week 2 | Independent Research | Researched and summarized articles on the topic of UAV Collision Avoidance | Samuel Adedeji |
| Week 3 | Independent Research | Re-identified Augmented Reality System as a problem that the project development could provide a solution for. | Samuel Adedeji |
| Week 4 | Independent Research | Researched and summarized articles on the topic of Augmented Reality System | Samuel Adedeji |
| Week 5 | Independent Research | Received raspberry pi, camera module and additional hardware that will be used to implement the project tasks | Samuel Adedeji |
| Week 6 | Independent Research | Initialized the Linux Raspbian operating system OS on the Raspberry Pi 2 to begin implementing tasks | Samuel Adedeji |
| Week 7 | Independent Research | Re-identified IOT Home Automation as problem that the project development will provide a solution for | Samuel Adedeji |
| Week 8 | Independent Research | Researched and summarized articles on the topic of IOT Home Automation | Samuel Adedeji |
| Week 9 | Independent Research | Searched online for available websites that provide an API to access weather information | Samuel Adedeji |
| Week 10 | Task 1 Subtask 1,2 | Write python script. Write a module that makes an HTTP request to weather API | Samuel Adedeji |
| Week 11 | Task 1 Subtask 3,4 | Continuation of writing the python script, include a module that retrieves JSON data from weather API | Samuel Adedeji |
| Week 12 | Task 1 Subtask 5 | Finalize python script, Parse retrieved JSON data for desired weather information | Samuel Adedeji |
| Week 13 | Task 2 Subtask 1 | Obtain leds, resistors, and circuit board | Samuel Adedeji |
| Week 14 | Task 2 Subtask 1 | Setup the electrical components on circuit board | Samuel Adedeji |
| Week 15 | Task 2 Subtask 2 | Use the raspberry pi to run the python script created | Samuel Adedeji |
| Week 14 | Task 2 | Output the proper voltage signal from the GPIO pins to turn the LEDs on and off | Samuel Adedeji |
| Week 15 | Task 1 & 2 Complete | System finalization and delivery. Finish all documentations and ready for presentation. | Samuel Adedeji |

1. Implementation

## Implementation of Task 1.

***4.1.1 Subtask 1. Identify and sign-up for a weather website with accessibility to a weather information through an application program interface API.***

The internet of things requires being able to collect and acquire information online and transfer this information from and to different embedded objects on the network in order to control different appliances. In order to control the internal temperature of the house or a water system, one needs to collect weather information. The weather API that was chosen to acquire this weather information was the weather underground API available at the following link <https://www.wunderground.com/weather/api/>. The figure below shows what the website log-in looks like.

