Wi-Fi Room Control

Muhammed Khan  
Yousuf Khan  
Chi Shing Poon  
Alejandro Valencia  
David Sigala

Electrical Engineering

The University of Texas at Arlington

Arlington, Texas

[muhammedihs.khan@mavs.uta.edu](mailto:muhammed.khan@mavs.uta.edu)

[yousuf.khan@mavs.uta.edu](mailto:yousuf.khan@mavs.uta.edu)

[chishing.poon@mavs.uta.edu](mailto:chishing.poon@mavs.uta.edu)

[alejandro.valencia@mavs.uta.edu](mailto:alejandro.valencia@mavs.uta.edu)

[david.sigala@mavs.uta.edu](mailto:david.sigala@mavs.uta.edu)

***Abstract*—It is possible that every electronic aspect of this world can be controlled from the palm of one’s hands. This is an especially promising idea considering that if applied to a house, or in this case a room, most things can be controlled with the swipe of a finger. The main objective is to create an “internet of things” where each device is connected together and can then be controlled through a smartphone application. This is a very possible scenario that will be demonstrated in this project. The project will be separated into four parts which would be assigned to different team members. The end project would consist of an application to control certain variables in a room (blinds, lights, temperature, etc.), a hub that will send and receive the signals from and to the application and the hardware, and finally a room that would demonstrate the functionality of this project. The project is planned to be finished by the end of October, with the remaining time used to optimize the project, as well as adding extra features.**

***Keywords—****Hub, Humidifier, App, Environment, MSP430, Sensors, Photocell, Android Studio, Control Room, Gantt Chart, HC-05, DHT22, Relays, SDK*

# Introduction

Although at its basis a room is composed of four walls, a floor, and a ceiling, it also has other features. Some of these are things such as blinds, lights, temperature controls, etc. Imagine all these parts individually interconnected through Wi-Fi/Bluetooth from a mobile device. The way the device will communicate between the user application and the Control room is through the microcontroller, MSP430. The MSP430 will be used with a HC-05 bluetooth module to allow it to send and receive signals via bluetooth over the same network. The way it sends and receives signals is through a component called the Hub. The Hub’s main job is to process the signals that are sent from the other hardware or the application, and decode the signals so it can communicate with every component in this project within a timely manner without little signal loss, if any. However, all the hardware required such as the lights, fans, humidifiers, temperature control, and blinds will be connected through a bluetooth module,; making it simpler and more organized to interconnect every components and enable us to find any errors within the circuitry caused by any voltage spikes or any short circuits. The HUB would be how the data is sent and receive within the microcontroller. Initially, the hub was intended to be based out of the IBM Watson cloud. Nevertheless, another app was introduced through an MIT app inventor making it simple to understand.

# Background

The world is becoming more and more automated as technology advances [7]. Menial tasks which used to require hours of human labor are now done automatically by highly accurate machines at faster rates than were previously thought improbable. This technological revolution does not need to be limited to manufacturing factories however, as it is much more commonly available. Computers have been personalized, the internet is in many households, and yet there are still parts of our lives that have not been optimized.

One such scenario is in our homes. Many of us still need to do one of two extremes. In the first case one needs to get up and pull the switch for the ceiling fan, turn on the air conditioner, push the switch for the lights, set the humidifiers to make sure it doesn’t get so dry as to cause a sore throat, and by the time said individual sits down they’ve already forgotten they also needed to close the blinds. It’s too much to handle at once. This is made especially clear on days when environmental temperature impacts indoor temperature and all one desires to do is come home to a comfortable habitat. The other case is that each has their own remote control, which in itself is impractical. Having to switch between multiple different remotes is tedious and undesirable. People already have enough remote controls as is, with an average of four within a single living room [8]. Instead, it would be much easier and more efficient to be able to control all of these devices at once.

Another scenario where this is applicable is in hotels. Some hotels, such as the Hilton, choose to control all temperature aspects directly at the expense of their guests [6]. This is done because giving guests complete control over a room can lead to many different incidents, however guests feel it undermines their autonomy. Both parties can be satisfied if there was an automated system in place to control the room locally and through hotel staff.

