Smart Curtain System

Final report

Team 3N3

Arthur Chen

David Erdner

Caleb Key

Pierre Vu

James Streets

Department of Computer Science
Texas A&M University

Date 05/03/2022

Table of Contents

Constraints and Engineering Standards Index
For Smart Curtain System

Project Title: _____Smart Curtain System – Team 3N3____________
Team Members: ____David Erdner, James Streets, Arthur Chen, Caleb Key, Pierre Vu____

CONSTRAINT	PAGE	PAGE	PAGE	PAGE
Economic	45			
Environmental	44			
Social	44			
Political	45			
Ethical	43			
Health and Safety	43			
Manufacturability Constructability	45			

1 Executive summary

Smart home technology has been booming more than ever over the last decade. The total worldwide market for Internet of Things (IoT) devices is projected to be over $75 Billion by 2025. With the Smart Curtain System, we aim to create a practical device that is the next step in home automation. The hope is that this system will be implemented on a large scale across the market in an attempt to make people's lives better, but also to save energy. Because our product brings these two important ideas together, we believe that the Smart Curtain System is set up for success. In the home automation industry, the Smart Curtain system will provide a simple solution to an often ignored problem.

Needs - The sun brings a lot of energy into a room through windows directly exposed to sunlight. Without such a system, curtains must be controlled manually, requiring time and effort that would otherwise be avoided with an automated system. This project will implement such a system, offering users simple and intuitive control over their home that would be managed and scheduled internally. Completing this project will be achieved through automatic control over the curtains combined with an effective measurement system. The success of the product will be determined by the amount of energy saved through blocking sunlight from entering the covered window.

Final Design Implementation - The system design will be covered in much detail later on in this report, but to summarize, we use a combination of Arduino input components, Android application software, and a SwitchBot Curtain motor to bring our project to life. These components are connected to one another through bluetooth/Wifi, and makes the system easy for the user to install, setup, and operate the Smart Curtain System with no issues.

Results Obtained - Once the final system was up and running and had been tested for reliability, the results of the project specifically in terms of energy saved and user home improvement were numerous. Not only does our system automatically open and close the curtains based on what the desired temperature is indoors to save energy, but it allows the user to easily control the curtains manually. Thus, our project's results are twofold. First, the curtains save the home electricity when in automatic mode because of the effect that opening/closing curtains has on heating and cooling a home. Second, the ability to control the curtains manually with just the press of a button results in true home improvement, and is the next step in home automation.

Experiences with Team/Project Management - Ever since the first day of the project once the teams were assigned, our team has operated with a high level of excellence. Meeting multiple times in person throughout the week, as well as constant communication via Discord and GroupMe has set the team up for success. One thing that has really helped the team is the intercommunication between roles. Even though each team member is technically responsible for at least one large aspect of the project, there has been thorough communication between role leaders that has helped the project become successful. Our project management has been properly organized through Gantt and flow charts, and we have done an excellent job of staying true to our project commitments.

2 Project background

In the home automation industry, the Smart Curtain system will provide a simple solution to an often ignored problem. The sun brings a lot of energy into a room through windows directly exposed to sunlight. Without such a system, curtains must be controlled manually, requiring time and effort that would otherwise be avoided with an automated system. This project will implement such a system, offering users simple and intuitive control over their home that would be managed and scheduled internally. Completing this project will be achieved through automatic control over the curtains combined with an effective measurement system. The success of the product will be determined by the amount of energy saved through blocking sunlight from entering the covered window, or by how much energy is saved by the sun heating the room instead of the house electricity used.

2.1 Needs statement

Without blinds or curtains, the sun can heat a room through a window substantially after prolonged exposure. Covering the window with curtains will block the energy from the sun and prevent a majority of the energy from heating the room. While this can be managed by manually opening and closing blinds, the amount of time that it would take to do so could stack up for many people as they go throughout their daily routine. When leaving the room or the building, users are unable to manually control the curtains. A system that could manage the curtains without the need for manual control would allow users to freely go about their typical routine without needing to worry about managing the curtains.

2.2 Goal and objectives

The goal of our system is to create an environment in which the energy from the sunlight would always be controlled to the desires of the user. If the user would like a cool room, the system would automatically block out sun from the room by closing the curtains, and vice versa. Our proposal is to solve this problem using an automatic system that will schedule the opening and closing of curtains in a room. If done properly, the room will be heated and or cooled by blocking or allowing ambient sunlight to enter the room. While this would not be able to heat or cool the room past the possible temperature allowed by the sunlight, it would push the room temperature in the correct direction or, if possible, to the exact desired temperature.

2.3 Design constraints and feasibility

Throughout the development of this project, our design constraints were followed closely. Our final product is low profile and safe, which makes it more attractive to a customer. The designed system will need to either be low profile or have an aesthetic design. The system must be safe, including a secure mounting, contained electronics, and substantial warnings to users. The system could fall without proper mounting. Exposed electronics could potentially be harmful to users. Finally, consider the curtains catching on an object while opening or closing; users should be warned to keep the area clear.

2.4 Literature and technical survey

This project's goal is to build a curtain system that will save energy by determining the need for more or less radiation from the sun through the window that the curtain will be setup at. Based on this goal, we needed to begin our research with finding the radiation from a common exterior window. Once we found this information, we then needed to discover what material(s) would best block that radiation. We were able to find a material blocking out most of the radiation from the window which would help us reach a near binary state of the curtain system since ideally, we could block out all the sun, none of it, and everything in between. These two pieces of information were a catalyst for our design process and along with that, we deemed it necessary to be concerned with similar products currently available on the market. We found 3 various automatic curtain systems that had various levels of influence on our design process.

Mengting Zhu and others from the College of Civil Engineering and Architecture at Wenzhou University in Zhejiang, China performed a research on the influence of key parameters of exterior windows on energy saving. During this research they used seven different types of glass, conducting thermal analysis on each of these through a simulation software. The environment parameters for this study were daily average temperatures through 4 months, varying in their time of year. These months included January, April, July, and October with January and July simulations being the two extremes of the experiment. For our project, our only concern were the common exterior windows which the research concluded that “When the glass is ordinary glass, the incident rate of solar radiation is 100%...In this case, 15% of the total solar radiant heat is isolated and 85% is transmitted indoors.” [1]. This means that we will be concerned with 85% of the total heat radiation from the sun. Knowing this now, we must find a material that will achieve close to a 100% blockout rate.

A research conducted by Joanna Szkudlarek and others from Lodz University of Technology analyzed the light intensity that broke through 3 different materials, particularly their fabric structure. Fabric 1 was a smooth, uniform material, fabric 2 being a woven material with a higher linear density, and fabric 3 was a smooth material with the lowest area density. This research concluded that fabric 2 had a permeability rate of 0.095%, or a black-out rate of 99.905%, which was roughly 0.025% better than fabric 1 and 1.185% than fabric 3 [2]. We used this information to select a curtain with a similar material structure to that of fabric 2 of this study. This would allow us to achieve a similar blackout rate and thus preserve more energy when attempting to block out all of the heat radiation from the sun.

Our first product review was a shading system developed by Graywind that has much potential with their use of blackout curtains and has the convenience factor with the ability to be controlled through home speaker devices such as Alexa, and Google Home. This product does not advertise energy-saving capabilities or user-less control. The blackout curtains as stated above would be a great energy-saving feature and thus would have decided to take this route in our design.

Our second product review, SimpleSmart's curtain track, has many capabilities as it can be controlled through a remote control, through an Iphone/Android application, or through home speaker devices such as the ones mentioned above. This is a very simple track system with only user-mode controls and no energy-saving features. Our main takeaway from this item is it's diversity of control platforms through the use of Alexa, an application, and an external remote.

Our final product, developed by SmartBot, is an electric motor that glides across a curtain rod system. This product can be controlled by either a remote control for the device or through their iphone/android application. This design seems simplistic and easy to use which is why our design will use this product to move the curtain. This can be shown in the design section of this proposal.

This research review has shown us that our product is not only distinct but also very practical. Through the window parameter research by Zhu, we saw that common exterior windows have a very poor energy-saving performance which opens the door to desirability of our product. Then the research conducted by Szkudlarek gave us an insight into the blackout curtain material desired for the lowest permeability of light which is highly beneficial to reaching our goal, saving energy for the user through the use of a curtain system. The product reviews showed that there are developments in this area that have potential but lack the creativity and ingenuity it takes to have energy-saving features. Through all this, we concluded that our project is distinct and that there is a market for our product, waiting for a group to take over the landscape through the use of automated energy-saving features.

Existing product or projects:

- Proposal of Objective Assessment of the Phenomenon of Light Passage through Blackout Fabrics
- Study on the influence of key parameters of exterior window structure on building energy saving effect
- Graywind Motorized Shades
- SimpleSmart - Motorized Curtain Tracks
- SwitchBot Curtain Smart Electric Motor

2.5 Evaluation of alternative solutions

There are multiple alternative solutions existing in the related field of study. Based on the literature and the technical survey in the previous section, the alternative solutions will be evaluated by comparing their pros and cons. Lastly, after the evaluation of all the alternatives, we are proposing a particular solution which will be detailed in the following section.