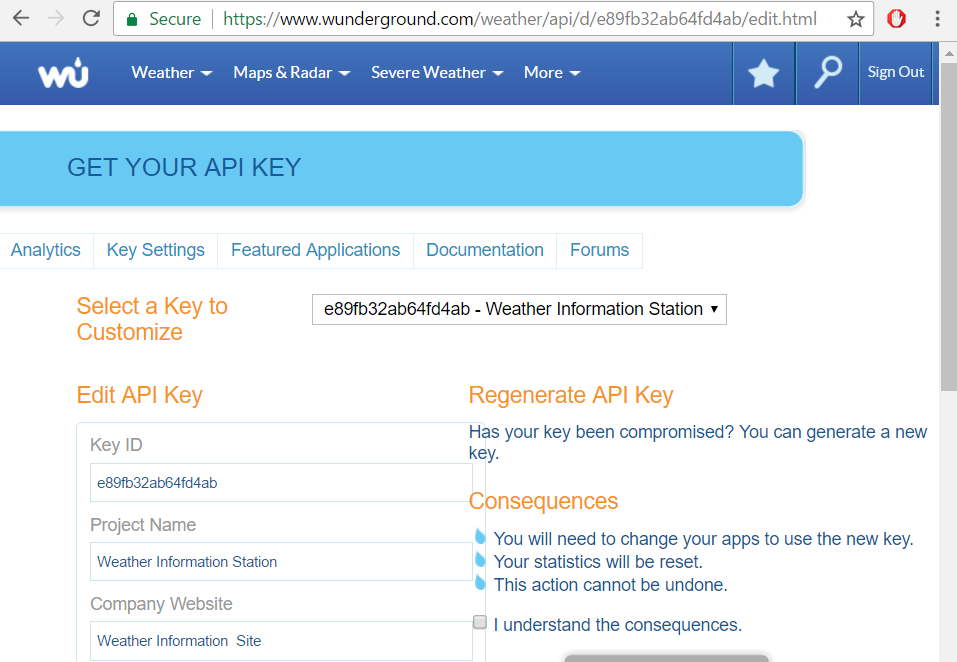


Figure 4 User login to the weather underground webpage.

This API is a free web service that allows the user to sign up and create a unique key ID that will allow them to retrieve the specific weather forecast of any location inside the United States as long as the user makes a valid API request. In order to make a valid API request

**4.1.2 Python program development to facilitate proper information exchange through internet protocols**

There is a multiple step approach that must be followed in order to complete a successful API call to a web page server. The first requirement is establishing the proper programming language that is compatible with the raspberry pi platform and the internet protocol used to communicate with the webpage. Python is the chosen language used to fulfill this task. While developing the program, the python script the url and json library have to be imported in order to communicate with the website and initiate the API retrieval request. Once the information has been retrieved in the JSON data form, the remainder of the program code can be used to parse for the specific weather data such as wind speed, humidity weather forecast, etc. Afterwards, variable declarations allow the programmer to specify which data he or she wants to display back to the user through a user interface.

## Implementation of Task 2.

***4.2.1 Subtask 1-3. Demonstrate the ability to control an appliance based upon information retrieved from online.***

A circuit composed of a power supply, leds, and resistors are used to visualize the result of being able to extract weather information and use that weather information to display how a home appliance can be controlled. The circuiThe process of extracting information requires using a software development IDE along with using a raspberry pi with internet access. The user of the system will able to control when the weather extracting program will be able to run through the use of one of the multiple interface integrated into the home automation system. After retrieving the weather information from online the user will also be able to use the program code to communicate the retrieved information to the console of the program as seen in the figure below.

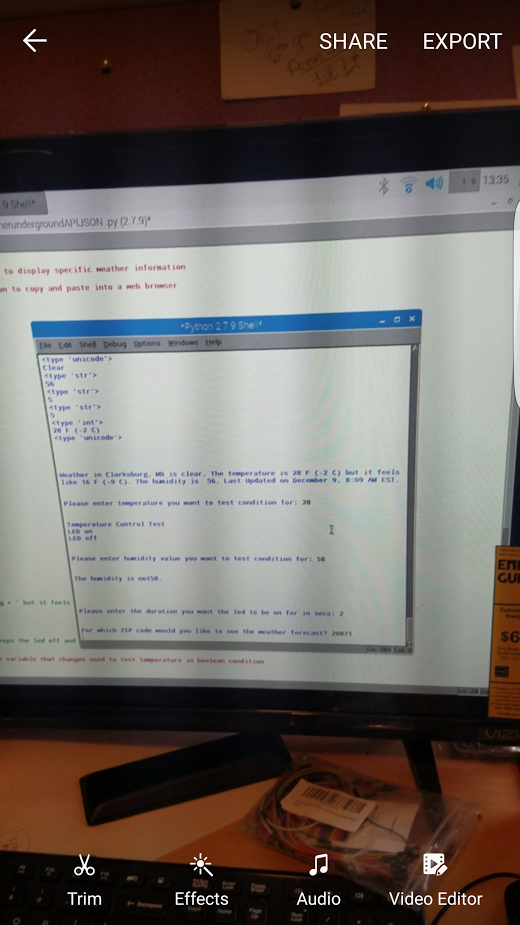


Figure 5 The terminal of the python integrated development program is used to output the retrieved weather information and the status of the Led lights.