# BLock Diagram & Design

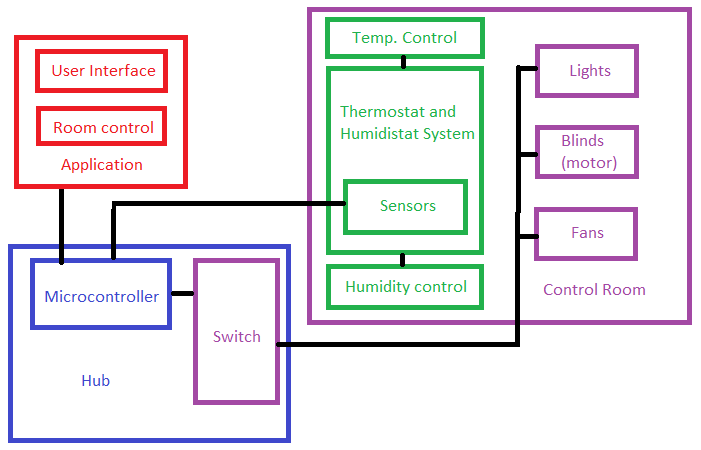
**Figure 1. Block Diagram for Room and Control System**

Figure 1 above shows the block diagram for this project. It illustrates how various components of the project are connected, and what those components are. Since the project is divided into multiple components, the block diagram is color-coded in red, blue, green, and purple for each team member. The team member’s name and their responsibilities can be found after the block diagram. Where it also includes further explanations on the approach of each component. The original block diagram created at the beginning of the semester still applies to this point. In the hub, the switch was changed. However, the diagram will not change because the switch will still be there in the form of a PCB circuit and relay which will be explained in detail further in this report.

The HUB will be hard-wired to the room via simple jumpers and the compatibility circuit mentioned above. This way devices such as lights, blinds and fans can be controlled through the microcontroller. The room will be plugged directly into a wall socket and can be demonstrated anywhere. When plugged in, the room will provide power to all components including the HUB. The thermostat and humidistat system will lie directly in the room and components such as the temperature/humidity sensor will be visible to the viewer. These components will also be hard-wired to the room and hub.

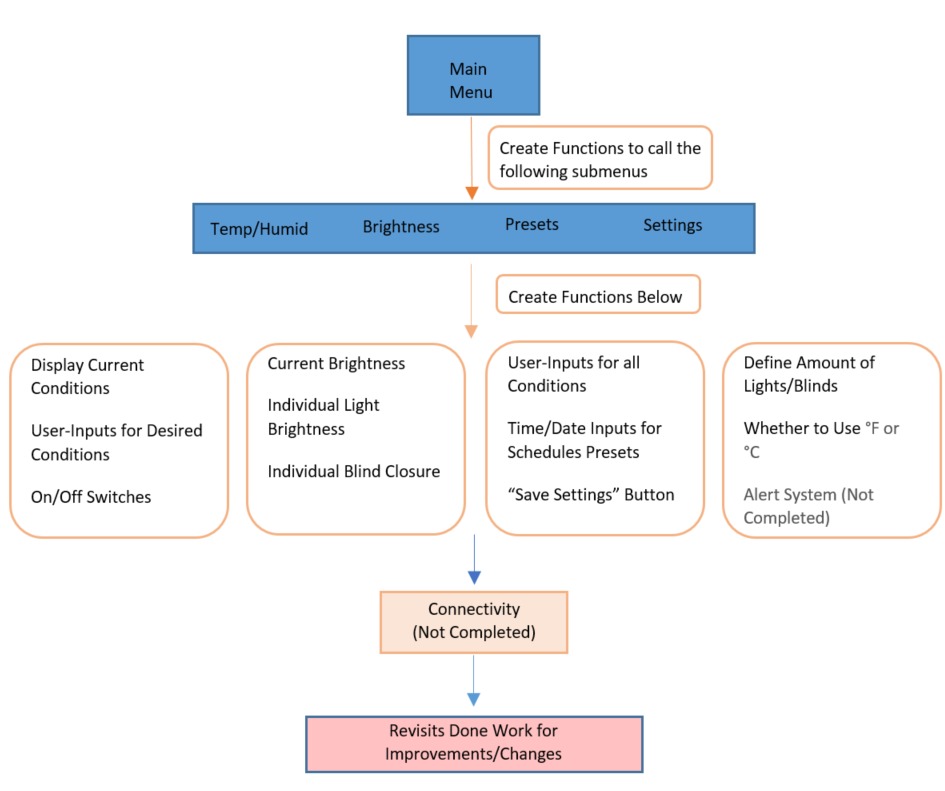
**Table 1. Team Members and Responsibility**

