
# Smart Home Monitoring and Control System

**by**
**Cristian Orna**
**Matthew Acosta**

**Faculty Advisor: Prof. Dmitri Donetski**
**Senior Design Project**
**September 2018**

## Executive Summary

Our goal was to design and build an automated smart home control system. This control system is a mobile application designed for Android phones. The application is able to control other auxiliary systems, such as a light control system, a heating, ventilation and air conditioning (HVAC) control system, and a water sprinkler control system. Each system is connected to the application through a WiFi module. The project primarily consists of designing and programming the control application, which was coded using MIT App Inventor software. This application was designed in a way that it can fulfill the user demands when it comes to smart home systems by being low cost and easy to install with no modifications to the property. Though we originally considered having a physical control center that would work with the app, we decided that having the app as the main control would be simpler and more cost effective, and would require less maintenance from the user. It would also reduce the work needed for designing the system. The app was tested by building low cost circuits representing each system. The app was tested by building circuits representing each system. Each circuit contains a connection directly to the app, which the user can control. In other words, the user will send commands via the mobile app and the circuits must gather the information process it and execute it. This function was done through the WiFi module. This module was coded using Arduino IDE.

## Table of Contents

## Introduction

### Project statement

Our goal is to design a mobile application that will help us control various auxiliary systems through WiFi modules. Due to time constraints, we will focus on three auxiliary systems: a light control system, a heating, ventilation and air conditioning (HVAC) control system, and a water sprinkler control system. The lighting control system will detect motion in a room and turn it on, then turn it off after a period set by the user. The HVAC control system will gather sense the temperature and determine whether the room needs to be heated or cooled depending on temperature range set by the user. Then, the system will open the vents and activate either the heating or cooling system accordingly. The sprinkler control will allow the user to schedule when the water sprinkler system activates. The application will be able to monitor and control these systems.

### Project Goals:

- Connectivity to any specified WiFi module
- Ability to send commands to activate or deactivate each system
- Communication to each system sensor
- User ability to add and designate systems to specific WiFi module
- Design UI
- Operation of the app:
  - Implement a Sprinkler Control System
    - Can schedule times to activate
    - Activates at scheduled time
    - Activates if moisture level goes below user specified threshold
    - Deactivates, or does not activate at scheduled time if moisture level is above user specified threshold
  - Implement a Lighting Control System
    - Can gather the data sent from hardware

    - Activates when motion is detected
    - Deactivates after user specified period
    - Can activated or deactivated manually by the user
- Implement an HVAC Control System
    - Can read the data from sensor
    - Temperature range can be set by user
    - Heat activates when the temperature is too low
    - Air conditioning activates when temperature is too high
    - Ventilation activates when either heating or A/C are activated
    - User can manually activate or deactivate any part of HVAC system

**Background**

➢ **Internet Of things (IoT)**

As time progresses the Internet of Things, commonly known as (IoT), has become one of the main topics for development and implementation of new technology. Internet of Things can be defined as the interconnections of many devices in a place that is the internet. The internet is a universal system that connects many networks via communication protocols like the internet protocol address (IP address). In other words, devices that have the right kind of hardware and software that are available to connect and communicate in the internet are consider smart devices and run in this environment of IoT. Some examples, are cars, cellphones, smart surveillance, etc. Take for instance, smart hubs like the Amazon Echo which is considered part of this interface in which a person through voice command can get information like the news, hear music from different applications, access the weather, get notifications from social media, and much more. In fact, there are even cities now that are connected in IoT and are considered smart cities. This consist of installing different devices all over the city, that contain many electronic pieces like sensors that are available to exchange, gather, and interpret different kinds of data in the internet. This helps the city gather information like traffic, weather, GPS and more. The information is store in the cloud and the users, and other devices can access it thought their phones, computers, tablets, and electronic systems as long as they have an internet connection

## ➢ Smart Home Automated Systems

The idea of an automated home has been around for a very long time. Think about the different home function that appeared in back to the future II, for instance the home entertainment and the Kitchen that was automated [1]. In fact, many scientist scientists like Leonardo Da Vinci created devices and concepts that can think for themselves for purpose creating a home that can maintain itself [2]. This was before the first automated home system was invented, which was the ECHO IV develop in 1966 by Jeff Bezos [3]. After, in 1979 X10 was invented the better version of the echo IV [4]. Since then home automatization has increased popularity dramatically due to the improvement in technology like the creation and development of IoT. Now, a smart home automated system can be defined as the integration of technology within a home to help improve the living environment in which people can benefit in many different ways, such as bringing safety to the home by controlling cameras and other electronic equipment.

**Existing Smart Home systems**

In today's modern world there are different types of smart home systems available to the public. Each of this system perform different kinds of functions and are available to help improve people's live. Some of the systems include:

- **Control4**

Control4 is a one of the most popular smart home automated systems in the market. This system can perform many more functions than other systems. This is because the system is compatible with other devices, that share the same connectivity like WiFi or Bluetooth. Also, is user friendly. Meaning that it has the user can control and access the system through different devices. According to their website the system can perform the following: control the lighting in the home, entertainment systems, climate system, and blinds and shades system [5]. One problem with this system is that for every system that a user wants to control they would have to buy different products that add to the Control4. Which means that people need to have different technologies to operate the different controls. For example, in their website the company mentions that for climate control the system can be only use if there is existing HVAC, radiant

flooring, forced air, dual fuel and geothermal system in the house. Control4 is quite expensive and depending on what the user wants to control it can increase to thousands of dollars. Meaning that the company mainly targets the upper class. Other problems are mentioned by Lobaccaro, Carlucii, and Lofstrom which state that "The system must be installed professionally by an authorized dealer. It offers limited mobile access functionality with its base system setup. The Help & Support section of the system is poorly performed." [6]

- **Crestron**

Creston is another automated system that is very popular. This system has the same functionality as Control4, but it's more compatible with other devices. Also, it can control different systems at the same time allowing the user to keep adding new appliances without having to upgrade every time. This company also makes this product very user friendly. Some drawbacks to this system are that it can only be installed by professionals and that it doesn't monitor its energy use [7]. This product is also quite expensive and, like the previous home system, if you want to add more functions you have to pay more, which makes it difficult for lower class people to afford this.

