Smart Curtain System

Critical Design Review

Team 3N3

Arthur Chen

David Erdner

Caleb Key

Pierre Vu

James Streets

Department of Computer Science

Texas A&M University

Date 03/28/2022

Table of Contents

1. Introduction

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.

Scope/Background - The scope of the Smart Curtain project is in Automation, or specifically Home automation, otherwise known as domotics. This is a branch of robotics that includes the automatic monitoring and control of lighting, entertainment, temperature, security, as well as home appliances. In this case, the Smart Curtain project focuses on in-home climate control.

Needs - 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.

Goals/Objectives - The goal of our system would be 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.

To achieve this our first objective will be to automatically control the curtains through some electronically controlled medium. Requirements to meet this objective include electronic control of the curtains, full range of motion for the mechanical controls, and satisfiable conditions for the *closed* and

open settings. The *closed* setting should allow as little energy through the window as feasible for the given curtains. The *open* setting should allow as much energy through the window as possible. The second objective is to have an acceptable and readable measurement system. The two necessary measurements include indoor temperature, and light intensity in the window. Requirements to meet this objective include readable and accurate measurements using an indoor thermometer for the indoor temperature, as well as using a light intensity sensor for the window light intensity. Finally, we would be able to combine these to create a fully automatic smart curtain system that would control the curtains based on temperature and light intensity.

Design Constraints - 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.

Validation/Testing - We will test the basic functions of the curtain by manually sending the open and close commands to the motor at first to make sure that the motor functions work before we start to work on the algorithms. Then we will test the thermometer and light sensors using simple output statements once we figure out how to incorporate them into the software. A more indepth look will be laid out in the "**Updated validation and testing procedures"** section.

2. Proposed design

2.1 Updates to the proposal design

Our original proposal left room for interpretation for the user input for the preferred settings of the application. We were originally considering many options, including web applications, handheld remotes, and mobile applications. We decided to go for a mobile app, more specifically an Android application. This would allow us to stick with the bluetooth theme that many of the other connections in our project have. It would be easier to implement bluetooth than wifi, and it is easier to create and use Android apps as opposed to other options.

One alternative solution that we found while researching was the fact that the SwitchBot has a light sensor built into it. The reason our solution would be better would be to have a dedicated light sensor that can more accurately read the light. The SwitchBot light sensor is attached to the motor, so it would be

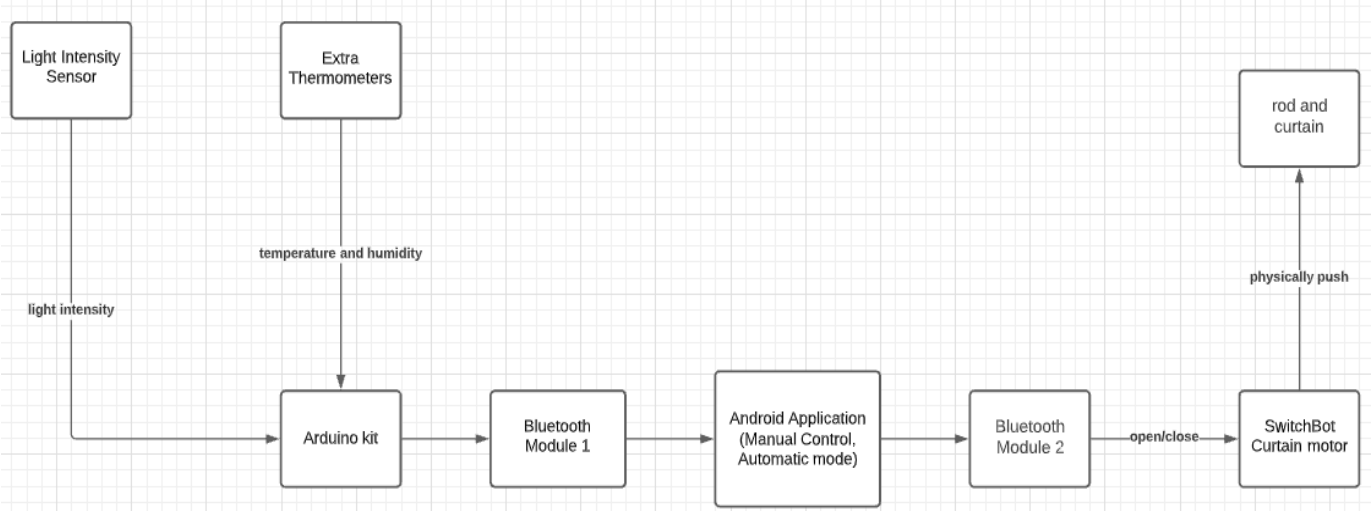
harder to read the light intensity. We also add a temperature sensor as that would have a greater impact on saving energy which is what our project intends to do as opposed to just the brightness of a room.