|  |  |  |
| --- | --- | --- |
| **Member’s Name** | **Color** | **Responsibility** |
| Chi Shing Poon | RED | Application |
| Muhammed Khan | BLUE | Hub |
| Yousuf Khan | BLUE | Hub |
| Alejandro Valencia | PURPLE | Control Room and Hardware |
| David Sigala | GREEN | Temperature and Humidity System |

## *Application (Team member: Chi Shing Poon)*

Since this portion was completely software based, there was no circuit or PCB board designs. The app was fully written in Android Studio. The design progress included was five sections, which is illustrated in figure 2.

It began with deciding how many activities there are. Activities are another term for the different screens an Android application has. Then each screen would have a different design layouts; all of them have various input and output attributes. Upon finishing those layouts, functions had to be added to each of those attributes based on their purpose. Connectivity follows, which is via Bluetooth. Finally, we revisit everything to make any changes and improvements.

****

**Figure 3. Application Workflow.**

In our case, the applications have various activities serving different purposes: Main Menu, Temperature and Humidity, Brightness, Presets, and Settings. The Temperature and Humidity Submenu, as well as the Brightness Submenu, both have the same core purpose: To display the current condition in the room and to allow user input their desired conditions, in regards to their submenu topic (temperature, humidity, or brightness).

T/HFIGURE (Figure 4)

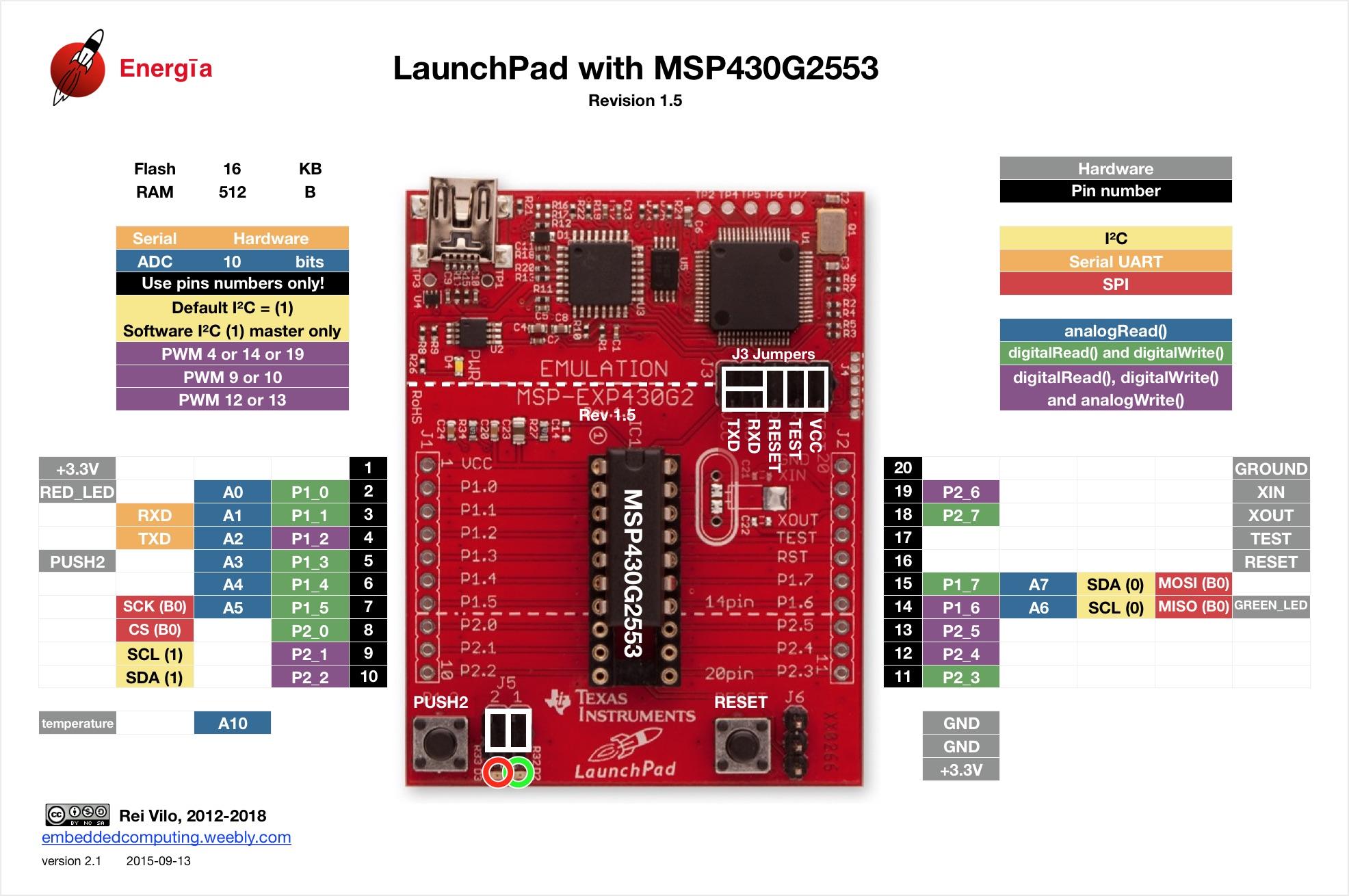
BRIGHT FIGURE (Figure 5)