**Customers and their Requirements**

Automated smart home systems are not being adopted at high rate because they require specialized technicians to maintain and are costly as a result. This project is trying to create a system that is user friendly and cost efficient to the user. Therefore, we took the following considerations when designing this project. First, it had to be user-friendly, easy to use and maintain. Second, it has to have the ability administrate the components and devices. Third, the user has to have full control of the app; thus, it has to have the ability to create profiles that contain the person's preferences when it comes to functioning their residence. Fourth, when developing the hardware to show the how the app works it has to have be low cost, so that user can use it in there on home, and also easy to install and maintain. Since, there is no estimate of how many people live in the home. Thus, it has to be able to connect multiple user to the same network. As mentioned before the app will be able to connect to preexisting technology in the

home. Thus, it has to be comparable to many devices. In the other hand, there are only four conditions that the user has to have which are that they must own a smartphone, have wifi connection in their home, willing to install, and invest in electronics that have wifi included in them. The last two conditions will depend on how technical the user is and how much effort they want to put on the devices. For example, if the user has no electrical background the best way is to buy the systems or have someone make or installed one.

**Engineering Specifications**

The application will be an Android application. Each major component of an Android application is known as an 'activity.' An activity is a group of related functions that user can perform, shown as functions on a single screen. The app will have a 'main activity' that can transfer control to other activities. Only one activity will have control of the app at time.

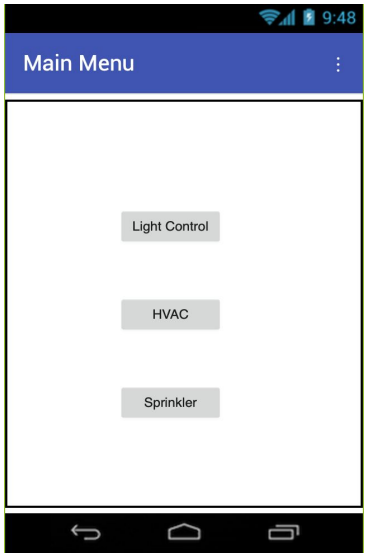



Figure 1. Main Menu

The app's main activity will serve as the main menu. It's primary function is to allow the user to access the control menus for each system. Each button calls the corresponding activity and transfers control to it. The screen will then transition to the proper control menu. The main activity also establishes a connection with all of the wireless modules on startup. The microcontrollers for the wireless modules are configured to act as WiFi access points. The app will connect to each module and ensure proper communication.

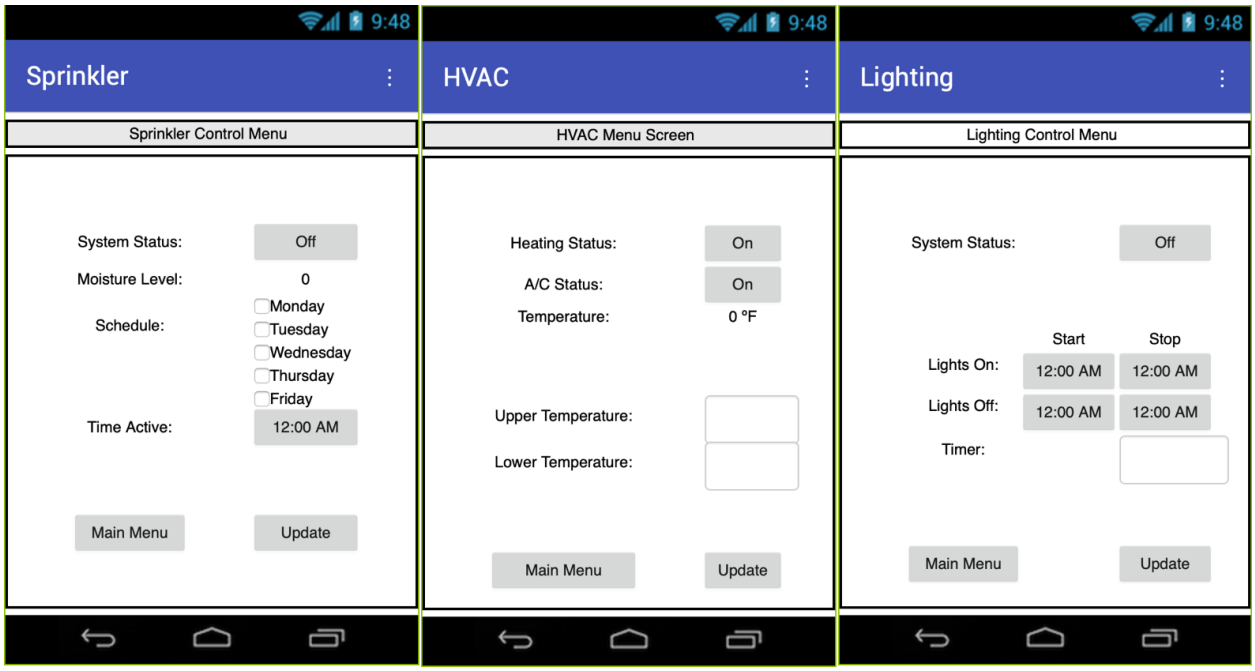



Figure 2. Control Menus

Each control displays the system status, sensor data from the wireless module, and the user settings that define the operation of each system. The user can check the system status, turn the system on or off, and change the system settings.

The app will be tested by building different circuits. The circuits we are trying to elaborate are low cost, and are constructed in a way that we could connect them to the traditional system already built in a home. Basically they will act like a switch to turn on and off the the different systems. The lighting system circuit will be built so we could attach it directly to the lighting sockets already build in a house. In the same way, the other circuits will be connected to the HVAC system meaning the fans the boiler and other electrical components that for thet system. The irrigation system also has to be connected to the water mains in the home. This allows a traditional home to become a smart home system at a low cost.

Since we want to communicate via WiFi, we need a WiFi module in this case the ESP8266. There are many versions of this device but the most adequate is a development board because it has other components that could reduce the price of the project. The board that is going to be used in this project is a NodeMCU. This is development board that has a serial

TTL-to-USB adapter bridge and a Micro-USB connector coupled, which will help us code the device without any external components. Also, it has a 3.3V power regulator already built into the board, which reduce the power consumption from the circuit. The board will be used to have the app communicate with the circuits. The microcontroller will be used to process commands from the app and transmit data from the sensors.


Figure 3: System Communication

There will be three different circuits which will have a sensor and a NodeMCU as shown in the figure above. The first circuit will have a Digital Temperature Sensor which will measure the temperature in the room.. The sensor reads the temperature in degrees Celsius. The app will convert it to degrees Fahrenheit, if the user desires. The WiFi module will transmit the sensor data to app, and the app will decide whether the system should be activated or not. The app sends commands to the WiFi module to either activate or deactivate the system. The second circuit will contain a Soil Moisture Sensor, which will measure the moisture in the soil. The value will be represented in percent of water in the soil. The sensor will transmit the sensor data to the app, and the app will use this to decide which command to send to the WiFi module. Lastly, the last circuit has a motion sensors that will detect if there is any movement in the area. When the sensor detects movement, the WiFi module will transmit the data to the app. The app will transmit the command to either activate or deactivate the lights based on the system status, sensor data, and the current time.