Due to the use of bluetooth modules in multiple steps of our project, the options arise to change the design between each of the components. As we haven't come across any problems with our design, we will keep going with it until a problem arises.

The budget for this project was purposely left open as there are many small parts that need to work together for it to work. This means that we have room to order different parts or replacements if the ones we receive do not work.

2.2 System description

High-level data flow chart:



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 send a bluetooth signal to the microcontroller.

Bluetooth modules

These modules will be used for communication between the microcontroller and the remote for the curtain. This will also be used to send commands to the microcontroller.

Curtain Remote

This remote will be used to send signals to the motor of the curtain and tell it to open or close. It will receive those commands from the bluetooth module that is attached to the microcontroller.

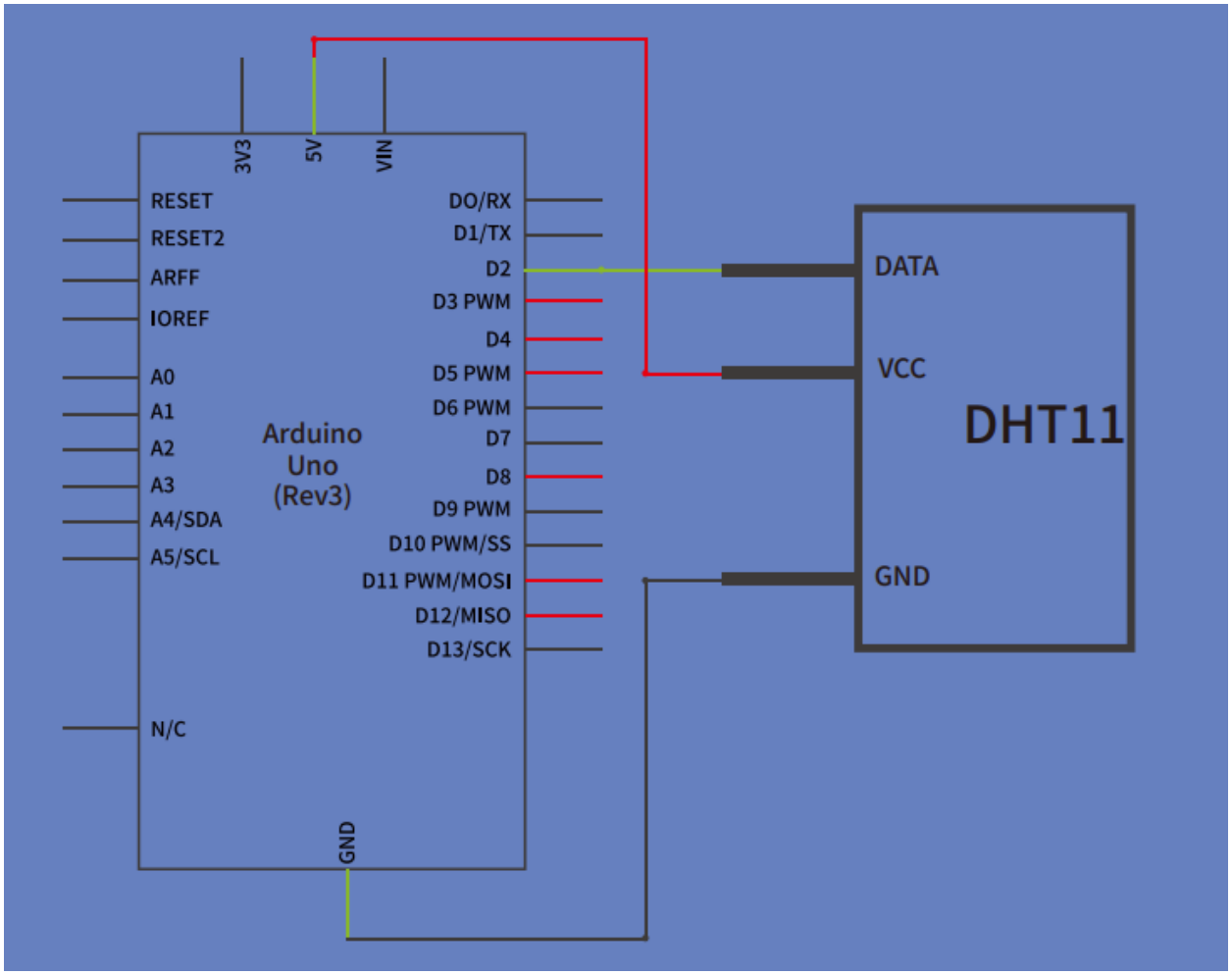