For the preset submenu, there are two parts of the user-input interface. First part is the top section of the screen, shown in figure 5. This is where the user input their desired conditions for the room, which includes temperature, humidity, and brightness. Preferably, the brightness section would have individual controls over each lights and blinds; but since our group decided to control both lights together and only have one blind, there wi

## *Hub (Muhammed Khan and Yousuf Khan)*

During the beginning of this project we decided to go towards the ethernet/ CAT 5 route however, there was a slight hiccup on that because initially we thought that the CAT 5 was programmable and all we had to do was reprogram the microcontroller inside the CAT 5 cable which is basically an Ethernet type connection but the manufacturer said that it is not programmable. After realizing that we had concluded that CAT 5 and anything related to it will not be used in this project.

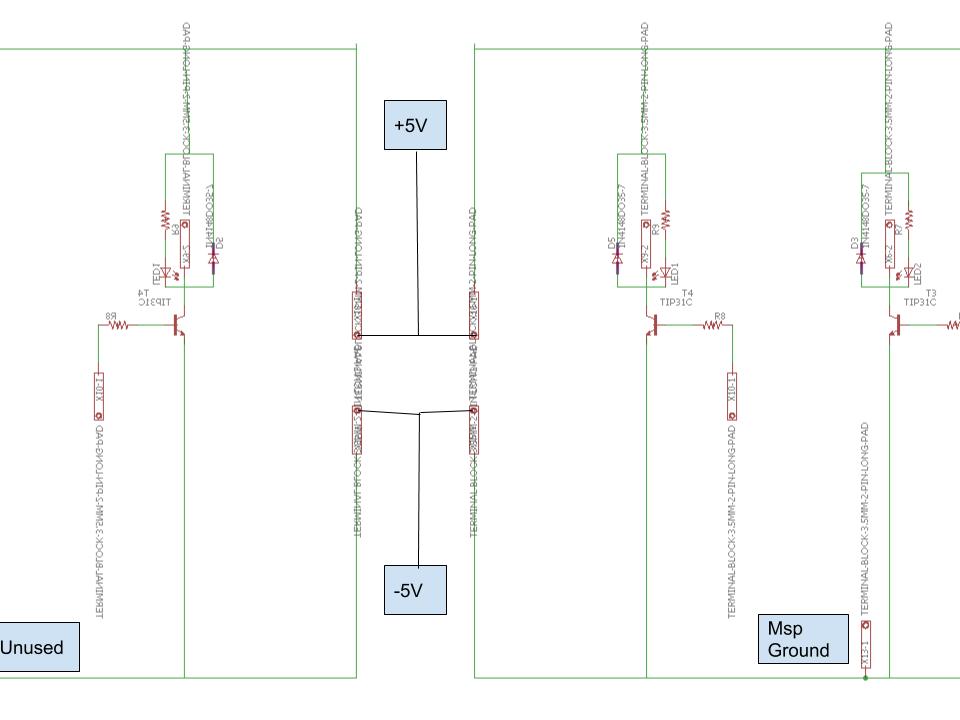
Wifi was then introduced into the project by using the CC3100 along with the MSP430 to try to see if it can be controlled through an active internet connection, this was achieved first by doing some extensive research on how we can get started because we did not have much experience with code and the examples we had come across were mostly arduino based and it did not meet our requirements. However, after some research there was a program called energia that takes the code from arduino and helps translate it into an MSP430 code after manually converting the pin out shown below.