The NodeMcu Pin layout it's as follow:

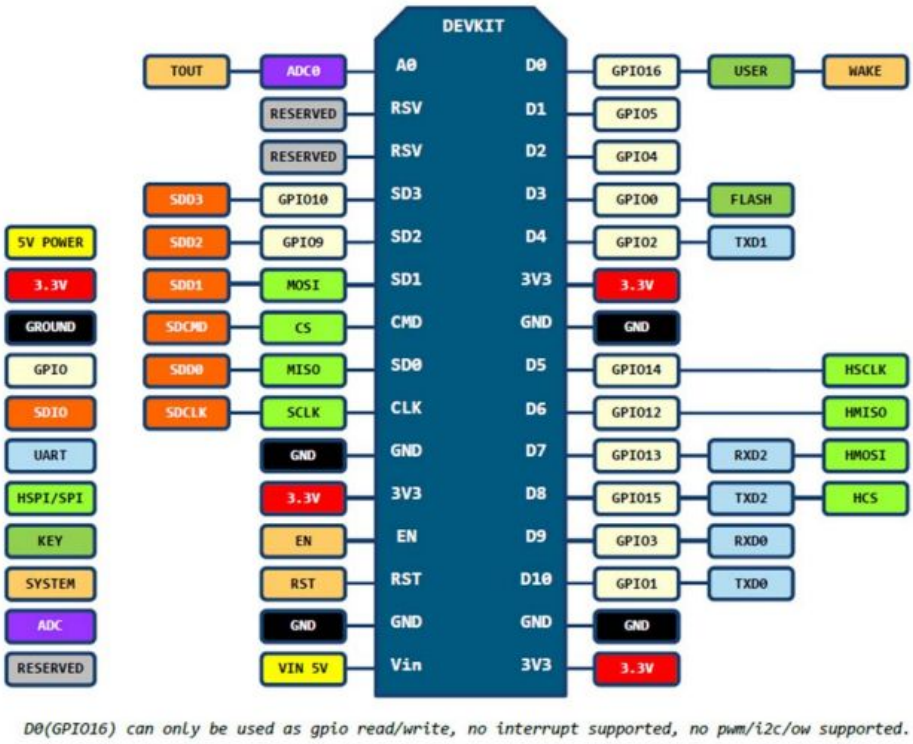



Figure 3: NodeMcu Pin Layout[8]

The I/O of the circuits and its  will operation is as follow

| **Systems** | **Commands** | |
|---|---|---|
| | **Send** | **Read** |
| **Lighting** | -From the GPIO 05-15 pin D2-D10 can be my outputs. Depending on how many lightbulbs are going to be connected.In our case is only one so pin D2 is the output. It will send a 1 to turn on the light bulb and 0 to turn off. | -GPIO05 pin D1 will work as the input from the sensor. The sensor will assert 1 if there is motion in the room, else a 0. |
| **HVAC** | -Pins D2-D4  are going to be the output of the system.<br>-D2 is connected to Heating.<br>-D3 is connected to ventilation.<br>-D4 is connected to the AC.<br>-It will send a 1 to turn on the and 0 to turn off the respective component. .<br>- If the sensor detects that the temperature has fallen below the user's input temperature , it will send 1 to PinD2 to activate Heating and 0 to Pin D4 to deactivate the AC. If the temperature is above this range, it will due the opposite. The Pin D4 will activate if either the heating or the air-conditioning Pins are 1 . | -GPIO05 pin D1 will work as the digital input from the sensor. The sensor operates with 12 bits thus it will send the binary value through this pin to store in the memory of the NodeMCU. |
| **Sprinkler** | -From the GPIO 05-15 pins D2-D10 can be my outputs. Depending on how many sprinklers are going to be connected. In our case is only one so pin D2 is the output. It | .-GPIO05 pin D1 will work as the digital input from the sensor. |

| | | |
|---|---|---|
| | will send a 1 to turn on the sprinkler and 0 to turn off. | |

The circuits explained previously will contained the following parts. Table 1 shows the main components of the circuits. It includes the name of the part, the amount that is going to use and the cost of each component. It also contains components need in each of the circuit.

| Part | Part Name | Units | Cost |
|---|---|---|---|
| Temperature Sensor | DS18B20 | 1 | $3.05 |
| Soil Moisture Sensor | SEN-13322 | 1 | $4.95 |
| Motion Sensor | IRA-S210ST01 | 1 | $3.12 |
| Wifi Module | ESP8266 | 1 | $6.95 |
| Transistor | 2N700 | 1 | $0.39 |
| LDR | 485-161 | 1 | $0.95 |
| Light Bulb | | 1 | $2 |
| Water Valve | | 1 | $17 |
| Leds | | 3 | $1 |

Table 1: Sensors and parts

The three circuits will be connected to the power supplies of the residence; thus we will need power supply to convert a AC to DC converter. This is useful because we need to transform the 120VAC that is outputted from a normal power supply outlets to 5V that that NodeMCU uses to operate. In other words, this component will help us power on the NodeMCU. According to Digikey there are power supplies that will convert the voltages. Some of the parts are helpful to convert because it regulates and short-circuit-proof isolated DC outputs, low standby power consumption and has a -25°C to +80°C operating temperature range. Also, some have a built-in Class A / FCC Part 15 EMC filter, which are certified to EN60335, EN60950 and EN62368 safety standards and come with a three year warranty[9].

Figure 5: 1-channel Relay[10]

Since our circuits need to turn on and off the light bulbs, sprinklers, and the HVAC systems respectively we need 3 relays. The electrical component will be connected to the NodeMCU. In the market there are relays with different number of channels, depending on the components that are going to be connected. A relay can be defined as switch that controls a High current circuits, which are the light bulbs, sprinklers and other system, with a low current signal. This will help us operate the systems as instructed by the app.