Written in the body of python code used to retreive the weather information are some boolean conditions that determine whether or not certain weather information is or is not a desired number or matches a specific weather condition like sunny or cloudy. These boolean conditions can be changed by the programmer on his or her own accord. The action of turning the leds on and off is one of software and hardware interaction running the written source code using the IDE. Lastly onces the raspberry pi processes the boolean conditions written in the code and sets up the general purpose input and output pins it is able to send a 5 volts voltage signal to the series circuit. The 5 volt voltage signal is able to turn on the LED and indicate to the user weather or not the indoor temperature is above or below a desired temperature or wheather or not the humidity is above or below a certain percentage, etc.

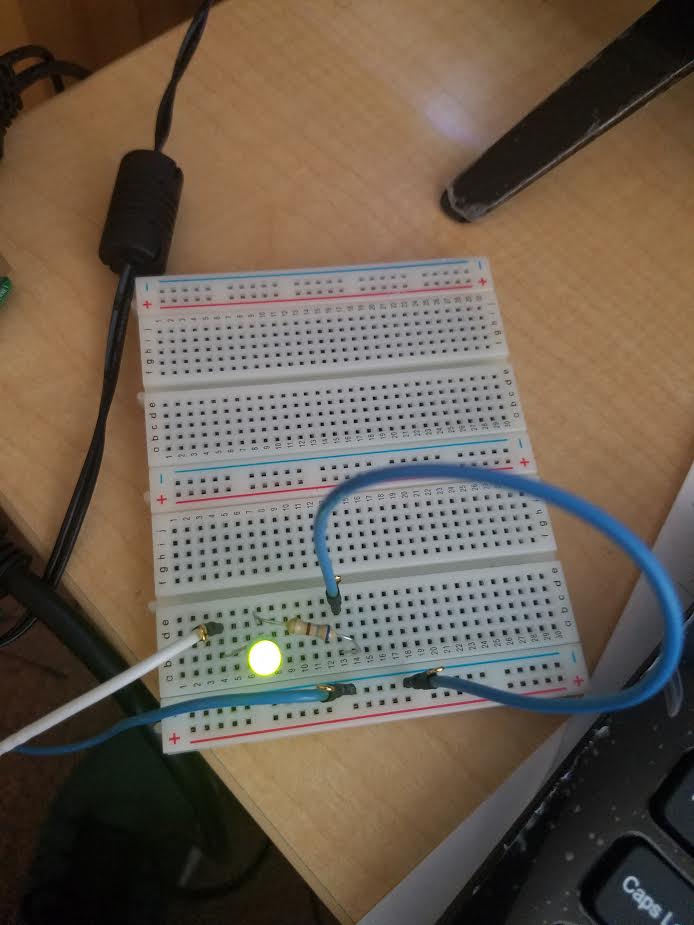


Figure 6. The led lights turn on after the code has run.

## Implementation of Task 4.

### Subtask 1-3. Demonstrate the ability to control an led based upon node-RED local server configuration.

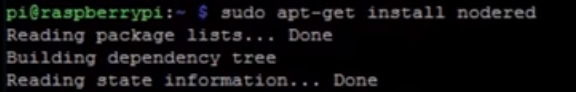


Figure . installation of node-red packages through raspberry pi shell

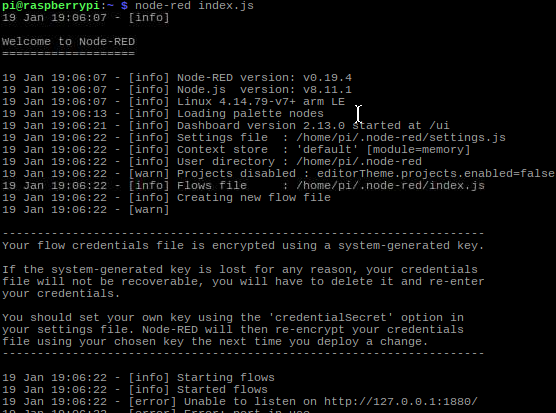


Figure Shell command to run local server on specified network web address

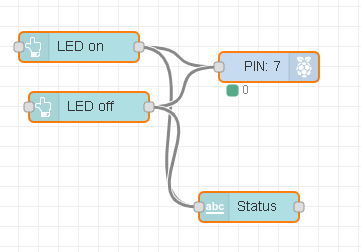
.