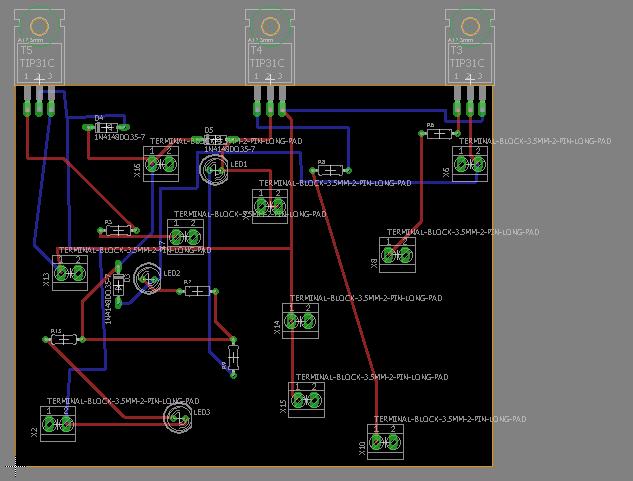
**Figure 6. MSP430 pinout for Energia**

Next the connection between the msp430 and the hub is now ready to go. The hub in this case would be an IBM Watson platform that allowed the device to communicate to the server and this connection was successful after coding the msp430 along with the cc with some code to help it connect. However, the connection over wifi through the CC3100/ MSP430 was successful but the issue was that the Application our other group member was working on did not comply with the IBM watson cloud requirements which made it difficult for us to proceed. After successfully being able to communicate with the IBM watson Bluemix server the device was registering to the cloud at a constant rate of what the temperature in the room is and it constantly send this data over the internet every couple seconds. once again, it was very challenging trying to create an Application to process this data and send it out all in a short amount of time that was given. More details on the application will be provided under the Application section.

After a rough ride through both the CAT5 and the Wifi not properly functioning as anticipated. The group decided to go ahead and go with the Bluetooth route because that did not require as intense of an application as it did if the project continued with wifi. However when doing the bluetooth a proper module was needed instead of the previous module used for wifi known as CC3100 this time a HC-05 module was used to establish a connection through bluetooth and a simple application was used known as serial bluetooth terminal. this application was coded in order to just recieve simple commands and simple streams of data, all components of this part were successful in working using this application. However, this was not the final design for the application as the vision of the application was much more complex. Another key component to remember is that the MSP 430 is only allowed to send max 3.3 volts and the lights we are using require 120v so we ran the mps430 through a NPN BJT transistor and a relay allowing everything to be activated. the schematic was built in eagle and the relay is shown above.



**Figure 7. PCB connections**



**Figure 8. EAGLE connections and routing**

After the schematic was built in Eagle the PCB design was also routed and sent to milling on PCB board. The entire routing was done manually and ensured that there weren't any overlaps. This was by far the most challenging part of the project because it required intense rerouting ensuring no wires were overlapping. After multiple efforts it came out successfully and worked with our project. The next step was ensuring that the temperature sensor and humidity sensor was also configured into the original code of the MSP430 and HC-05 because originally the group decided to work separately on different parts and now we are making sure everything comes together and is properly aligned with everything making sure nothing is out of place. The way this was done by adding another function into the code for both humidity and temperature allowing it to function with the serial link application. The code was then used to make sure that it was compatible with the application using the MIT app inventor.