SwitchBot Curtain Motor

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

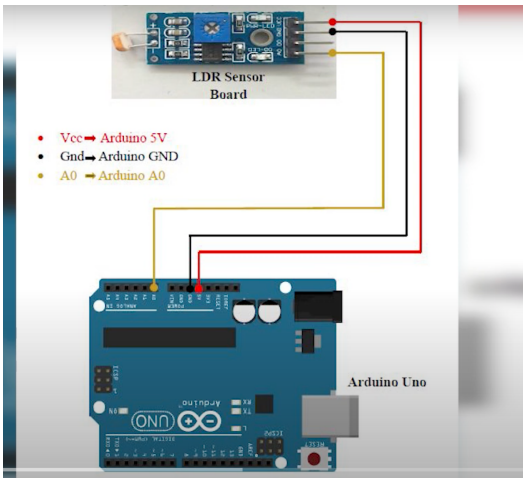
This is a standard curtain rod that is used for holding curtains. It needs to be able to move freely, so that the motor can work. The curtain is going to be a blackout curtain because that will be the most effective in blocking out sunlight and maintaining temperature inside. Any amount of translucency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

2.3 Complete module-wise specifications

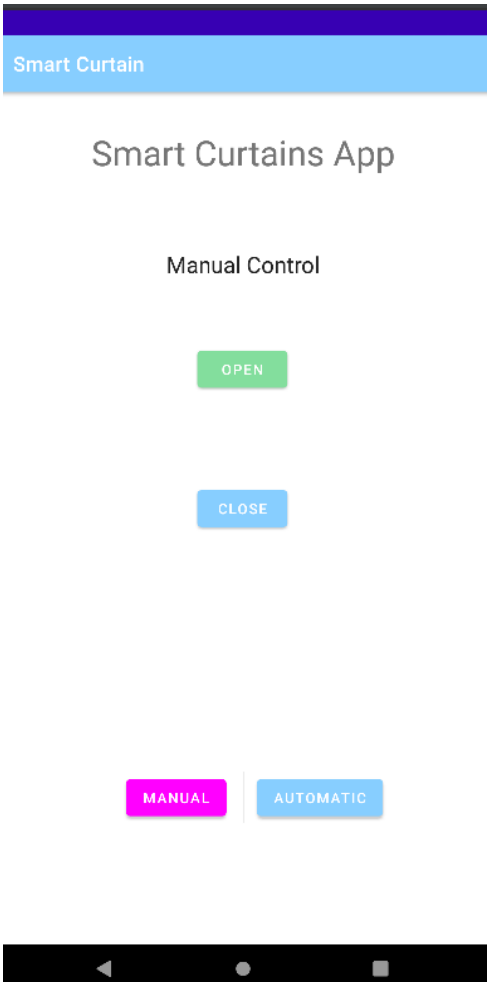


DHT 11 Temp/Humidity Sensor Connection

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.



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
Total	$228.42	

Parts List

The parts list has remained unchanged since our initial proposal. Each part has arrived with the exception of the curtain rod. We have been able to use temporary solutions while waiting, so this has not delayed testing and development. 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. Project 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.