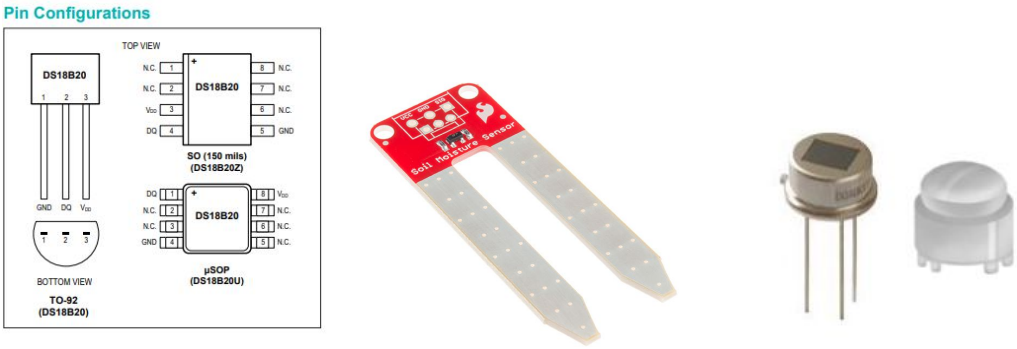



**Figure 6: Sensors**

As mentioned before the NodeMcu will be connected to a sensor, which depends on the system. Figure 6. Shows the sensors that the project will contain. The temperature sensor will be connected to the HVAC system to determine the operation of the app. The sensor chosen for the project is the DS18B20, which is digital thermometer which provides the room temperature in a 12-bit Celsius output, which will be send to the nodemcu to store it. Also, according to their website it has an alarm function with nonvolatile user-programmable upper and lower trigger points[11]. This sensor has electrical characteristics that match the NodeMCU, so that it can operate properly.  The irrigation circuit will have a soil moisture sensor 1568-1360-ND. Which will measure the amount of water in the soil. The sensor will send analog signals to the NodeMCU which will store and convert  to digital, and then send back to the app. In other sensor will act the same way as the previous one, but it will detect motion in an area. The sensor is

called IRA-S210ST01, which is an infrared sensor that use ceramic to detect motion. This is also a analog sensor. Thus, the NodeMcu has to interpret the input data.

**Design Concepts**

The smart home system initially proposed by the team members and instructor was to design and build an automated smart home control system. This would be a physical control system that would be able to three other auxiliary systems: a light control system, a heating, ventilation and air conditioning (HVAC) control system, and a water sprinkler control system. The system would consist of a main control center which would monitor and control the other systems, and three separate systems connected to main control center through wireless modules. The modules would activate or deactivate the system, and send sensor data to the control center. The user would be able to control each system through the control center and the control would communicate with the wireless modules to receive data about the system status or send commands to the module. The control center would also be able to be controlled wirelessly through an Android application. We design a simple version of the auxiliary systems to show how they would interact with the module.

The goal of our project largely remained the same. However, due to time and cost restraints, we decided that designing a physical control center along with the actual systems would not be feasible and decided to switch to a primarily software based project. Our primary concern when designing this project was ensuring that it easy to use, maintain, and that it was inexpensive for the user. So, instead of designing a physical control center, the Android application would now be the main control for the system. The app will now monitor and control each auxiliary system and there will be no physical control center.

The mobile app was first proposed to be created using Android Studio. However, after carefully doing research and decision making we decided to changed the integrated development environment (IDE) from Android Studio to MIT App Inventor, a open-source web application. It is a cloud-based integrated development environment, which allows users to create projects from a web browser. User create programs using virtual instruction blocks which can combined to build the logic for an application. The reason we changed to MIT App inventor was that

attempting to develop the app using Android Studio required extensive knowledge of Java syntax and Android libraries that neither of us had. MIT App Inventor abstracts programming using code blocks with built-in functions and a simple interface. This allows us to focus on developing the logic of the application without worrying about the intricacies of Java and Android programming. Furthermore, it is well documented, so learning to the interface is easy. The tradeoff for this is that we won't be able to access the full capabilities of the Android programming environment, but this is ultimately not a concern for us as MIT App Inventor is powerful enough for our needs.

After taking some time to familiarize ourselves with App Inventor interface, the first task we accomplished was to recreate the user interface as mentioned in the previous section and we obtained the interface shown in Figure 7. , and Figure 8.