There are the alternative solutions:

- Proposal of Objective Assessment of the Phenomenon of Light Passage through Blackout Fabrics

  This research paper is an existing approach to discover the relationships between the transparency of light through the wild range of the materials. This paper helps us to decide which type of fabric we should use in our design.

- Study on the influence of key parameters of exterior window structure on building energy saving effect

  This research paper found the correlation between the temperature and the matrix equation between the energy saving effect through gray theory and regression analysis, which helps us to understand why energy could be saved through proper control of the curtain.
- Graywind Motorized Shades

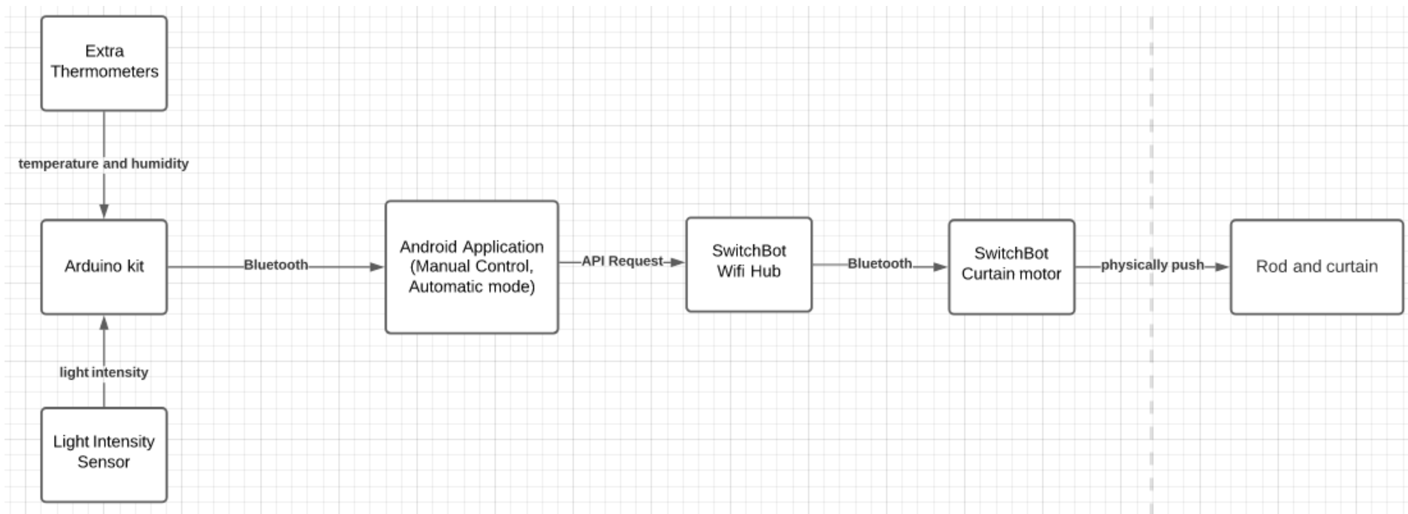
  This particular shape of curtain is widely used commercially due to the fact that the spaces preserved within the holes in the shades reduce the heat exchange between indoor and outdoor.
- SimpleSmart - Motorized Curtain Tracks

  SimpleSmart is an existing product that uses an autonomous system to control the curtain remotely and automatically, which adjusts curtain positions to save energy when residents are not around while keeping the desired temperature.
- SwitchBot Curtain Smart Electric Motor

  This electric motor is similar to the previous one mentioned above, where it is widely used as well. The product uses an alternative approach to control the curtain, by using a motor to physically push the curtain.

3 Final design

3.1 System description



Light Intensity Sensor

The light intensity sensor will be used to measure the intensity of the light from the Sun outside the window. This sensor will provide this information digitally, so that we can use it in our code to program the motor.

Thermometers

The thermometers will be used to measure the temperature of the room and of the outside. These thermometers will be digital, so that we can use the information provided in our code to program the motor.

Microcontroller kit

We will be using a microcontroller to program our motor. Between a microcontroller and a microprocessor, the microcontroller is more commonly used for simple motor functions, so we thought that it would be easier to use for a smart curtain. The microcontroller will read in the information from the light sensor and the thermometers, and we will use that information in our program to tell the motor whether to open or close the curtains.

UI Input

This will be our way of communicating with the microcontroller the settings that can be changed. We will use a bluetooth module to send the preferred settings to the system and then to the curtain. The UI input will be done via an Android application. This will receive a bluetooth signal from the microcontroller.

Bluetooth modules

These modules will be used for communication between the microcontroller and the Android application. This is changed from our previous design as we have included the SwitchBot cloud hub in order to connect to the SwitchBot. This means we only use bluetooth for one connection instead of both.

SwitchBot Cloud Hub

The cloud hub allows for us to connect to the SwitchBot curtain motor easily via an API request. This is different from our previous implementation as it is easier to use their API than make a whole new bluetooth connection for their motor.

SwitchBot Curtain Motor

This is a motor specifically designed for moving curtains, so we are using it in our project. This motor already has a connection with its remote, so we do not need any additional configuration besides using the remote, or bluetooth signal from the remote to control the curtains.

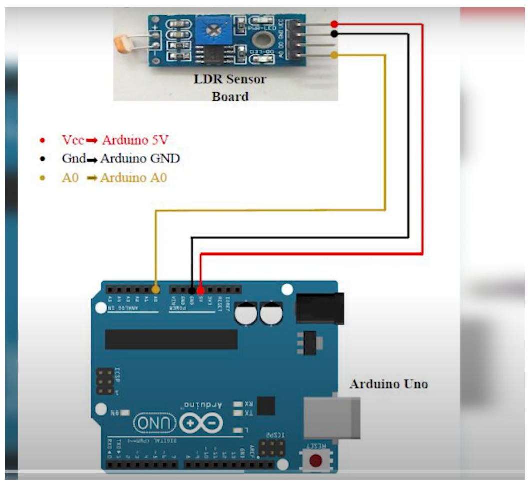
Rod and Curtain

This is a standard curtain rod that is used for holding curtains. It needs to be able to move freely, so that the motor can work. The curtain is going to be a blackout curtain because that will be the most effective in blocking out sunlight and maintaining temperature inside. Any amount of translucency will make it significantly worse in stopping the sun’s energy.