To update on the mechanisms that the team is using to manage the project, we have still been in constant communication over discord. This tool allows 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 have every Monday evening, the team has been 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.

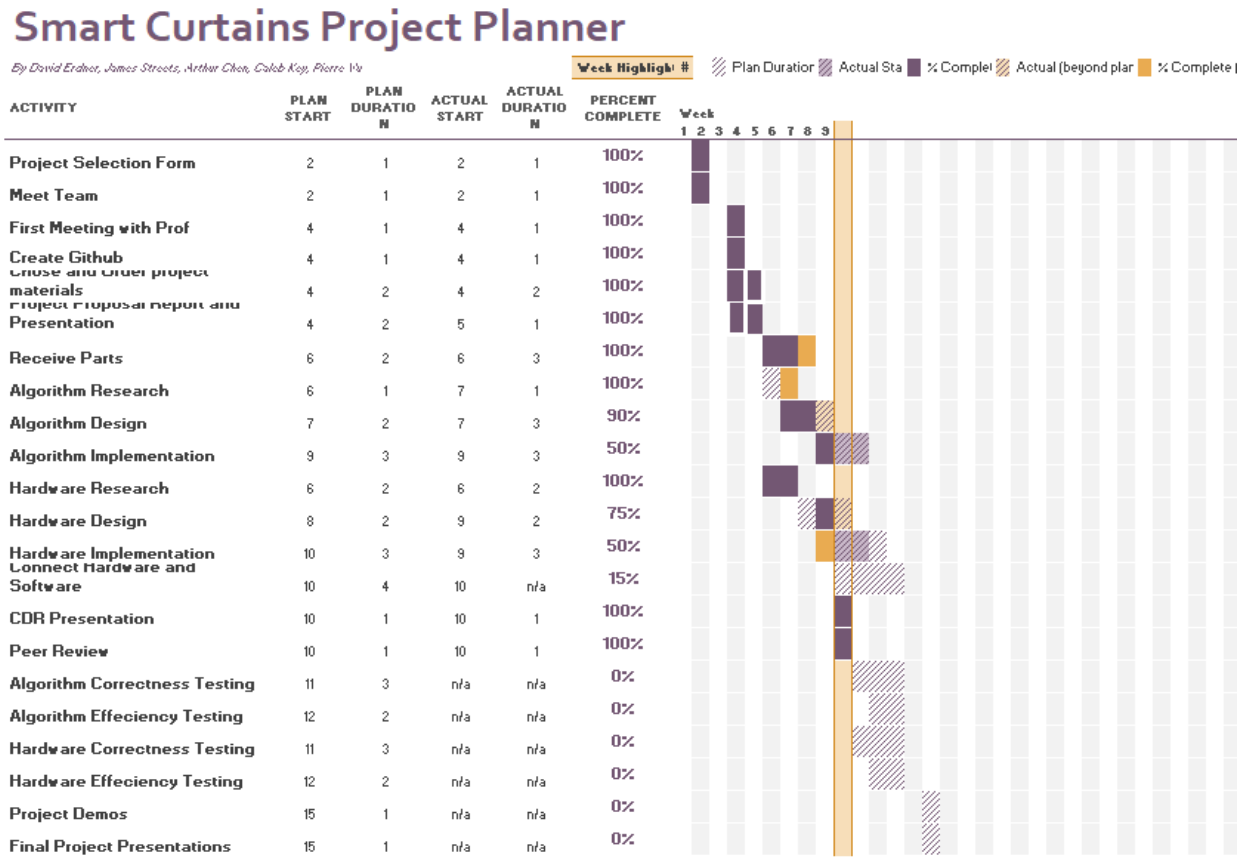
The Gantt Chart has been our main way of keeping track of where we are on the project schedule, and we have been updating that weekly. A few things (delay in parts arrival from Amazon, parts malfunctioning, Android Studio issues) have come up that have made it difficult to stay exactly on schedule with 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 have been using has worked very well, and the team has made very good progress on the project so far. There is still a lot of work to be done, but the way

that we have been managing the project so far is the way we will continue to do it throughout the rest of the project duration.