Figure 8.. The flow module diagram of inputs and outputs used to control an led in node-red workspace.

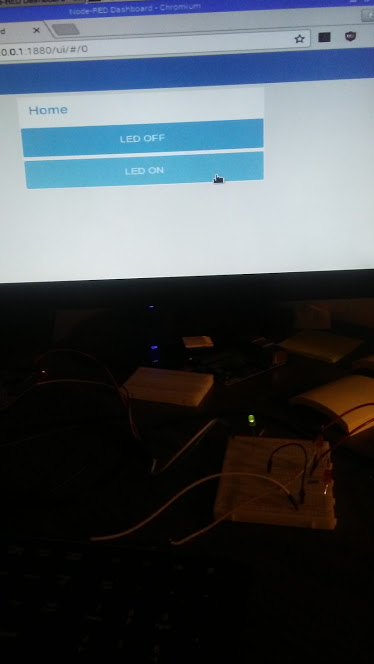


Figure . Led light on via local server and node-red web user interface

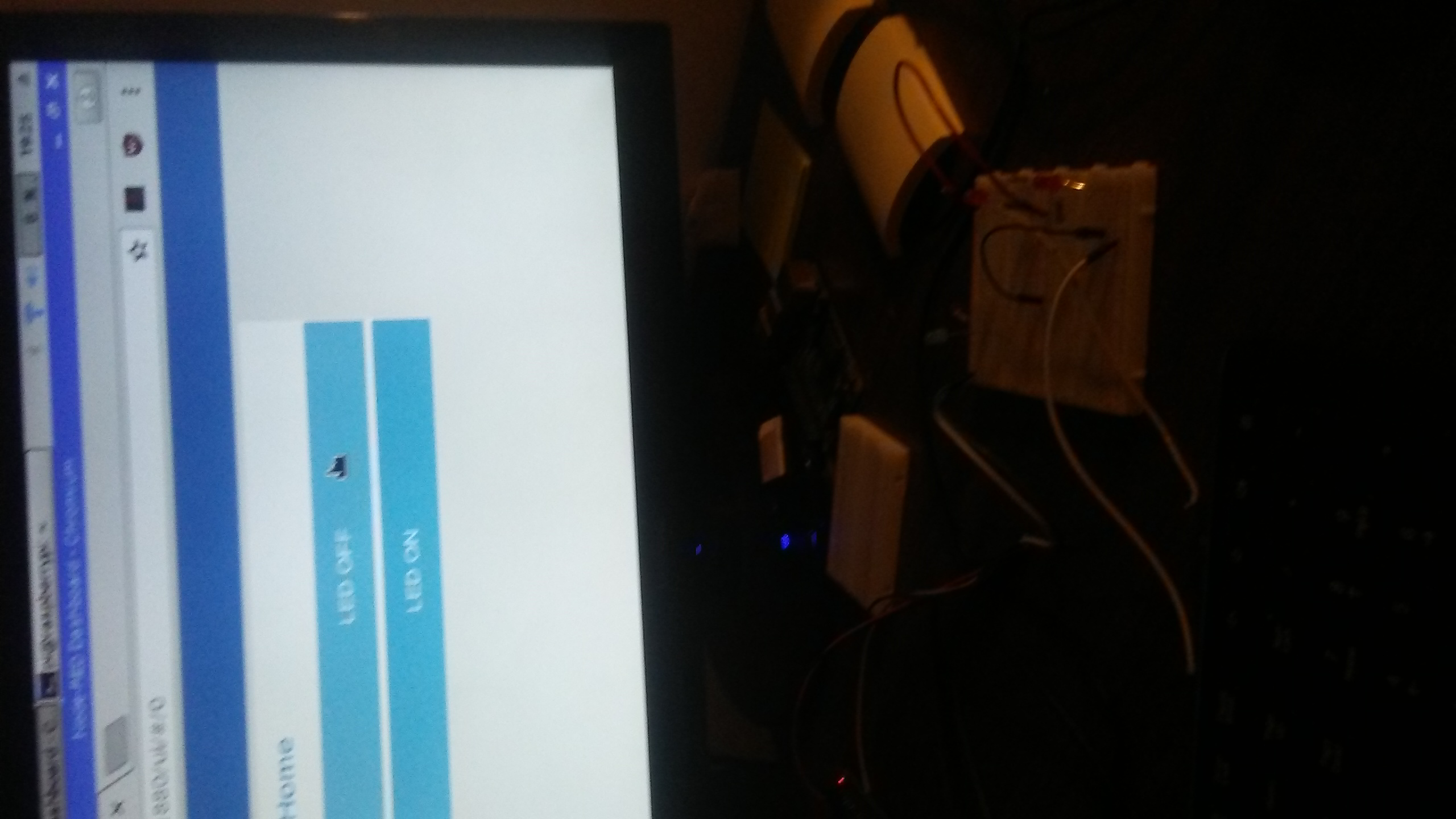


Figure . Led light off via local server and node-red web user interface

1. Conclusion.

By the end of the project, conclude the project and your learning experience.

Acknowledgement

If you get help or support from someone else (besides the team member and the advisor) and want to show your appreciation, put here (**do not include the advisor**).

Appendix

You can put reference info here, including:

1. specs of components used in the system,
2. source code (must be here but not in the body text), iii) CAD figures, etc.
3. Component Specs
4. Rapsberry Pi Specifications

The Raspberry Pi 2 Model B is the second-generation Raspberry Pi. It replaced the original Raspberry Pi 1 Model B+ in February 2015. It includes:

* A 900MHz quad-core ARM Cortex-A7 CPU
* 1GB RAM
* 4 USB ports
* 40 GPIO pins
* Full HDMI port
* Ethernet port
* Combined 3.5mm audio jack and composite video
* Camera interface (CSI)
* Display interface (DSI)
* Micro SD card slot
* Video Core IV 3D graphics core

Because it has an ARMv7 processor, it can run the full range of ARM GNU/Linux distributions, including Snappy Ubuntu Core, as well as Microsoft Windows 10. The Raspberry Pi 2 has an identical form factor to the previous (Pi 1) Model B+ and has complete compatibility with Raspberry Pi 1.

1. Specs of Arduino
2. Source Code.
3. Source Code for Weather Information Retrival

import urllib2