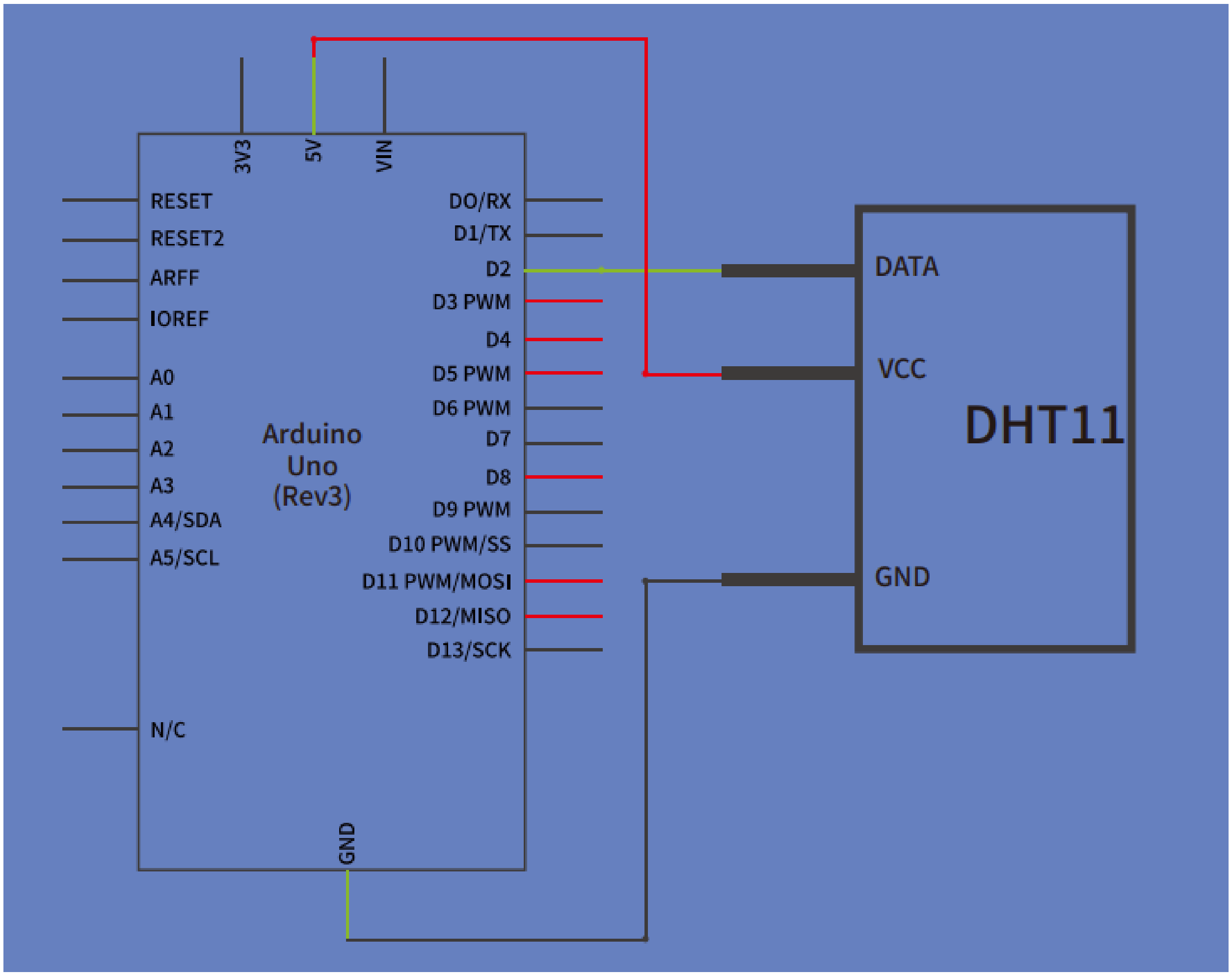
3.2 Complete module-wise specifications

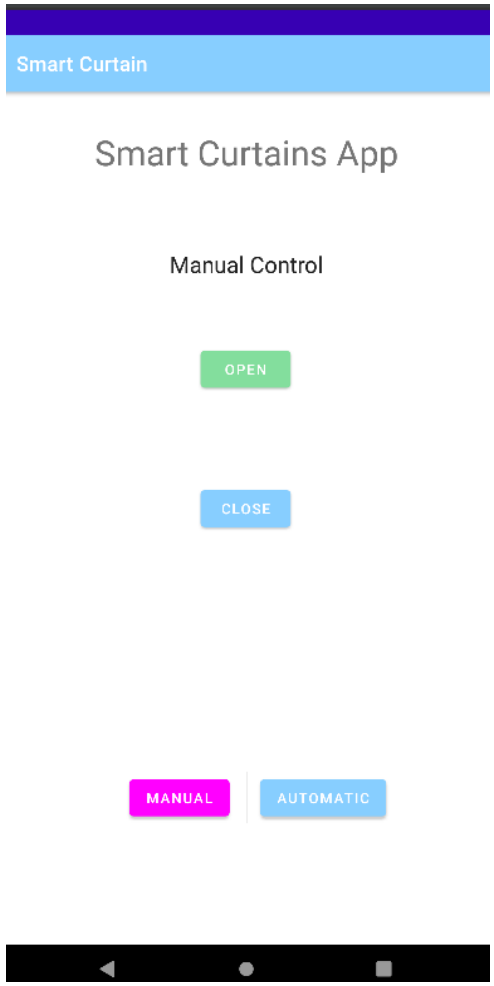
DHT 11 Temp/Humidity Sensor Connection

The above figure shows us the connection between the DHT 11 Temp Sensor and the microcontroller. We have VCC connected to 5V and the temp sensor ground to the microcontroller ground. The temp data bus will be connected to the digital pin 2 on the microcontroller which will be the input for the sensor information.



The above figure shows the light sensor pin configuration, with a simple power/ground input layout. The demonstration later in the report will show the effectiveness of the sensor.


3V3
5V
VIN
RESET
RESET2
ARFF
IOREF
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
Arduino
Uno
(Rev3)
DO/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
GND
DATA
VCC
DHT11
GND



Manual Control Interface Displaying to User Open/Close

These figures display our current app prototype. As of now, we have a manual page and automatic page for our users. In the manual page, the user can set the curtains to be open or closed. Once set, the app will display to the user which preference is currently set through the change of color. We decided that the green would contrast the white background and the other blue buttons schemes as well as being more intuitive as we typically think of green as an “active” color. The automatic page currently blocks the open and close buttons out so that the user cannot activate them and will have an input for them to tell the app what their thermostat is currently set at and our algorithm will then provide energy efficiency for the user by trying to help reach the desired thermostat temperature.

Component	Price	Link
Microcontroller kit	$38.99	microcontroller
SwitchBot Curtain motor	$99.00	SwitchBot motor
Curtain Remote	$16.15	Curtain remote
Extra Thermometers	$10.29	thermometers
Light Intensity Sensor	$10.99	Light intensity sensor
Bluetooth Module for Microcontroller(2)	$18.00	Bluetooth module for microcontroller
Rod for holding curtains	$19.00	Rod for curtains
Black Out curtains	$16.00	Black out curtains
Switchbot Minihub	$39.00	Switchbot Mini Hub
Total	$267.42	

Parts List

We have added the SwitchBot cloud hub to our parts list. This lets us connect to the SwitchBot API and send API requests to the SwitchBot motor instead of trying to connect via bluetooth. After receiving the components, we took inventory and confirmed all components that had arrived and tested their functionality. The curtain remote is the only component that did not arrive in working order. We believe the cell battery has drained in shipping, and will need to get a new battery to test it once more. We have not identified any components that we are lacking that would prevent the completion of all features for our system.

3.3 Approach for design validation

The procedures that we will use to test the system as a whole will be separated into 2 parts. The first part will be testing the system in manual control mode. This mode is the simpler mode of the 2, because the only input needed from the user is if the curtains should be opened or closed. Here is the procedure for the manual mode testing.

Step 1: Open the Android application

Step 2: Switch the mode to “Manual”

Step 3: Press the “Open” button on the application

- If curtains are open, nothing should happen.
- If curtains are closed, the Android device should send a bluetooth signal to the curtain motor to immediately open the curtains.

Step 4: Press the “Close” button on the application

- If curtains are closed, nothing should happen.
- If curtains are open, the Android device should send a bluetooth signal to the curtain motor to immediately close the curtains.

Step 5: Repeat steps 3 and 4 to ensure reliability

Step 6: If any of the previous steps fail to work for any reason, the system testing failed. Otherwise, it is considered a pass for manual control mode.

Now that the system testing is complete for manual mode, the procedure for automatic mode is a little more complicated because it involves more inputs. The procedure for testing the system in automatic mode is below.

Step 1: Open the Android Application

Step 2 Switch the mode to "Automatic"

Step 3: Set the desired temperature

Step 4: Monitor the room temperature and compare with desired temperature and compare the two

> High light intensity - If the desired temperature is less than the room temperature, the curtains should close, lowering room temperature.
>
> High light intensity - If the desired temperature is greater than the room temperature, the curtains should open, raising room temperature.
>
> Low light intensity - The curtains should default to closed curtains as the low light intensity will do little to affect room temperature.

Step 5: Change the desired temperature at different light intensities, to make sure that all cases are considered.

Step 6: If any of the previous steps fail to work for any reason, the system testing failed. Otherwise, it is considered a pass for automatic control mode.

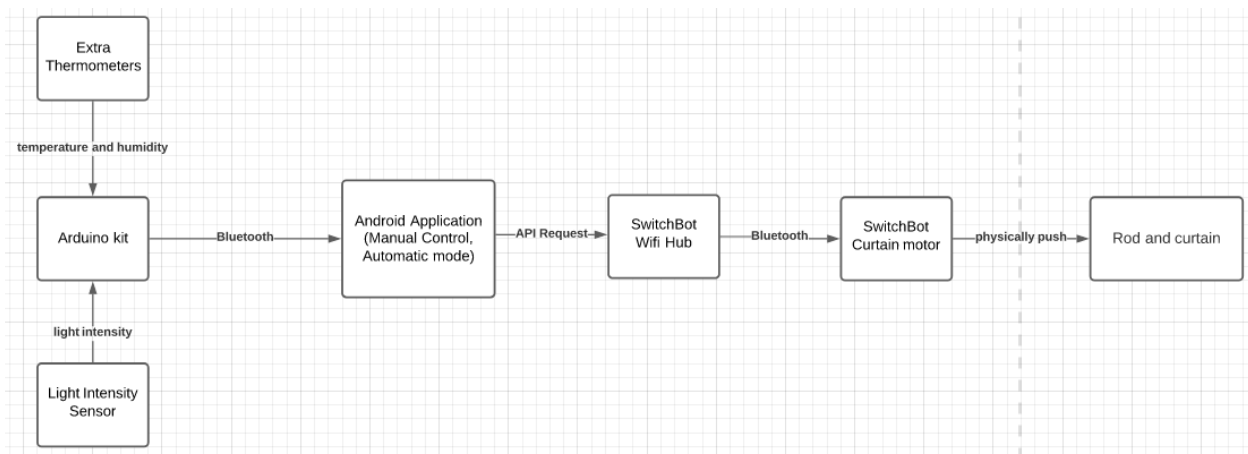
If all of these 2 main system tests pass, then the system will be considered fully functional. This will likely be the way that this project will be demonstrated as proof of a working system.