The hub will take commands from the application through serial data. For example, each command will be assigned to a character (number, letter or symbol) that will trigger an action in the HUB. Furthermore, to keep a level of simplicity brightness, temperature and humidity will also behave in the same way. The user will be given a range of temperature that they can set, for this project it will be limited from 68-78 degrees fahrenheit. Further applications can include a wider range. Brightness will be controlled through PWM and will be arranged in intervals of 50 from 0-255. Lastly, the Humidity will range from 15-50% in intervals of 5. This will allow the user to use only realistic levels of humidity in their control. In more complex applications of this technology, a wider range and more precise levels can be applied for all three of these variables.

The code for temperature and humidity will act very similarly. Depending on the user’s input the variable for desired temperature or humidity will change within the microcontroller. That will be compared with the actual value and whether it is higher or lower, it will trigger the fans and/or humidifier. Though an activation for a heater was included in the code, the final design will not include an activation for the heater.

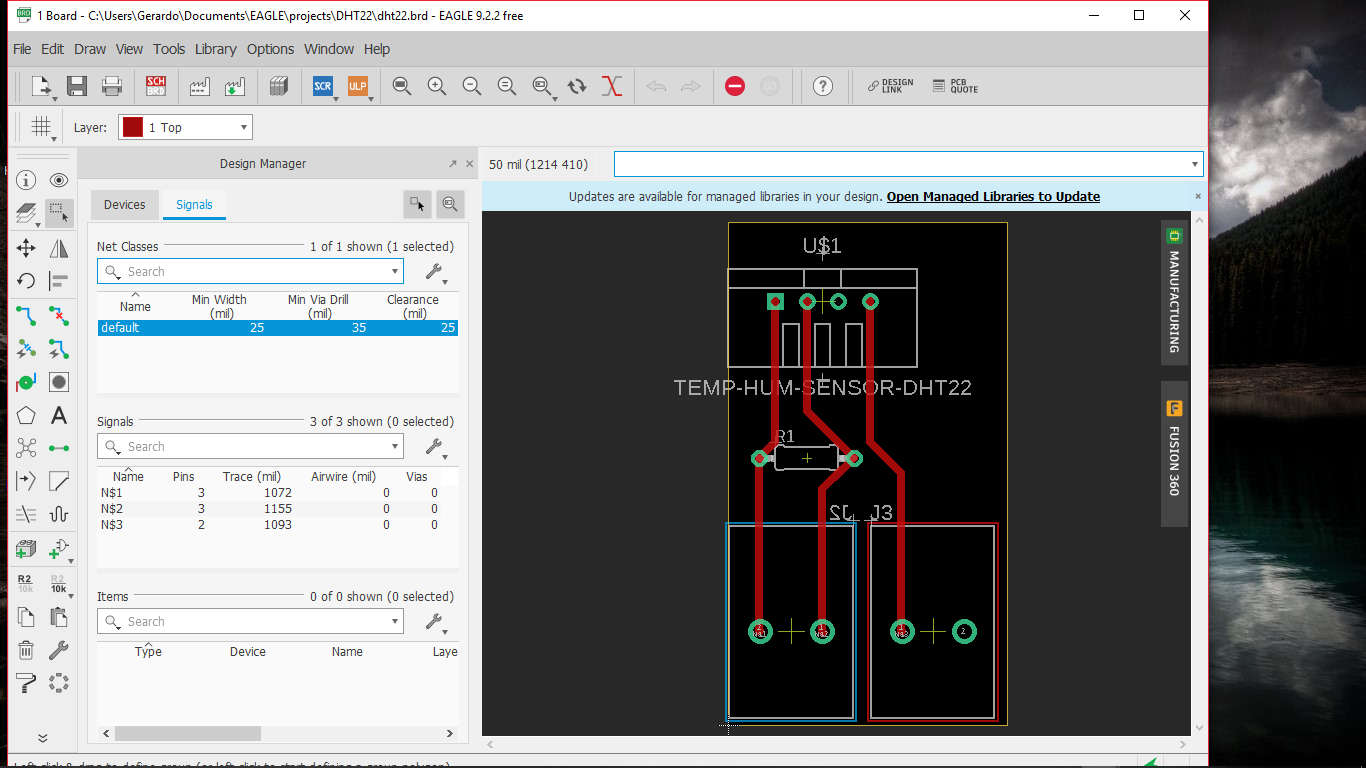
Data is sent back to the application using the “Serial.write” function. This way data about the current state of the room such as brightness temperature and humidity can be visible in the application.