3.1. Updated implementation schedule

Below is the updated Gantt and Flow Chart for the project implementation.

Gantt Chart:

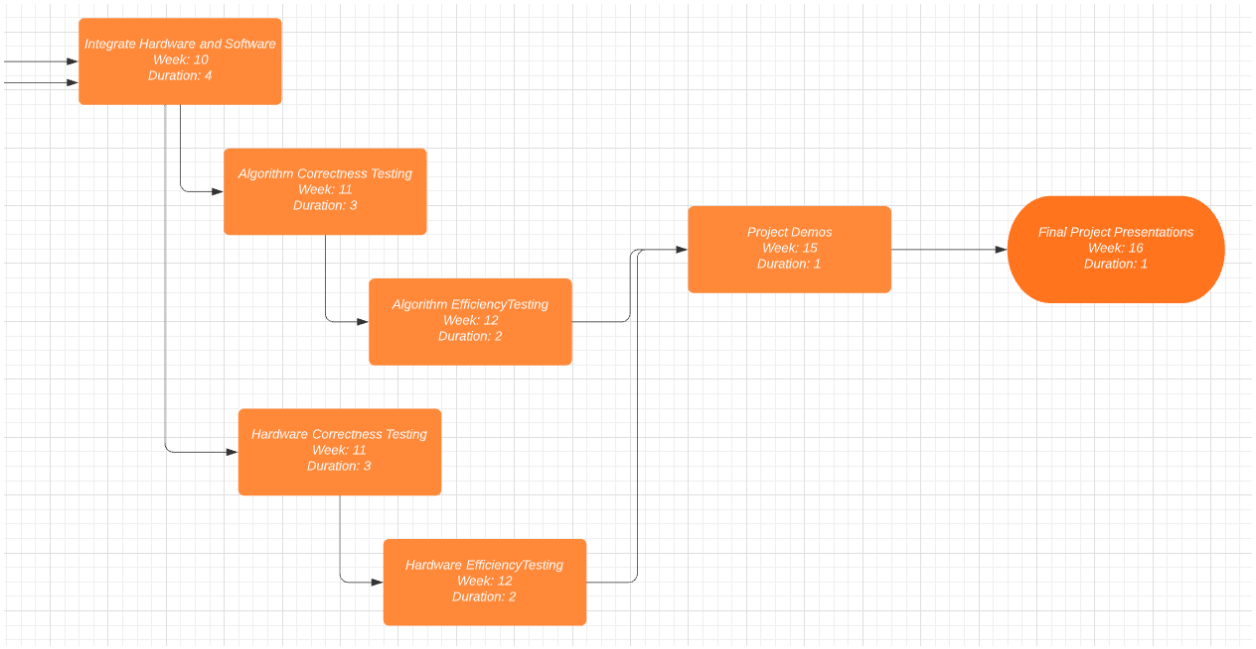


Tasks and Dependencies Flow Chart

(Critical Path follows top arrows and is 16 weeks in length)

Link (Must have Lucid account to view):
https://lucid.app/lucidchart/90d94c2f-fc2a-4fff-a340-663b13384dc0/edit?invitationId=inv_db9dd6bf-7e75-4846-9719-195d84b5b5a4


Smart Curtain Tasks and Dependencies
Class Begins
Week: 1
Duration: 1
Project Selection Form
Week: 1
Duration: 1
Meet Team
Week: 2
Duration: 1
First Meeting with Prof
Week: 4
Duration: 1
Create Github
Week: 4
Duration: 1
Choose and Order Project Materials
Week: 4
Duration: 2
Hardware Research
Week: 6
Duration: 2
Receive Parts
Week: 6
Duration: 2
Hardware Design
Week: 8
Duration: 2
Hardware Implementation
Week: 10
Duration: 3
Algorithm Implementation
Week: 9
Duration: 3
Algorithm Research
Week: 6
Duration: 1
Algorithm Design
Week: 7
Duration: 2
CDR Presentation Prep
Week: 9
Duration: 2
CDR Presentation
Week: 10
Duration: 1
Peer Review
Week: 10
Duration: 1



3.2. Updated validation and testing procedures

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.