4 Implementation notes (10-20 pages; 30 points)

This section contains the implementation notes for this project. It includes

1. Recap on high level system design
2. List each hardware component and entail the configuration and technical specifications for each of them

To recap on the high level diagram,



The main inputs to the arduino are the thermometers and the light intensity sensors. These inputs are fed into the arduino, which then sends them to the android application when in automation mode. The user sets either manual mode or automatic mode, and then the app sends an API request to the switchbot wifi hub, which then sends a bluetooth open/close message to the switchbot curtain motor. The motor then pushes the curtain along the rod to open or close the curtains.

Hardware Components

This section shows off the hardware components used for the design, below is a picture of how the temperature sensor is wired. Our project actually contains two sensors, one for the window, and one for the room.

Temperature sensor

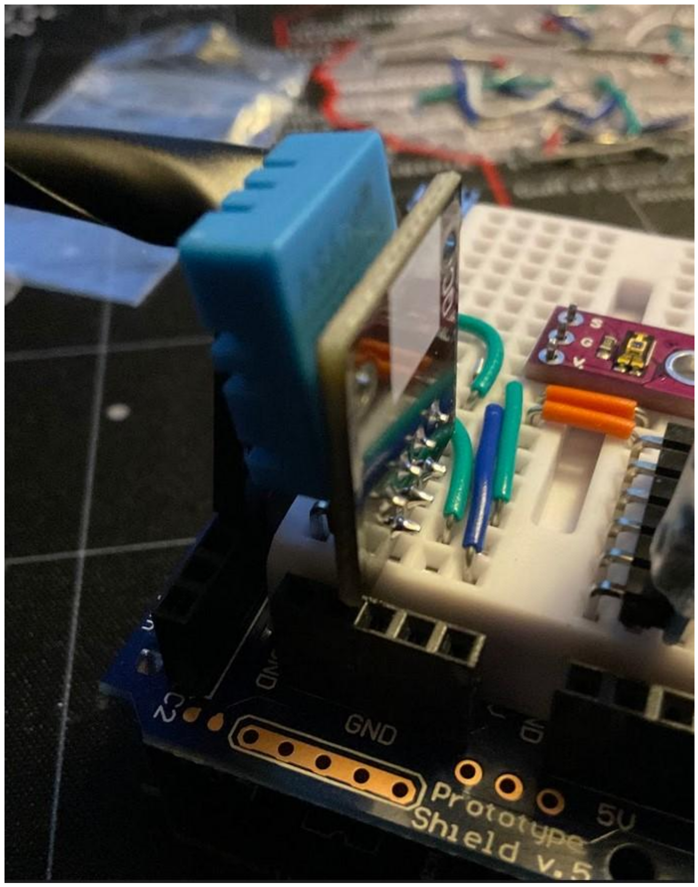


Figure 6.1) Temperature Sensor

The pin connections are as follows. Vcc should be connected to 5v power. The output reads in ℃. If the connections are loose, then the reading will be 0℃. This needs to be converted to ℉ for our calculations.

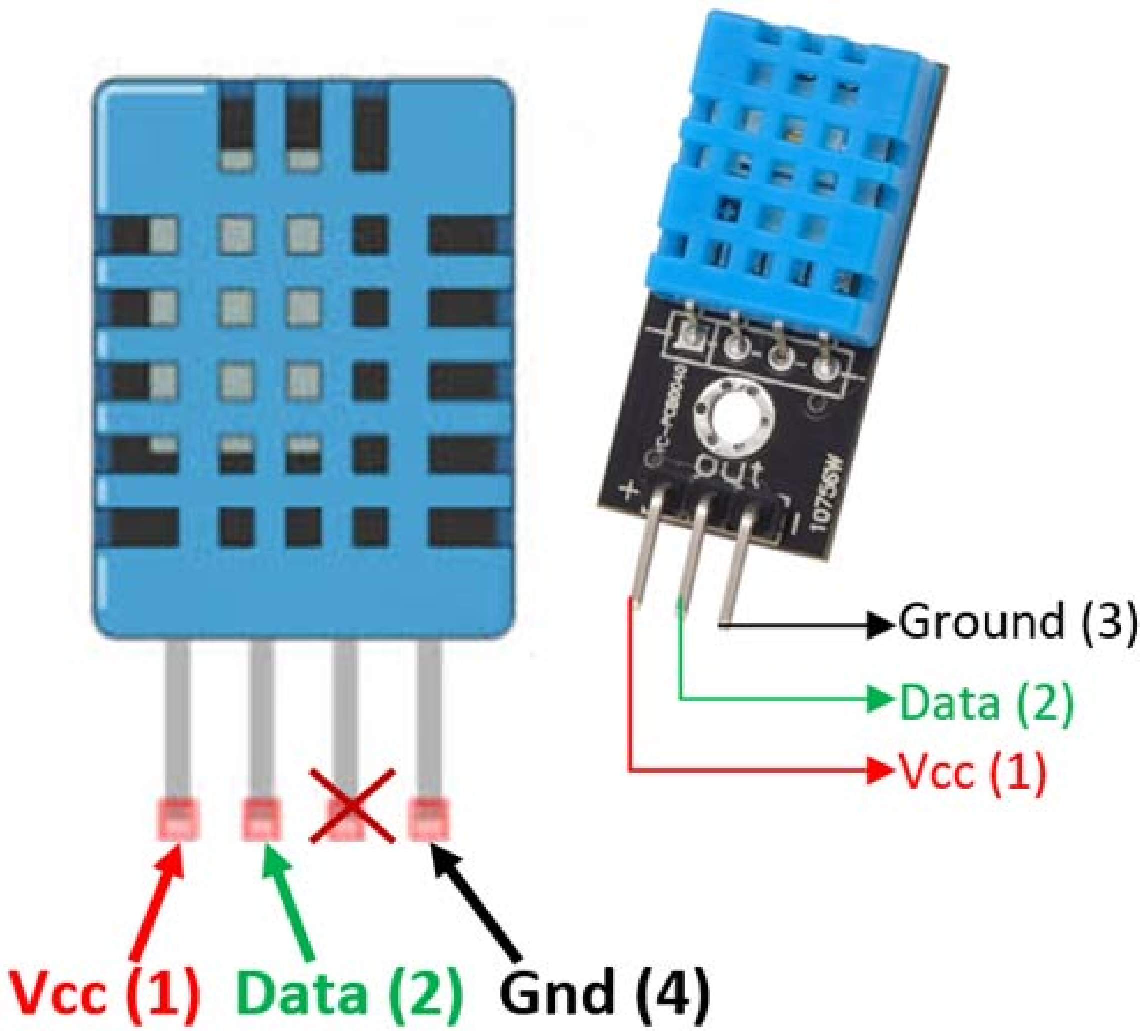


Figure 6.2) Temperature Sensor Diagram

Below is a screenshot of the microcontroller that we used for the project. This is the main component used for the connection between parts, and also allows us to run code to send temperature and light intensity readings to the android application.

Figure 6.3) Arduino Microcontroller

The pin configurations are as follows

Microcontroller		Component/Pin
Vin	->	HC-05 / VCC
GND	->	HC-05 / GND DHT11 (1) / GND DHT11 (2) / GND
A0	->	DHT11 (1) / Out
A1	->	DHT11 (2) / Out
Digital Pin 3	->	HC-05 / RXD
Digital Pin 2	->	HC-05 / TXD

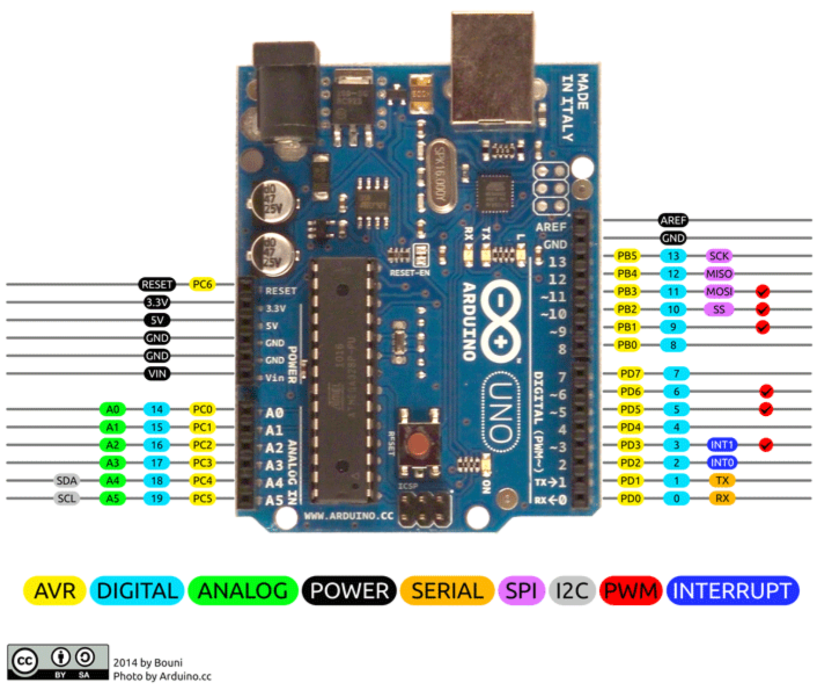


Figure 6.4) Arduino Pin Configuration

Software Component

Microcontroller Code

bt_temp_light

```
1 #include <dht11.h>
2 #include <SoftwareSerial.h>
3
4 #define dht_window_apin A1
5 #define dht_curr_apin A0
6
7 const int RX_PIN = 2;
8 const int TX_PIN = 3;
9 SoftwareSerial serial(RX_PIN, TX_PIN);
10 char commandChar;
11 int state;
12 dht11 DHTwindow;
13 dht11 DHTcurr;
14
```

Figure 6.5) Arduino Code A

Lines 1-2 are the library for reading the temperature from the dht sensors and the library to send/receive data through the bluetooth module, respectively.