## *Control Room (Hardware) (Alejandro Valencia)*

Procedure, circuit diagram, simulation results, test results from breadboard, PCB layout (if you built any), describe different components, why you chose that particular component and what challenges you faced while designing that part.

## *Temperature and Humidity System (David Sigala)*

For the Temperature and Humidity system we chose ultimately to go with the DHT22 to measure the values, the UCareAir cool mist humidifier to add moisture to the room, and a system of fans for cooling and dehumidifying. The DHT22 is a digital humidity and temperature sensor that uses a Thermistor to gauge temperature, and capacitive elements to detect the humidity. Thermistors are resistors whose resistance changes with temperature. Capacitive humidity sensors usually work by having a bit of metal oxide between electrodes. This metal oxide is sensitive to capacitive changes, so capacitive humidity sensors measure the difference percentage wise for relative humidity. These features make the DHT22 accurate enough. This sensor only uses three of its four pins VCC, ground, and the data input. To ensure accuracy, a pull up resistor of 4.7k-10k ohms is used between the VCC and data input pin.



**Figure 9. EAGLE connections and routing for sensor**

In the end we chose to go without the LCD screen. Although we had it correctly configured to display the temperature and humidity, due to pin limits on the MSP430 and its admittedly redundant nature with respect to the app, we chose to scrap it.

We used the UCareAir cool mist humidifier for humidifying the room. However, the activation of the device via button proved beyond our level of expertise. Instead we opted to to approach this from a different angle. Instead of turning the device on and off, we could leave it on the whole time and instead control its flow. While a solenoid valve would have been ideal, the pressure and current requirements were higher than even our relays would be capable of reaching without burning out. We opted to control the flow of the mist with a small 1” fan. This fan would be always on to prevent the always on humidifier from humidifying the room. When the room needs humidity the fan will turn off. To make sure the mist wasn’t shot out stronger than the fan, we sent it through a few inches of 1 ⅛ “ PVC pipe with two openings. One leads to outside of the room, and the other inside.

For cooling the room we used a system of relays to power on fans. The plan is to have these fans turn on when the temperature detected is greater than the desired temperature. Likewise, when the Humidity is less than the desired, the pipe fan will turn off. When the Humidity is greater than what the desired humidity is, the exhaust fan (modeled similarly to the cooling fan) will activate pushing the moist air out of the room to be replaced by dryer air. We had a plan to make a heating element, but because of fire hazards and massive power consumption we had to scrap it.

# Decision on Solution and PArt Selection

**Table 2. Part Selection for Hub and Application**

|  |  |  |
| --- | --- | --- |
| **Item** | **Price ($)** | **Description** |
| **MSP430G2559** | **0** | **Microcontroller** |
| **HC-05** | **0** | **Bluetooth Module** |
| **USB-C** | **0** | **Connector Cable** |
| **Google Pixel** | **0** | **Application Development** |
| **Google PIxel 2** | **0** | **Application Development** |
| **Jumper Cables** | **0** | **Connectivity** |
| **8- Channel Relay switch** |  | **Control** |
| **NPN BJT Transistors** | **0** | **Control** |
| **Resistors** | **0** | **Control** |
| **Diodes** | **0** | **Control** |
| **LED’s** | **0** | **Control** |

One of the main reasons the MSP430G2 was chosen because of its compatibility with the selected bluetooth module. The bluetooth module being mentioned is the HC-05. This device allows for easy connectivity and pairing with all bluetooth enabled devices. Simple jumper cables were used connect the microcontroller to devices such as the relay. Android application development was chosen because of its friendliness to developers versus iphone development. The application will be run on both the Google Pixel and Pixel 2 because they are the ideal devices for Android applications and they run the most pure, stock version of Android. These devices use USB-C cables to load the applications onto them so that was also needed. All of these components for these sections besides the relays are already owned by members of this project or were retrieved from the university so the net cost for this portion would be the price of the relay.