import json

import RPi.GPIO as GPIO #Import GPIO package

import time #Import time package

key = 'e89fb32ab64fd4ab' # Weatherunderground API key

while 1: # while loop to continue running the following code to the python shell to display weather information until user exits

print '\n\n\n'

timeOn\_Off =float(raw\_input('Please enter the duration you want the led to be on for in secs: '))

print '\n'

zip = raw\_input('For which ZIP code would you like to see the weather forecast? ') #Asks user for input to enter desired zip code to access information

url = 'http://api.wunderground.com/api/' + key + '/conditions/q/' + zip + '.json' #full url to contact webserver and communicate with the weather underground api

openUrl = urllib2.urlopen(url) #opens the url

json\_string = openUrl.read() # Retrieves the JSON object data from Weather Underground request over HTTP

parsed\_json = json.loads(json\_string) # Stores JSON object dictionary data into variable that will be later parsed to display specific weather information

print '\n\n\n'

print 'Copy and paste the following link into your web browser: '+ url # prints full url giving the user the option to copy and paste into a web browser

print '\n\n\n'

print json\_string #Prints the JSON data in column format

print '\n\n\n'

print parsed\_json #Prints the JSON data formatted more compactly

#state = parsed\_json['location']['state'] #Received an error because location not in json data dictionary list

city = parsed\_json['current\_observation']['display\_location']['full']

weather = parsed\_json['current\_observation']['weather']

weatherStrA = weather.encode("ascii")