Lines 4-5 are macros linking our pins to corresponding variable rates. These define which output pins correspond to which temperature sensor. The DHTwindow is the temperature sensor inside the window, and similarly for DHTcurr, which is the temperature sensor inside.

Lines 7-8 are constant integer values representing the receive/transmission pins on the microcontroller.

Line 9 instantiates a SoftwareSerial object called "serial" and passes in the receive/transmission pins as constructor parameters.

Lines 10-11 declare a character to assign to the received data from the android application and a state value used to determine if the bluetooth module should or should not send the temperature. Lines 12-13 are instantiation of dht11 sensor objects which will be used to read in the data later.

bt_temp_light

```
27 void loop( )
28 {
29   /* Measure temperature and humidity.  If the functions returns
30      true, then a measurement is available. */
31   if (state)
32   {
33     DHTwindow.read(dht_window_apin);
34     serial.print(DHTwindow.temperature * 9/5 + 32);
35     serial.print("W");
36     DHTcurr.read(dht_curr_apin);
37     serial.print(DHTcurr.temperature * 9/5 + 32);
38     serial.print("C#");
39 
40     delay(5000);
41   }
42 
43   if (serial.available())
44   {
45     commandChar = serial.read();
46     switch(commandChar)
47     {
48       case '*':
49         DHTwindow.read(dht_window_apin);
50         serial.print(DHTwindow.temperature * 9/5 + 32);
51         serial.print("W");
52         DHTwindow.read(dht_curr_apin);
53         serial.print(DHTcurr.temperature * 9/5 + 32);
54         serial.print("C#");
55         break;
56       case '0':
57         state = 0;
58         break;
59       case '1':
60         state = 1;
61         break;
62     }
63   }
64 }
```

Figure 6.6) Arduino Code B

The loop functionality is as follows, the bluetooth module listens for data from the app in lines 43-63. If it receives an asterisk then it will attempt to send both the window and current room temperature to the app.

If it is a zero then it sets the state to 0, and if it is 1 then lines 31-41 will start being executed which is sending the temperature data to the android app every five seconds.

Android App

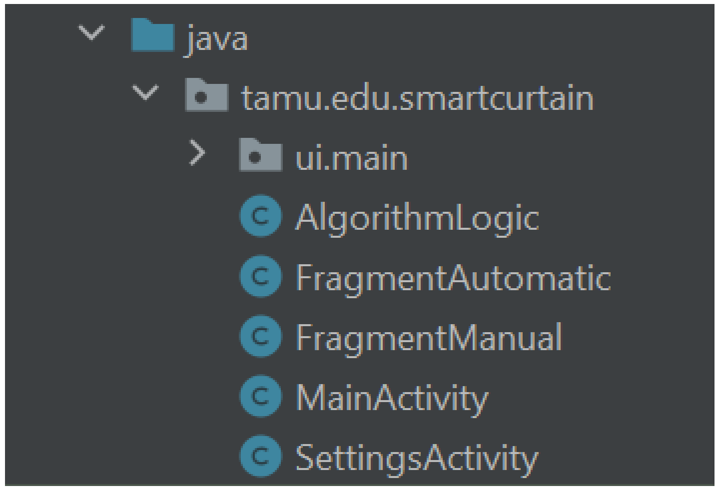


Figure 6.7) Android Application Hierarchy A

The android app hierarchy for the logic is all written through java. The algorithm logic is a class that contains our algorithm which includes determining whether or not the curtain should be open/closed. The fragments are the logic for tabs seen in the app. The manual fragment has an asynchronous thread that will attempt to send http requests including a json with commands to the switch bot. The automatic fragment has the same thread above and it has an additional thread which listens for data from the bluetooth module and contains a handler to filter the data. The activities are two different pages, the main being the page that holds the fragments as well as thread for connecting to the bluetooth module.

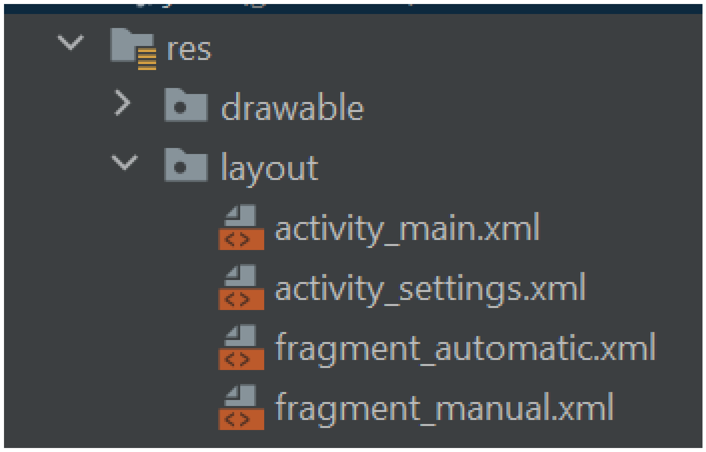


Figure 6.7) Android Application Hierarchy B

The above hierarchy is a part of the android app as well. These xml files hold the UI information such as the main and settings page. The fragments include the buttons and texts needed to display to the user.

5 Experimental results

We applied a number of tests to all aspects of our system. This section will be separated by the different areas of the product. In each section we will describe what components required testing to validate its functionality, how the tests were performed, an analysis and discussion on the results, and finally a picture depicting that aspect of the system.

Hardware - More specifically, there are two major hardware components of our product, the switchbot system, and the arduino configuration. We will begin with the Switchbot.

5.1) Switchbot - As discussed previously, the Switchbot is a product available on the market to open and close curtains suspended on a curtain rod for in-home use. We decided that this was a sufficient place to start since the Switchbot itself did not satisfy the requirements for our project, but gave us a very good starting point to implement our Smart Curtain. Specifically, the Switchbot would not monitor the environment and make decisions to automatically open and close the curtains based on the ambient temperature and light intensity in the window. The **use** of the Switchbot is to open and close the curtains on command from our android application. To see the tests and its following results, see the **Manual Mode** listed below, since that section is an extension of the Switchbot's use.

Figure 5.1) Switchbot

5.2) Arduino Configuration: Temperature Sensor - As described in our design layout, the arduino is the microcontroller that receives information from the light intensity monitor and the temperature sensor. These two components needed verification of their functionality if we were to hope to implement our product. First the temperature sensor. The ambient temperature of the room we were testing in was 74℉. This number was set by the thermostat, but it is not reasonable to expect the temperature at all points in the room to be exactly 74℉. When monitoring the temperature read by our sensor, we expected anything in the range of +/- 5℉. At the beginning, we were getting odd readings of 32℉, but the reasoning for this is that the sensor is read in celsius and converted to fahrenheit, and a error reading of 0℃ would read as 32℉. This error was later realized to be due to incorrect wiring of the sensor. After correcting this, we were able to output a reasonable reading to the arduino. This testing was conducted using a laptop hardwired to the arduino, and outputting readings to the serial monitor.