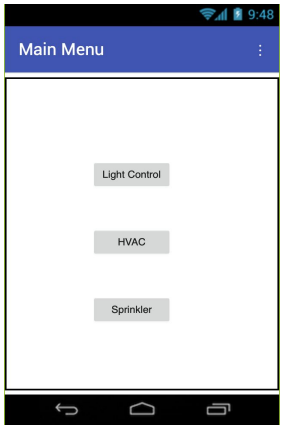



**Figure 7: Main menu**

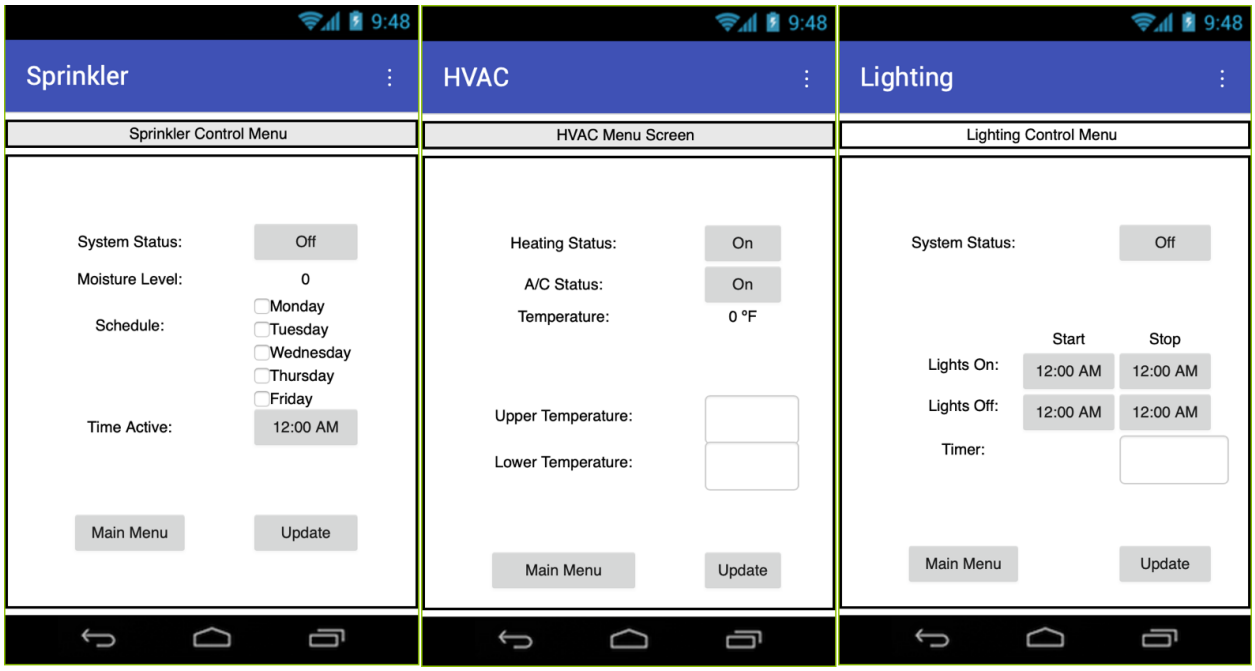



**Figure 8. Subpages**

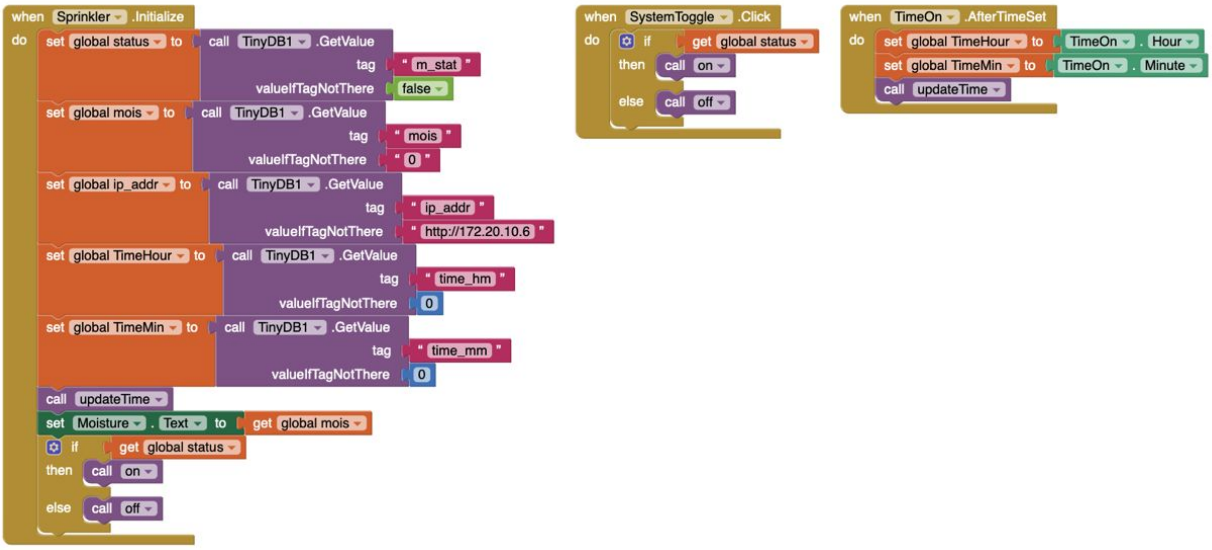
After, the interface was done the functionalities of each Android page was created. The main menu shown in Figure 7. allows the user to access each of the control menus. In Figure 8 we have 3 subpages that will control the HVAC, Sprinkler, and Lighting. The subpages have the following functionalities:

- HVAC Control:
  - ➢ Heating Status: Shows and toggles status of heating system.
  - ➢ A/C Status: Shows and toggles status of A/C system.
  - ➢ Temperature: Shows current temperature.
  - ➢ Upper Temperature: Sets the temperature at which the A/C system will automatically activate or deactivate.
  - ➢ Lower Temperature: Sets the temperature at which the heating system will automatically activate or deactivate.
- Lighting Control:
  - ➢ System Status: Shows and toggles the status of lighting system
  - ➢ Lights On: Sets a time where the lights are always on
  - ➢ Lights Off: Sets a time where the lights are always off

- ➢ Timer: Sets how long the lights remain on when activated

- Sprinkler Control:
  - ➢ Timer: Sets how long the lights remain on when activated
  - ➢ System Status: Shows and toggles the status of sprinkler system
  - ➢ Moisture Level: Shows current moisture level
  - ➢ Schedule: Displays and changes the days of the week that the sprinklers run
  - ➢ Time Active: Displays the time of day that the sprinklers run

As was mentioned before the interface and functionality of the App was programmed using MIT App Inventor. the code is the following:




Moreover, App needed to be tested to see if it worked or not. Thus, we establishing a connection between the app and the WiFi module. We discovered that MIT App Inventor has a function that allows apps to make HTTP requests to a web server. Through this function, we were able to devise a control method to send commands to the WiFi module. The module is set up to act as a WiFi access point that the user's phone can connect to. The app can then connect to the module by sending POST requests to its IP address and sending the command signal using JSON tags. The module has several specific addresses that correspond to either activating or deactivating each system. The specifics of the programming of the module are discussed in more detail in the hardware section below. To activate or deactivate any system, the app calls the

specific address and sends the command. The module receives the incoming request to the address, processes it, and performs the proper command. Through this method, we were able to control each system wireless through the app. We implemented commands through controls in the user interface.

Also, Circuits were created as mentioned in the previous section to test the app. This circuits were design after doing careful and extended research and calculations. First, we designed a circuits representing the HVAC, Sprinkler, and lighting systems. The HVAC system is going to be represented as followed:

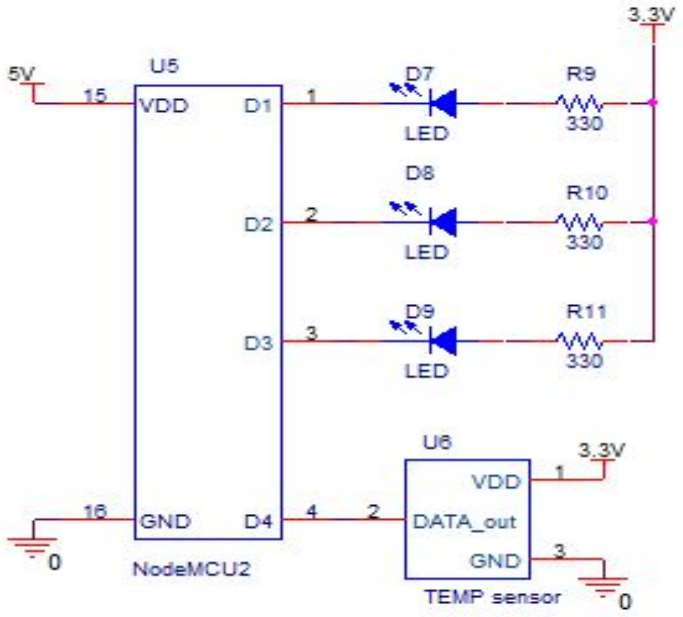



Figure. 9

A traditional HVAC system contained different mechanical and electrical components that are very hard to built in the short time we have for the project. Thus, instead we develop a circuit shown in Figure. 1, which is going to help the user better understand the operation of the app with the system. The circuit uses the following components:

| Part | Description |
|---|---|
| LEDs | Three normal LEDs |
| Temperature Sensor | One digital temperature Sensor |

| | |
|---|---|
| | DS20B18 |
| Wifi Moule | NodeMCU |
| Resistors | Three 330 ohm resistors |

Table. 2

The leds are going to be use to represent the heating, ventilation, and A/C. They will be shown in three different colors, so that the user can differentiate them from one another. The WiFi module will read the data from the temperature sensors and send it to the app so that it can process it and then send commands back to the module to operate it. The app will have a temperature set by the user, and it will be use to send commands to the module to operate as follow:

- Heat activates when the temperature is below the fix temperature.
- Air conditioning activates when temperature is too high.
- Ventilation activates when either heating or A/C are activated
- User can manually activate or deactivate any part of HVAC system.