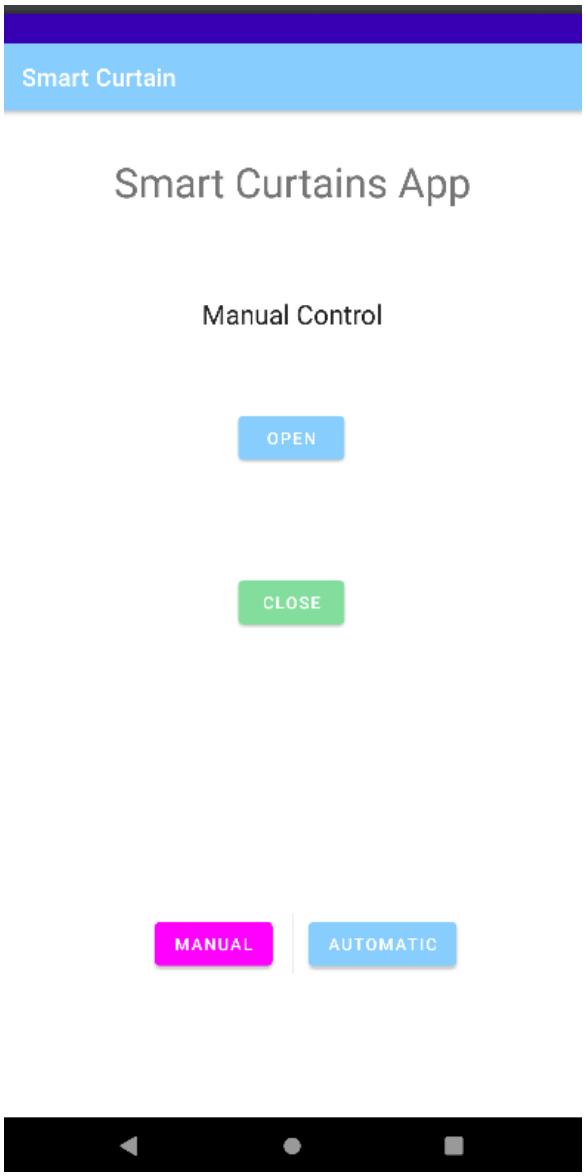
Step 6: If any of the previous steps fail to work for any reason, the system testing failed. Otherwise, it is considered a pass for automatic control mode.

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.

3.3. Updated division of labor and responsibilities

Currently we are on week 10, and there is the list of labor division with due dates:

1. Android app frontend (due by the end of week 11, but almost completed)- James



Demo of frontend screenshot

2. Android app algorithm implementation(due by the end of week 11) - James, Pierre

   According to our improvement in system design, we decided to perform the computation and decision making on the android app instead of the Audrnio as previously proposed.

3. Algorithm correctness and efficiency testing(due by the end of week 12) - Pierre

   Pierre will be writing all the unit tests on our Android and aiming for at least 95% testing coverage. Additionally, Pierre will calculate the space and time complexity of the algorithm and

aim for at most O(n) on both space and time, where n is the total number of data sampling and user input.

4. Arduino hardware code for data input and output (due by the end of week 13)- David, Arthur

   This task will include the correct pin configuration and wiring between the sensor components and the Arduino board.

5. Software and Hardware synchronization(due by the end of week 14) - David

   This portion of the project will be relatively easy, just make sure the sampling frequency of the hardware is higher than the frequency of data intake of the software, which is correct of course.

6. Hardware correctness and efficiency testing (due by the end of week 14) - Arthur, Caleb

   The expectation is that the Arduino interface is able to display the sensors reading data correctly with required frequency. Therefore, the maximum frequency of sensor input will be tested and the correctness will be tested against the weathering app on any smartphone, where we aim the error to be within 5%.

7. System-level testing (due by the end of week 15)- Caleb

   The system-level testing will be performed from a user perspective. Imitating the actual user experience.