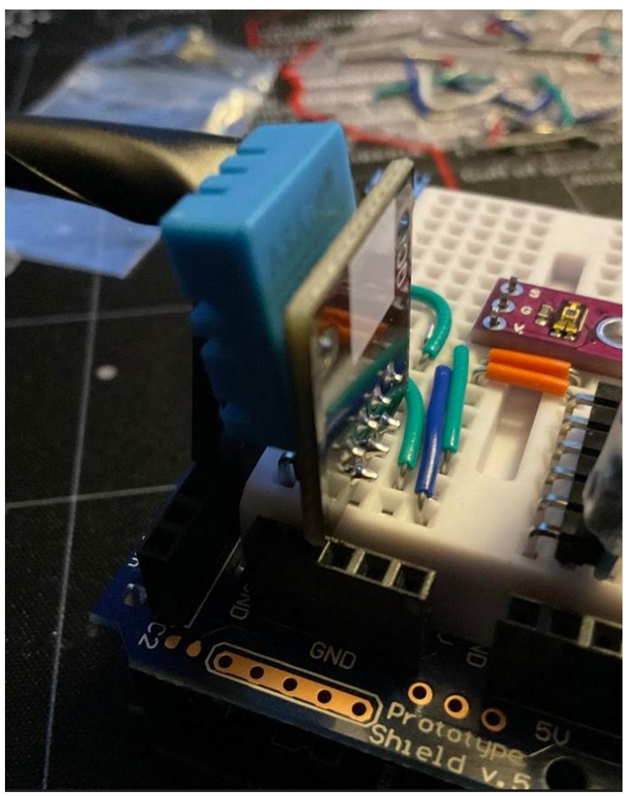


Figure 5.2) Temperature Sensor

5.3) Arduino Configuration: Light Intensity Monitor - The lighting of the room we tested in was typical for a well lit room. The sensor measures in lumens, and the average lumen count for a room varies. That being said, we noticed that in darkness, the lumen output read by our arduino was close to zero, and when exposing the component to a flashlight, read over 500 lumen. This is as expected, not because we knew the exact lumen count to expect, but because there was a distinct and constant difference in the output when we expose the sensor to light, and when we cover it in darkness. After showing a number of correct readings in a row, by shining a light on and off again, we determined that the light intensity monitor was working correctly. This specific component had some difficulties, since the connection points were loose on the components that we had ordered. This led to readings of 1200+ in darkness with no change after shining our flashlight on and off, which we recognized as incorrect readings. When adjusting the connections, the readings would suddenly go back to our expected readings. This situation was solved by confirming a good connection between our light sensor and the arduino.

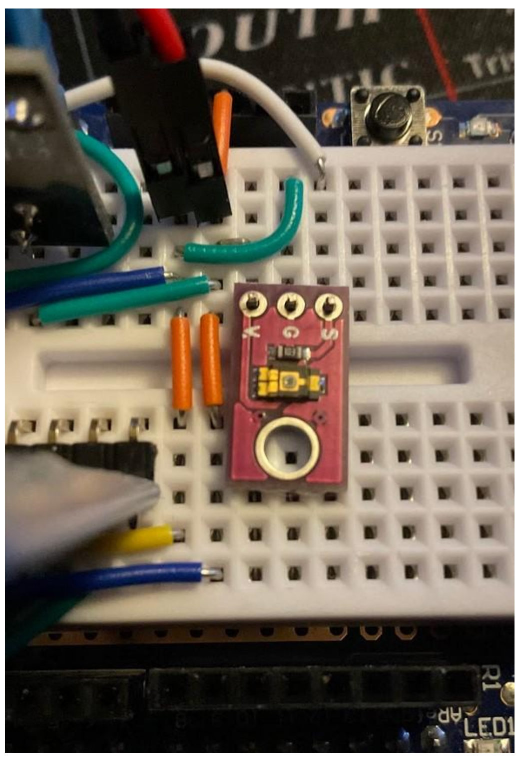


Figure 5.3) Light Intensity Module

5.4) Arduino Configuration: Bluetooth Module - The bluetooth module needed to be connected to the android device so that the device could receive information on the android application. This process began with some troubles connecting to the device, but this was resolved by going to a different site and using non-TAMU wifi. The information was able to be controlled by the arduino, and if the arduino had correct information, then the application would receive the correct information. Therefore, the only testing used with this module was confirming that the android application was receiving information. Once this was confirmed, no further testing was needed.

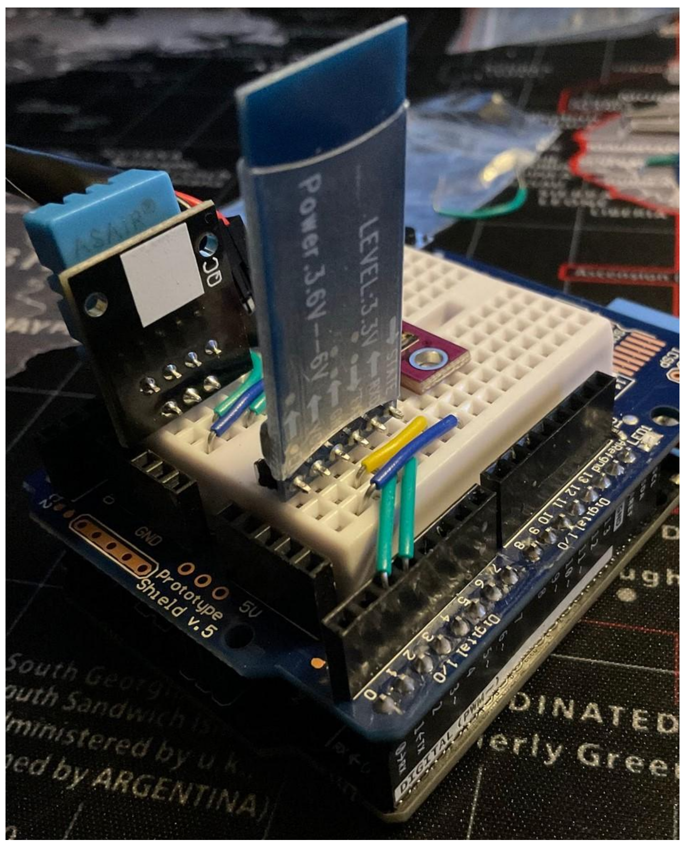


Figure 5.4) Bluetooth Module

Software/Final Product - After developing the android application, we needed to test its functionality. Since the application's functionality is directly linked to the product's functionality, we have grouped the two together here.

5.5) Application: Manual Mode - The manual mode's function is to open and close the curtains on command. The testing for this mode is as follows. When the user selects the open option, the switchbot will move along the curtain rod to the "open position'. This position is characterized by the curtain covering less than 20% of the window. If the curtain motor moves to this position when the open option is selected, then this portion of the mode passes our testing. Similarly, the 'closed position' is characterized by the curtain covering more than 90% of the window. If the curtain motor moves to this position when the closed option is selected as well as for the closed option, then the entirety of this mode passes our testing..

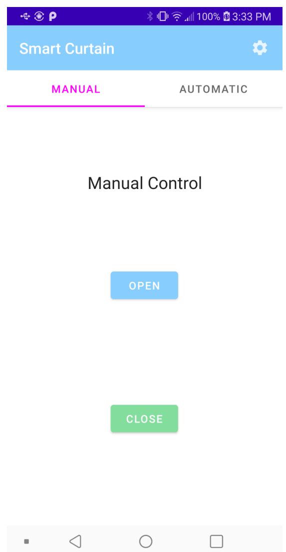


Figure 5.5) Manual Mode

5.6) Application: Automatic Mode - The automatic mode's function is to automatically open and close the curtains to optimally adjust the current indoor temperature to minimize the energy needed from the indoor climate control system. This is achieved by entering a desired temperature through the app in the automatic page. Then the system reads the window and indoor temperature, as well as the light intensity. The testing for this mode is as follows. The desired temperature is entered and if it is a high temperature, say 80℉, then the logic will send a command to open the curtain to allow heat to enter the room. Similarly if the desired temperature is low, say 60℉, then the logic will send a command to close the curtain to block heat from entering the room. If both of these situations are confirmed to work in testing, then the testing for this mode is complete.

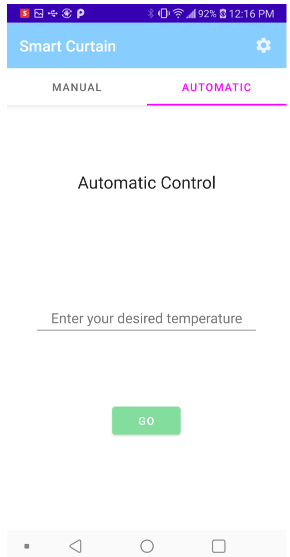
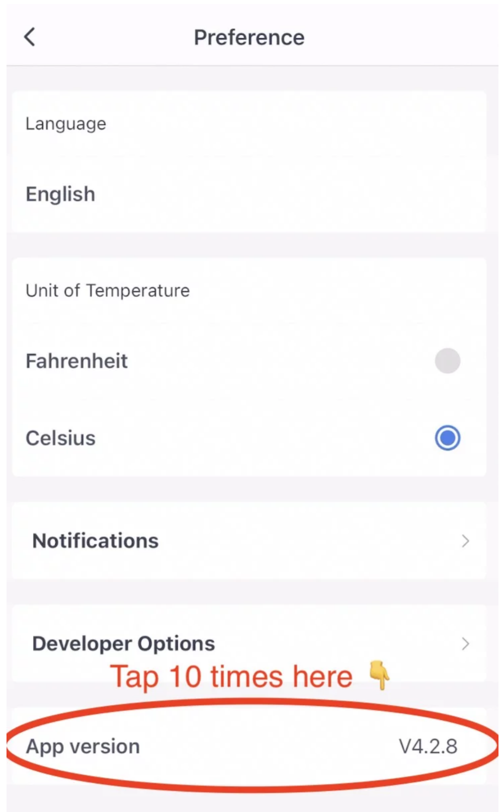