Therefore, the circuit then will turn on and off the LEDs depending if the system is activated or deactivated. As we can see the WiFi module will be the one driving the circuit, so it was coded to performed the above functions from the app. We have initialized the coding, which will be using Arduino IDE. The coding is as followed :

```
void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println('\n');
  pinMode(D0, OUTPUT);
  digitalWrite(D0, HIGH);
  pinMode(D1, OUTPUT);
  digitalWrite(D1, HIGH);
  pinMode(D2, OUTPUT);
  digitalWrite(D2, HIGH);
  pinMode(D8, OUTPUT);
  digitalWrite(D8, HIGH);

  Serial.println("Temperature Sensor NodeMCU:\n");

  // Start up the library
  DS18B20.begin();
  wifiManager.autoConnect("AutoConnectAP");

  Serial.println("\nSuccessfully connected");
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());

  String message = "{\"ip\":\"";
  message += WiFi.localIP().toString();
  message += "\"}\n";
  Serial.println(message);

  server.on("/stat/h", handleStatH);
  server.on("/stat/ac", handleStatAC);
  server.on("/data", handleData);
  server.on("/wifi", handleWiFi); //Associate the handler function to the path
```

Moreover, for the lightning system control will have the following functions

- Activates when motion is detected
- Deactivates after user specified period
- Can activated or deactivated manually by the user

Thus, the app will send on and off commands to the WiFi module. It will read the data and will activate and deactivate the system accordingly.

We created the circuit design to test the mention functionality. This circuit is shown in figure 8. The components used in this circuit are as shown in table 2.

| Part | Description |
|---|---|
| Optotriac | MOC3021 |
| Motion Sensor | 555-28027-ND |
| Wifi Moule | NodeMCU |
| Triac | BT136 |

| | |
|---|---|
| Light Bulb | 120 VAC at 60Hz |

Table. 2

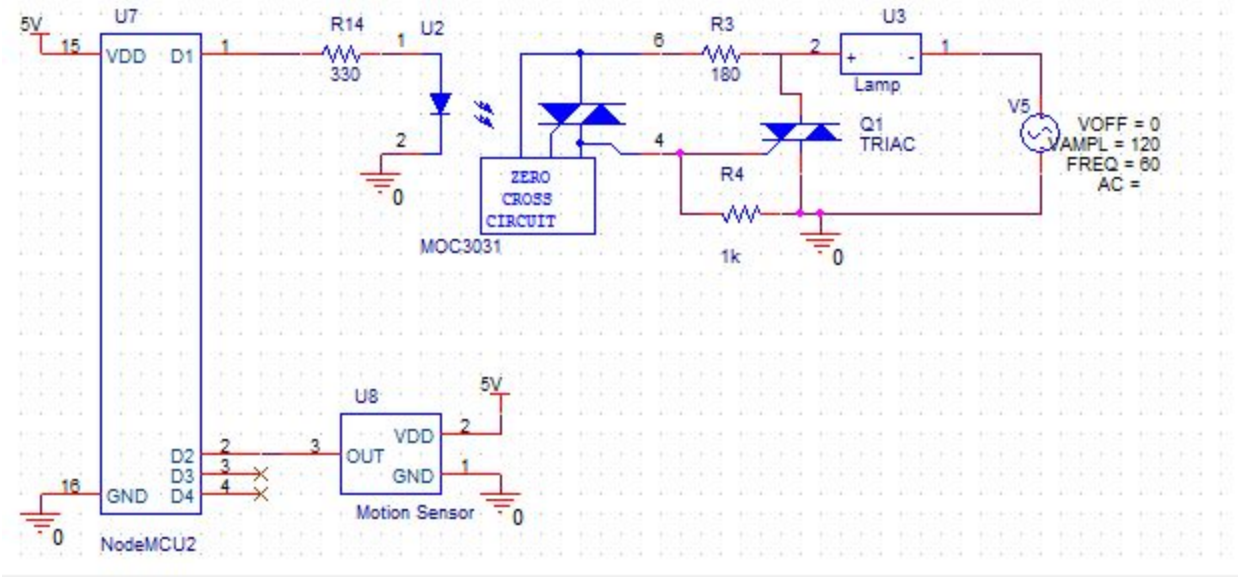



Figure. 8

In Figure. 8 the driver is the nodeMCU, which will gather information from the sensor and the app to perform different functions. The module will send 1 or 0 to activate the light bulb. Since, the light bulb operates in AC and the module in DC we created a solid state relay. We connect the optotriac to the nodeMCU. Optotriac are devices use to isolate loads. So, when the nodeMCU sends 1. It will activate the led in the optotriac. This will help turn on the triac inside. Opto Triacs are recommended to be used as a triggering device but if the load is high AC, it should not be used alone. This is mention on device data sheet in page 4 [1]. Therefore, we added another triac to drive the light bulb connected on the socket, which runs in 120VAC at 60Hz.

The module needs to be coded to gather, store and perform commands. Thus, we coded it in the same manner as the previous circuit and the code is shown below:

```
void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println("Motion Sensor NodeMCU:\n");
  pinMode(led, OUTPUT);      // declare LED as output
  digitalWrite(led, HIGH);
  pinMode(D7, INPUT);     // declare sensor as input
  pinMode(D8, OUTPUT);     // declare sensor as input
  digitalWrite(D8, HIGH);

  // Start up the library
  wifiManager.autoConnect("AutoConnectAP");

  Serial.println("\nSuccessfully connected");
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());

  String message = "{\"ip\":\"";
  message += WiFi.localIP().toString();
  message += "\"}\n";
  Serial.println(message);

  server.on("/stat", handleStat);
  server.on("/data", handleData);
  server.on("/timer", handleTimer);
  server.on("/override", handleOverride);
  server.on("/wifi", handleWiFi); //Associate the handler function to the path

  server.begin(); //Start the server
  Serial.println("Server listening\n");
  digitalWrite(led, LOW);
  message = "{\"stat\":\"0\"}\n";
}

void loop() {
  String message;
  server.handleClient(); //Handling of incoming requests
  //Serial.println(motionsen());
  if (!(override)) {
    if ((digitalRead(sensor) == HIGH) && ((millis() - start_scan) >= scan_time)) {
```

Furthermore, we also have advance in the sprinkler system. The app will help people schedule times to activate and deactivate the system.