8. Management-related design and documentation (due by the end of week 16) - Arthur

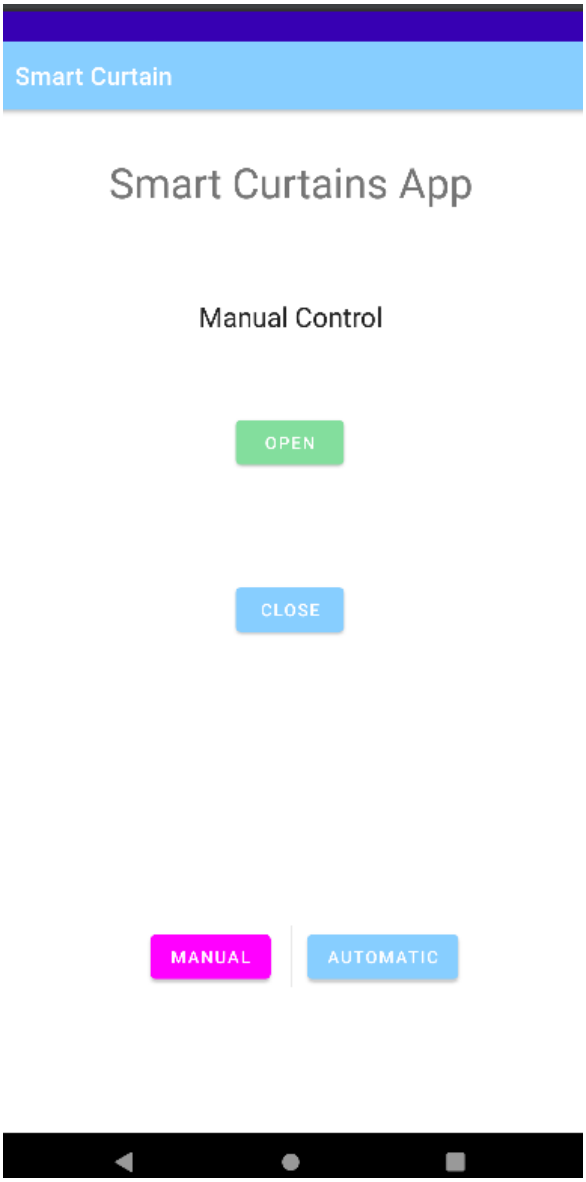
   This portion of the project includes scheduling weekly meetings and log generation and making sure the documentations are submitted on time.

4. Preliminary results

Preliminary testing has been completed on the following components: light intensity, temperature, and curtain motor. This has allowed us to complete separate parts of the system including the climate monitoring system, the curtain maneuvering system, as well as the android application that will be issuing instructions from the user. The separate demos will be shown below. The temperature worked very similarly, and correctly read the room temperature.

https://drive.google.com/file/d/1wYVM4N7gseh7eC6GmLX9wZ4SGycOm-pc/view?usp=sharing

The link above is a video where we can see the connection between our arduino and light intensity sensor. This connection is shown in our previous section. Running the code shows the output seen in the demo. This will output a low reading (<20) when blocking the sensor from light and a much higher (>200) reading when we move the sensor to the light.



Above we can see a demonstration of our android application. In this app we will give users the option to choose manual or automatic control. The functionality of both settings have been previously explained. In the manual setting, the user will be able to open and close the curtains from the app. In the automatic setting, the user will be able to decide a desired temperature, and our monitoring system will work with the curtain control and change accordingly.

https://drive.google.com/file/d/1qjbAzkNfjPXUMlsIDyr0aI7c6bvrLcid/view?usp=sharing

The link above is a video where we demonstrate the curtain control system. When set to open, the motor will start to move the system to the predetermined open position. As it moves, it will drag the curtain with it. As long as there are no obstacles, the curtain will successfully open. Similarly for the

closed setting, the motor will move the system to the predetermined closed position, dragging the curtain with it. In the demonstration, the window opens halfway, before closing. This is because the curtain rod that we were testing on has a supporting bar in the middle, interfering with our motors ability to continue moving along the rod. This is a non issue as this was just a testing rod/curtain setup. In our final system setup, there will be no interfering rod, and once we receive our curtain rod, we will switch and continue testing on that setup.

This is the progress of our current system, with the monitoring system, curtain moving system, and application functional. Moving forward we will be connecting these separate parts into one cohesive system.