The relays that were used can only be controlled by a minimum of 5v. Our microcontroller only has the capacity to output a maximum of 3.3V at the high setting. To control the relays, a circuit that uses an NPN BJT transistor with a 5v source tied to it was used. The 3.3V source from the msp430 would control the transistor which, when on, would allow the 5v to activate the relays. This circuit included an NPN BJT transistor, resistors, LED’s, diodes and a 5V source. The schematic in EAGLE before being milled on a PCB board can be found in the HUB section of this report.

**Table 3. Components for the Control Room**

|  |  |  |
| --- | --- | --- |
| **Material type** | **Price ($)** | **Description** |
| **Scaffold** | **0** | **Skeleton** |
| **4x8 sheetrock** | **0** | **Skin** |
| **Joint Compound** | **0** | **Plaster** |
| **White paint** | **0** | **Cosmetics** |
| **12/2 Romex** | **0** | **Power** |
| **18/2 Low voltage wire** | **0** | **Dimming** |
| **Screen Mesh** | **0** | **Blinds** |
| **DC Motor** | **2** | **Motor** |
| **2 USB Desk fans** | **14** | **Fans/Exhaust** |
| **2 computer fans** | **20** |  |
| **2 pole/Dimming switch** | **0** | **Switch** |
| **Can Lights** | **0** | **Lights** |
| **Impact screw gun** | **0** | **Tools** |
| **LCD screen** | **11** | **Thermostat/**  **Humidistat System** |
| **DHT22 Sensor** | **10** | **Thermostat/**  **Humidistat System** |
| **Cool mist humidifier** | **0** | **Thermostat/**  **Humidistat System** |

`Originally we were planning to use the HDC2010YPAR temperature and humidity sensor, however as can be seen in the figure, the size proved to be much more difficult to work with than we were intending.

(insert picture side by side)

This led us as a group to decide on choosing a more user friendly sensor, the DHT22.The use of a thermistor and capacitive humidity sensing makes this sensor accurate enough.Another reason this sensor was chosen was for the relative abundance of sample programs. Since the sensor takes in small analog inputs and performs digital conversions with a small chip inside of it. By getting the correct libraries, it is easy to read the temperature and humidity values. For humidity, we chose to go with the UCareAir Cool Mist Humidifier. Normally the cool mist can launch mist on average to 23”, which if the sales pitch is to be believe is better than the average humidifier. The reason we chose a 1” fan was because we planed to use pipe to slow down the flow of mist and work off of our already working fan controls. The pipe was picked because of its approximate size to the mist output hole of the UCareAir cool mist humidifier.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

.

# Conclusion

Despite all the decisions and the pivots made in this project. Students learned a tremendous amount of knowledge. The HUB that was stated earlier went through tremendous amount of research and exploration allowing the Project to get to where it's at today. Some of the things that can be taken away from the HUB would be the way students learned to communicate between the different types of devices simultaneously through a Web server whether that being Amazon Web Services or IBM bluemix server which this project is primarily focused on. Unfortunately, some of the things did not go as desired as it was thought out, one being that connectivity could not be established between one of the devices, causing the primary focus of this Project to divert towards Bluetooth connectivity. However, given more time it could have been possible for it to work.

Bluetooth was made easier in terms of connectivity. However, the application also had to be changed in order for it to meet the standards of bluetooth. Bluetooth is fast and simple to set up, so simple that it requires the user just to have a device with bluetooth connection, no paid internet fees, no bandwidth issues. This creates more security and more reasonable expenses in regards to the user.

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**APPENDIX**

Insert Gantt chart, components list, and full-sized diagrams which cannot be seen clearly in the paper.

Submit the code files separately. Don’t include code unless you want to explain it.

s