- It will activate at scheduled time
- Also, it will activate if moisture level goes below user specified threshold
- It will deactivate if moisture level is above user specified threshold

This functions need to be tested, so we develop a circuit design. The circuit is represented in the figure below. The app and the circuit will operate a water valve. The components are shown in Table 3. The water valve operates at 24VAC that is why we convert 120VAC from the sockets to 24VAC to do this we used the transformer with 1:5 turns in the primary and secondary sections respectably. Attached to this is a solid state relay, which is also connected to the nodeMCU. The nodeMCU is the driver of the circuit. This circuit is very similar to lighting system, so it operates the same way. The app will send commands, the WiFi module will receive it, and perform different commands in the circuit.

| Part | Description |
|---|---|
| Optotriac | MOC3021 |
| Motion Sensor | 555-28027-ND |
| Wifi Moule | NodeMCU |
| Triac | BT136 |
| Electric water valve | 24VAC, Inrush current 0.4A, Holding current 0.2 A |
| Transformer | 1:5 transformer |

Table. 3

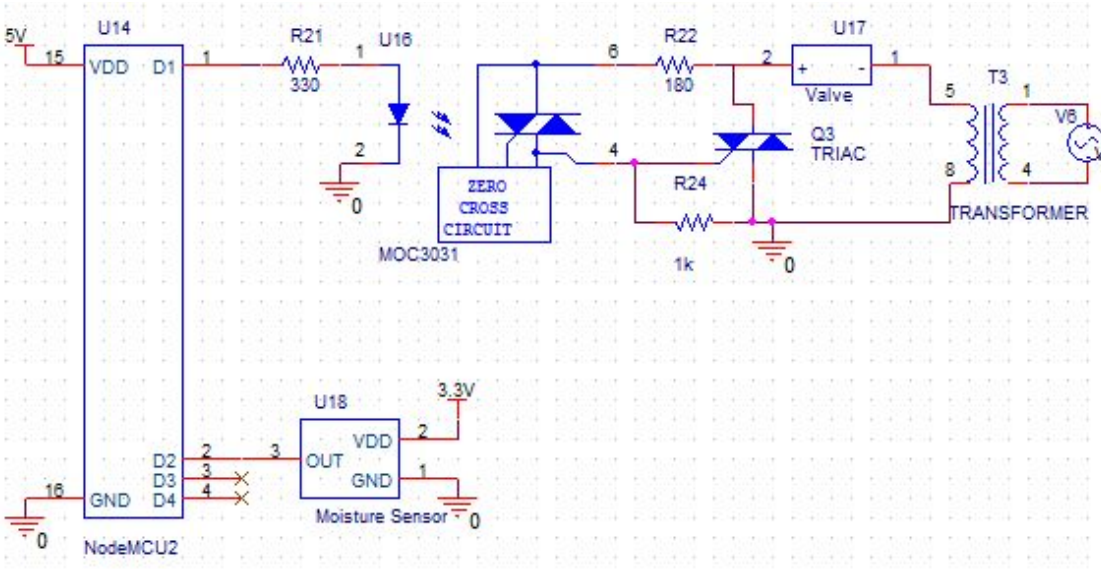



Figure.

The module needs to be coded to gather, store and perform commands. Thus, we coded it in the same manner as the previous circuit and the code is shown below

```
void setup() {
  Serial.begin(115200);   // open serial over USB
  delay(10);
  Serial.println("Moisture Sensor NodeMCU:\n");

  pinMode(led, OUTPUT);//Set D1 as an OUTPUT
  pinMode(soilPin, INPUT);//Set A0 as an INPUT
  pinMode(soilPower, OUTPUT);//Set D7 as an OUTPUT
  digitalWrite(soilPower, LOW);//Set to LOW so no power is flowing through the sensor

  // Start up the library
  wifiManager.autoConnect("AutoConnectAP");

  Serial.println("\nSuccessfully connected");
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());

  String message = "{\"ip\":\"";
  message += WiFi.localIP().toString();
  message += "\"}\n";
  Serial.println(message);

  server.on("/stat", handleStat);
  server.on("/data", handleData);
  server.on("/wifi", handleWiFi); //Associate the handler function to the path

  server.begin(); //Start the server
  Serial.println("Server listening");
}

void loop() {
  server.handleClient(); //Handling of incoming requests
  if ((millis() - start_scan) >= scan_time) {
    int sensorValue;
    sensorValue = readSoil();

    //Write what we want to desplay on the screen:
    Serial.print("Water Level: ");
    Serial.println(sensorValue);

    if (sensorValue <= thresholdDown) {
```

## Design Optimization

The user has the ability to control the system from anywhere, which gives the user more freedom and control. By making the system modular, systems can easily be added or removed with relative ease. The sprinkler systems traditionally depend on people to function. The system that we propose will help people reduce the manual labor that the traditional irrigation system requires.Also, the it helps reduce the amount of water that the traditional systems consume. Moreover, the automated light-controlled system is also going to benefit the user by implementing the following functions. First, the user will be able to turn on and off the lights through the app whenever they desire. Second, whenever the system detects motion the lights will turn on. Third, the people could schedule when, where, and for how long the lights will be on. Therefore, it will help reduce the amount of electricity wasted everyday by the user. Also, as the other systems the user will also be able to input when and where to turn it on. This functions together will not only increase the user economy but the national economy, and help reduce the global carbon footprint to reduce global warming.

The app will have little to no cost thus is very cost efficient for anyone to automate automate their residence. Also, we will test the app through circuits that are cost efficient by

using less expensive components. This design will be available for the user to use either to purchase the design built or they can make by doing it themselves. This can be possible because the mobile app will have the controller for all this systems.

Furthermore, a problem that many automated home systems have is that only one user can use the system. Our system will have the ability to be controlled by multiple users at the same time. Also, The app will be able to incorporate other devices that are already in the home, such as entertainment systems.

**Design Evaluation**

Our first design, the physical control center, has a few advantages over the control app. Having the control center as a physical component means that each unit functions exactly the same way. The uniformity of the design ensure behavior remains stable and predictable for each unit.

There are several disadvantages to this approach. First, this system would have to be installed and maintained by a technician, which would likely add cost to the user and make upkeep more difficult. This also makes upgrading the system more difficult. Upgrading a physical control system would require replacing physical components which could become costly and time-consuming. The average user would likely be unable to make these changes on their own.