# weatherStrU = weather.encode("utf-8") #encodes the unicode JSON data string to regular python string

humidity = parsed\_json['current\_observation']['relative\_humidity']

temperature\_string = parsed\_json['current\_observation']['temperature\_string']

tempNum = int (parsed\_json['current\_observation']['temp\_f'])

feelslike\_string = parsed\_json['current\_observation']['feelslike\_string']

humStr= humidity.encode("ascii")

humFStr = humStr[:-1]

humNFStr = int(humFStr)

lastUpdate = parsed\_json['current\_observation']['observation\_time']

print '\n\n\n'

print 'Weather in ' + city + ' is ' + weather.lower() + '. The temperature is ' + temperature\_string + ' but it feels like ' + feelslike\_string + '. The humidity is ' + humidity + '. ' + lastUpdate + '.'

openUrl.close()

'''

The following section of code determines if the temperature is a certain value

then turns the led on for desired length of time if temperature meets the boolean condition else keeps the led off and displays that it didn't meet the condition

'''

print '\n'

tempBC = int(raw\_input('Please enter temperature you want to test condition for: ')) #integer type variable that changes used to test temperature in boolean condition

if tempNum == tempBC:

GPIO.setmode(GPIO.BCM) #Setup gpio to communicate with board h/e not communicating with boar

#GPIO.setwarnings(False)

GPIO.setup(21, GPIO.OUT) #Sets up specific # pin on general purpose input/output pin as output

print '\n'

print 'Temperature Control Test'

print 'LED on' #print led on to show that the led is on

GPIO.output(21, GPIO.HIGH) #Sends either 5v or 3.3 volts to this GPIO pin

time.sleep((timeOn\_Off)) # Holds program for n secs

print 'LED off' #prints led off to show that led is now off

GPIO.output(21, GPIO.LOW) #Sends 0v to this GPIO pin to turn off led

time.sleep((timeOn\_Off)) # Holds program for n secs

GPIO.cleanup() #Resets the GPIO values used into this program back to their default values

else:

print'\n'

print 'The temperature is not ' + str(tempBC) + '.'

# enclose code in ''' \_ ''' to do block comment

'''

The following section of code determines if the humdity is a certain value

then turns the led on for desired length of time if humidity meets the boolean condition else keeps the led off and displays that it didn't meet the condition

'''

print '\n'

humBC = int(raw\_input('Please enter humidity value you want to test condition for: '))#integer type variable that changes used to test humidity in boolean condition

if humNFStr humBC: # old booleand conditon: humNFStr < 70

GPIO.setmode(GPIO.BCM) #Setup gpio to communicate with board h/e not communicating with boar

#GPIO.setwarnings(False)

GPIO.setup(26, GPIO.OUT) #Sets up specific # pin on general purpose input/output pin as output

print '\n'

print 'Humidity Control Test'

print 'LED on' #print led on to show that the led is on

GPIO.output(26, GPIO.HIGH) #Sends either 5v or 3.3 volts to this GPIO pin

time.sleep((timeOn\_Off)) # Holds program for n secs

print 'LED off' #prints led off to show that led is now off

GPIO.output(26, GPIO.LOW) #Sends 0v to this GPIO pin to turn off led

time.sleep((timeOn\_Off)) # Holds program for n secs

GPIO.cleanup() #Resets the GPIO values used into this program back to their default values

else:

print'\n'

print 'The humidity is not ' + str(humBC) + '.'

1. Source Code of Robotic Arm

…

REFERENCES

[1] Gigli, M., & Koo, S. (2011, July). Internet of Things: Services and Applications Categorization. Retrieved January 28, 2016, from http://file.scirp.org/pdf/AIT20110200005\_90413785.pdf

[2] Bojanova, I. (2015, March 31). What Makes Up the Internet of Things? Retrieved December 28, 2016, from https://www.computer.org/web/sensing-iot/content?g=53926943&type=article&urlTitle=what-are-the-components-of-iot-

[3] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (n.d.). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Retrieved January 13, from https://arxiv.org/ftp/arxiv/papers/1207/1207.0203.pdf

[4] Lu, J., & Sookoor, T. (n.d.). The Smart Thermostat: Using Occupancy Sensors to Save Energy in Homes. Retrieved January 8, 2017, from http://www.alice.virginia.edu/~stankovic/psfiles/sensys10-final169.pdf