Figure 5.6) Automatic Mode

6 User’s Manuals

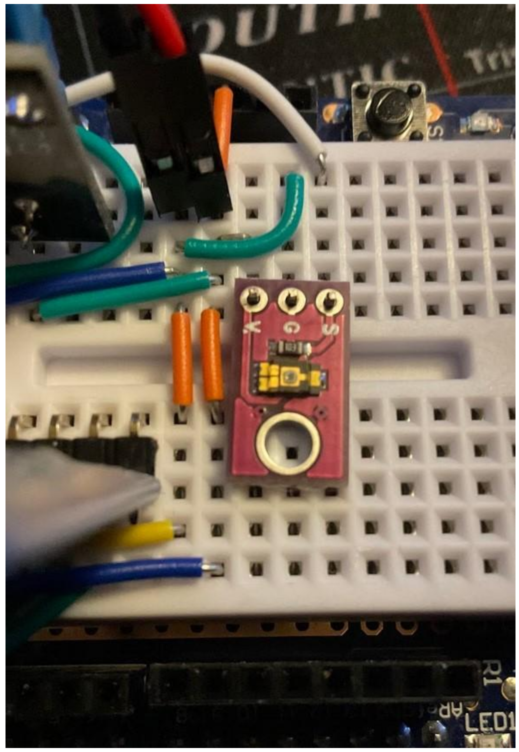
a. Software installation

This Android application has been developed and tested in Android 9 with API level 28. It may work with previous or later versions of Android, but it is guaranteed that it works on this version specifically. The code for the Android application can be found on our github repository [3]. These files can be built and loaded into the Android phone using Android Studio. This works with our specific SwitchBot curtain motor due to the fact that we have our own model number and authentication tokens. Before you do that, you need to install the SwitchBot application in order to get to the Open Token. To do that you need to connect your specific SwitchBot curtain motor to your application, and then you need to tap the App Version button 10 times. This will bring up the token which you need to save for access to the SwitchBot OpenAPI. This will give you the token needed to authenticate any HTTP requests made to the bot, and it also gives you information such as the model number in order to send the requests in the first place.

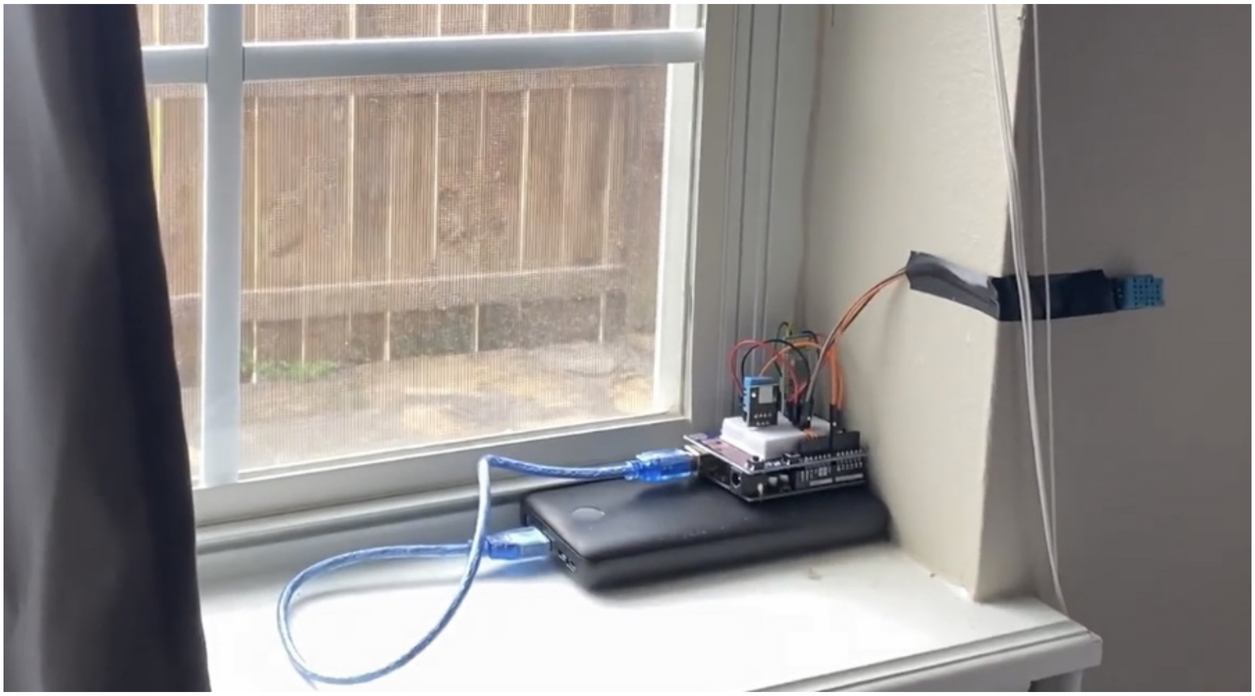


b. Hardware installation

The hardware installation is straightforward. The sensors are put into the Arduino already, but the pins are as shown below. There are two temperature sensors and one light intensity sensor for the readings. Then there is also a bluetooth module to connect the Arduino to the Android application. You also connect the battery to the Arduino in order to make the device portable.



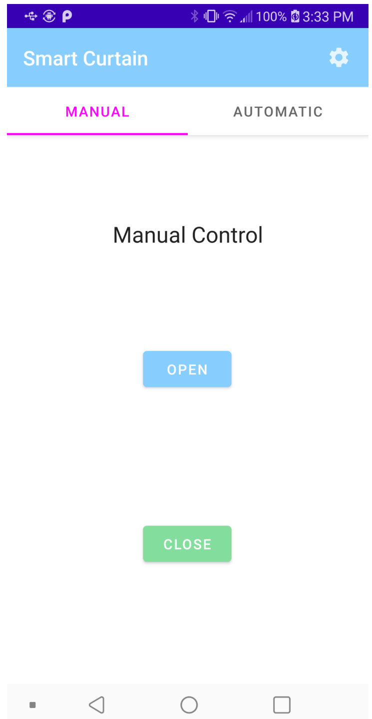
You need to install the Switchbot curtain motor behind your curtain so that it can push or pull the curtain easily. The Arduino can be installed on the windowsill also behind the curtain so that it can get accurate readings of the temperature and light sensitivity from the window.



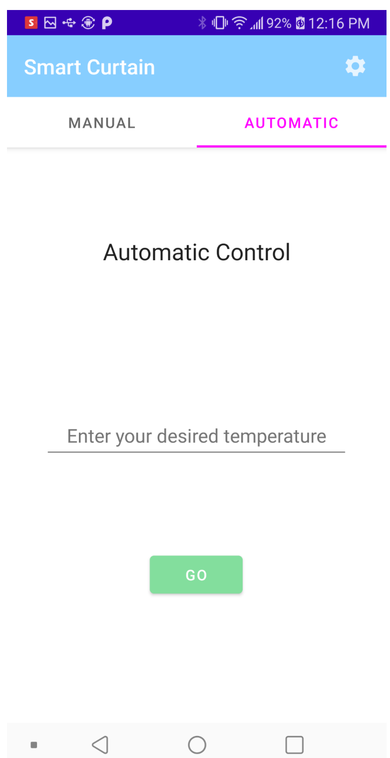
One thermometer should be exposed to the window so that it can get a temperature reading from outside, while the other should not be exposed to the outside and should be completely inside to get the temperature indoors. The Arduino is already connected to a battery pack, but it can easily be replaced if it ever runs out. If you need to use a different curtain system, you need to also download the official SwitchBot application. Then you need to register the SwitchBot curtain motor in the application. Then you need to calibrate the SwitchBot curtain motor to your specific rod and curtain system. There is a specific button needed to calibrate the SwitchBot curtain motor. It will go up and down the curtain rod to determine the specific length that it needs to go when it opens and closes. You will also need to connect to the internet in the SwitchBot application in order for the application to be able to connect to the SwitchBot hub. This is so that you can send the API requests for the push or pull commands.

c. Operation instructions

Once in the app, there is a manual mode and an automatic mode.



i.Manual mode



ii.Automatic Mode

The manual mode has both an open and a close button. You can control the SwitchBot motor manual from our app. The automatic mode is where you set the desired temperature you want in the room. If the room is hotter than you want, then it will close the curtain to remove the heat and sunlight coming from the sun. This will cool down the room, saving you electricity from changing the air conditioner. If the room is colder than your desired temperature, then it will open the window to increase the amount of heat that comes in through the window. This is done using the two thermometers and the light intensity sensor attached to the Arduino that was put on the windowsill. The Arduino will send the information to the Android application, and based on the user settings for desired temperature, it will send an HTTP request to the cloud hub which is connected via bluetooth to the curtain motor.