Our second design, the control app, has several advantages over the control center. Since it is primarily software, this design is more cost effective since it requires less hardware, requires less maintenance, and only requires that the wireless modules be installed, not a physical control center. The app also has the added benefit of being able to be optimized and improved over time easily with no cost to the user, since the app simply need to install an update. The app is ultimately more user friendly and requires less technical maintenance from the user.

However, this approach does have certain disadvantages as well. Android is an open-source software and used as the operating system for a variety of mobile devices, each with different versions of Android. As a result, it’s impossible to ensure compatibility with every single Android device since each manufacturer is responsible for delivering software updates for

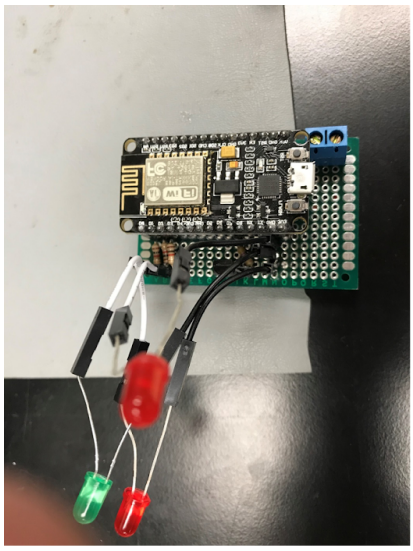
their devices. Using older versions of Android increases compatibility with older devices, but prevents us from using features in newer versions of Android. Similarly, using newer versions of Android allow us to take advantage of newer features and optimizations, but wouldn't be compatible with devices with older software. This may result in undesired or unpredictable behavior in incompatible devices. We decided to use an older of Android to maximize compatibility with most devices.

Another disadvantage of this approach is that new potential features are still limited by the hardware of the wireless components. They would have to have to be modified in order to add any major features to the control app. This would create an added cost to the user who would have to pay to upgrade the component.

This project present a lot of technical challenges. As a result, the work was divided so Matthew Acosta would be responsible for designing the app and user interface, and Cristian Orna would be responsible for designing the circuits for the wireless modules. Matthew Acosta had the most programming experience of the group, but had little experience in designing Android applications. Much of the design process was spent strengthening Java programming skills will learning how to design apps for Android systems. This was the biggest hurdle for the project. In the other hand, Cristian Orna knew more about the wifi module, so is incharge of designing the circuits to test the app. He started the first step towards this was finding the components for the circuit. The main consideration when choosing the hardware was trying to test the NodeMCU, for the electronic characteristics. This is needed because to operate properly all the components connected to the NodeMCU they have to match the electronic characteristics.

**Conclusion and Discussion**

Ultimately, were able to accomplish our major goals for this project. We built the circuits for the wireless modules and programmed the ESP8266s. We then built cases to demonstrate each system. We programmed the application were able to get it to successfully connect with the modules. The application was able to wirelessly activate and deactivate the systems. The systems were also able to activate or deactivate autonomously based on their sensor data.



In this project, we were able to apply knowledge we learned in class and in lab to create and build our design. We used this knowledge to design the power supply and microcontroller circuits for each subsystem. We also learned new skills as well. We learned how to work with Arduinos and wireless devices using C language. We worked with HTML and JSON tagging. We also gained valuable experience learning how to design apps with MIT App Inventor. We hope to able to use this skills in the future to design more complex projects.

**References**

[1] L. Kelion, "Back to the Future II: Hits and misses", BBC News,20-10-2015. [Online]. Available: https://www.bbc.com/news/technology-34569759. [Accessed: 24- Sep- 2018].

[2] “Figure 2f from: Irimia R, Gottschling M (2016) Taxonomic revision of Rochefortia Sw. (Ehretiaceae, Boraginales). Biodiversity Data Journal 4: e7720. https://doi.org/10.3897/BDJ.4.e7720.”

[3] “The ECHO IV Home Computer: 50 Years Later,” What Was The First PC? [Online]. Available: http://www.computerhistory.org/atchm/the-echo-iv-home-computer-50-years-later/. [Accessed: 24-Sep-2018].

[4] D. Rye, “My life at X10,” AV Systems magazine. [Online]. Available: https://web.archive.org/web/20140930080338/http://hometoys.com/emagazine.php?url=/htinews/oct99/articles/rye/rye.htm. [Accessed: 24-Sep-2018].

[5] “Comfort & Convenience,” Control4. [Online]. Available: https://www.control4.com/solutions/comfort-and-convenience. [Accessed: 24-Sep-2018].

[6] G. Lobaccaro, S. Carlucci, and E. Löfström, “A Review of Systems and Technologies for Smart Homes and Smart Grids,” Academia.edu . [Online]. Available: http://www.academia.edu/36251537/A_Review_of_Systems_and_Technologies_for_Smart_Homes_and_Smart_Grids. [Accessed: 24-Sep-2018].

[7] “Wireless Presentation Upgrade Program,” Company Overview [Crestron Electronics, Inc.]. [Online]. Available: https://www.crestron.com/. [Accessed: 24-Sep-2018].

[8] “NodeMCU Documentation¶,” *Overview - NodeMCU Documentation*. [Online]. Available: https://nodemcu.readthedocs.io/en/master/. [Accessed: 05-Nov-2018].

[8]“Nodemcu Devkit Instruction,” *Github*. [Online]. Available: https://github.com/nodemcu/nodemcu-devkit-v1.0/blob/master/Documents/NODEMCU-DEVKIT-V1.0-INSTRUCTION-EN.pdf

[9] Recom-power.com. (2018). [online] Available at: https://www.recom-power.com/pdf/Powerline-AC-DC/RAC01-GA.pdf [Accessed 14 Dec. 2018].

[10]Datasheets.maximintegrated.com. (2018). [online] Available at: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf [Accessed 14 Dec. 2018].

[11]Electronics, S. (2018). *SEN-13322 SparkFun Electronics | Development Boards, Kits, Programmers | DigiKey*. [online] Digikey.com. Available at: https://www.digikey.com/product-detail/en/sparkfun-electronics/SEN-13322/1568-1360-ND/5764506?utm_adgroup=xGeneral&slid=&gclid=EAIaIQobChMIx_HP5uqn3wIVlYrICh1HaA66EAAYASAAEgIgT_D_BwE [Accessed 14 Dec. 2018].

[12]Murata.com. (2018). [online] Available at: https://www.murata.com/~/media/webrenewal/products/sensor/infrared/datasheet_pir.ashx?la=en [Accessed 14 Dec. 2018].