7 Course debriefing

Team Management

The roles and responsibilities were given to each member of the team at the beginning of the project, and we tried to assign roles based on our strengths. The team has worked well together through these roles and responsibilities, and we have all been able to work on different aspects of the project, even if the person working on a task is not directly responsible for that part of the project. Below is a quick recap on which team member is responsible for each part of the project.

Arthur Chen - Team Leader. David Erdner - Systems Design. James Streets - Hardware Design. Pierre Vu - Software Design. Caleb Key - Testing, Technical Reporting.

As mentioned above, every team member has been able to contribute to other parts of the project aside from the part that they are directly responsible for. As an example, James Streets -

in charge of hardware design, has been very helpful in designing the Android app that we will be using to control the curtain system, as well as switching between automatic and manual mode. This has been a very important part of how the team manages the project, because interprocess communication allows us to be constantly in touch with every aspect, from hardware to software to testing.

Throughout the project, we stayed in constant communication over discord. This tool allowed us to easily send messages, host meetings, share files, and stay on schedule. Every time someone works on the project on their own time, updates are immediately sent out to the rest of the team so that everyone is on the same page. Outside of virtual meetings that we generally had every Monday evening, the team was committed to working on the project before and after our weekly Thursday meetings with the Professor and TA, as well as meeting on weekends to make serious headway on configuring and testing the project.

The Gantt Chart was our main way of keeping track of where we are on the project schedule, and we have been updating that weekly. A few things (delay in parts arrival from Amazon, parts malfunctioning, Android Studio issues) came up that have made it difficult to stay exactly on schedule with where the Gantt Chart would have us, but we still have been able to follow the schedule closely. Overall, the project management style that we used worked very well, and allowed the team to be successful on our project. If we were to do this project again, we would use the same project management schedule.

Safety and Ethical Concerns

There are not many safety concerns for our product, but we did still have to take this part of development into consideration. The majority of our safety concerns are handled by how well the user is setting up their curtains. If the user does not properly mount the switchbot on the rod, then they are at risk of the bot falling off the rod, or the whole system coming down, which could be harmful to the user. The area around the curtain must also be properly cleared so that the curtain automatically opening and closing does not knock anything over.

The only real ethical question that our project poses is if more automatic home appliances increases human reliance on technology for simple tasks. This question could be asked of any smart home service, like Amazon Alexa being used to turn off and on the lights, or Google Home being used for playing music. One could argue that all of these smart appliances promote a sedentary lifestyle, and we should not continue to increase our reliance on technology for simple tasks.

Environmental

One of our main goals for this project was to decrease the amount of energy saved by homeowners using our product. According to a couple separate reports, 10%-25% of thermal energy loss is through windows, and 25% of this energy can be saved with a blackout curtain.

If our system is used in automatic mode, our system can save the average homeowner roughly $125 per year, which is a massive amount of energy if implemented on a wide scale. Thus, our product has a vastly positive impact on the environment.

Social

One potential social impact that our project could cause is disturbing the users. Curtain motor can be loud, so turning on and off while the user is sleeping could be distracting. This could cause the user to become frustrated or upset with the system, which could lead to them scrapping the product. However, this is pretty unlikely to happen considering that the curtain can be put in manual mode, so the user simply just has to switch to manual mode to avoid this problem.

Political

A political concern that was brought up during the project was the privacy concerns associated with the curtains. The curtains can open and close on their own when in automatic mode. This is dependent on the temperature and light intensity in the window. Because of this, the curtains can open at unexpected times, and since this is in a home, there could be privacy concerns. This can of course be mitigated by the user by not using automatic mode, or not using the product in certain areas of the house, but this is not a good solution because it would dissuade the use of our product. A potential solution would be to implement a privacy mode that keeps the curtains from automatically opening and closing during certain hours of the day.

Economic

Another large advantage of our project is its economic feasibility. As mentioned above, this product could save the user around $125 per year, meaning that the price to buy the system would quickly pay for itself. It costs around $200 (see the budget section) depending on the setup configuration, so the user could quickly see the benefits in terms of money saved by using our product.

Manufacturability

Since most of the project is done through the Android App, the manufacturability for the product is very manageable. Only a few hardware components are actually used in our project, and most of these products are already produced by other companies, so this is not really an issue.

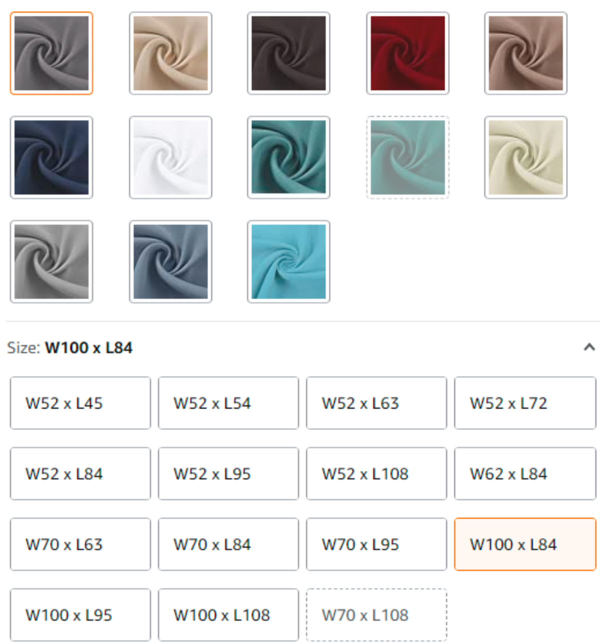
8 Budgets

A detailed summary of the costs of the prototype are below, and includes the sources for each product.

Component	Price	Link
Microcontroller kit	$38.99	microcontroller
SwitchBot Curtain motor	$99.00	SwitchBot motor
Curtain Remote	$16.15	Curtain remote
Extra Thermometers	$10.29	thermometers
Light Intensity Sensor	$10.99	Light intensity sensor
Bluetooth Module for Microcontroller(2)	$18.00	Bluetooth module for microcontroller
Rod for holding curtains	$19.00	Rod for curtains
Black Out curtains	$16.00	Black out curtains
Switchbot Minihub	$39.00	Switchbot Mini Hub
Total	$267.42	

As you can see, the main costs for our project comes from the Switchbot curtain motor, the Arduino microcontroller, and the Switchbot Minihub. All other expenses fell below $20, and some of the costs can even be eliminated. For example, the curtain remote was deemed to be unnecessary, and we only utilized one of the bluetooth modules for the microcontroller. Elimination of these parts would bring the price down by about $25, making the total cost $242.27.

One part that is worth mentioning that could add or subtract to the total is the black out curtains. For our prototype, we bought a single curtain that cost $16.00, and was 52in by 45 in. On amazon.com, there are many different options in terms of size and color, and each one comes with a slightly different cost. Below shows all of the possible options for the blackout curtain that our prototype utilizes.



These prices range from around $15 for the smallest one, to $32 for the largest one. Overall, this option would be up to the user, and if our product was produced in mass, we might even give the user the option to buy their own bar and curtain so that they are not subject to whichever one our system comes with. Doing this could bring down the price even more from $242.27 to $207.27. Furthermore, due to the COVID 19 pandemic, the price of microcontrollers has increased significantly, and as this price dwindles back down after the silicon shortage ends, the total price for our product could fall below $200. Overall, our final budget according to the numbers above is even lower than our initial budget because of the fact that our product did not need to use some of the components that we ordered. This is a good sign in terms of mass production, because the less parts needed, the better.

9 Appendices

Here you can include any additional information not already in the implementation notes (e.g., detailed circuit schematics, class hierarchy) that would help replicate or maintain your implementation.

[1] M. Zhu, R. Diao, J. Fu, X. Jiang, en Y. Zhang, “Study on the influence of key parameters of exterior window structure on building energy saving effect”, IOP Conference Series: Earth and Environmental Science, vol 675, no 1, bl 012051, Feb 2021.

[2] Szkudlarek J, Snycerski M, Owczarek G. “Proposal of Objective Assesment of the Phenomenon of Light Passage through Blackout Fabrics”, FIBRES & TEXTILES in Eastern Europe 2017; 25, 4(124): 50-58. DOI: 10.5604/01.3001.0010.2663

[3] Github Repository: https://github.tamu.edu/arthur00/Smart-Curtain

[4] Josh Peterson, Planet Green, “Consider Blackout Curtains to Save Energy and Control Light and Noise”, HowStuffWorks, https://home.howstuffworks.com/green-living/blackout-curtains-save-energy.htm

[5] Craig, “How Much Does It Cost to Run a Central Air Conditioner?”, Appliance Analysts, https://applianceanalysts.com/cost-to-run-central-air-conditioner/