Final Project in Instrumentation and Control Laboratory

**Arduino Home Automation System: Controlling Indoor and Outdoor Illumination through Mobile-To-Bluetooth Module App Connection**

CEAT-04-701A

Alagao, Frank   
Asistores, Jennalyn   
Badong, Danmar Joseph   
Baquiran, Rafael   
Barleta, Marc   
Bernardo, Raevon Thaddeus  
Cabosor, Jolly Joy   
Del Pilar, John Erick   
Fabito, John Kevin   
Ferrer, Romel   
Gaviola, Jayson   
Hermanes, Mark   
Montaron, Zedrick   
Nario, Aldrin John   
Pasion, Celorie   
Pimentel, Fernando   
Solis, Johnloyd   
Tayactac, Jhoanna

**Abstract**

Automation, especially in this modern-day era, has always been highly pursued by competitors in the industrial market. This is due to its inherent capability of making industrial processes relatively easier, safer, orderly, and efficient. However, although industrial application is usually the norm, it can also be found being innovatively integrated to other fields as well, more specifically in lighting systems of residential areas where lights are controlled or automated for convenience or home aesthetics. With this in mind, the researchers have designed, programmed, and developed a working house prototype that integrates the concept of home automation. In this project, the researchers have created a mobile application called “Alapa App” to communicate with a HC-05 Bluetooth module attached to an Arduino Uno. The main idea was that the said application will be given the role of enabling any mobile phone to serve as a remote-control device that can send specific commands to the module which will be transmitting them to the Arduino system for command identification and execution. Depending on the nature of the command received, corresponding indoor or outdoor LED lights can be toggled or controlled. Furthermore, the house prototype itself was installed with automatic lighting systems, automated within the Arduino system’s algorithm as well.

**Introduction**

Ever since the introduction and rapid growth, development, and integration of technology in our society, the world has undeniably progressed towards a more technologically-centered era wherein numerous local and global competitors across every field or market has integrated technology in their research and development as well as in their products and relative systems to adapt to ever rising technological trends. The reason behind all of this is that technological advancement is proportional to the betterment of human life in every aspect (Grant, 2019). Although technology brings forth both positive and negative effects, with proper utilization, it improves all areas of life, affecting the way people communicate, think, learn, study, and work. Prime examples of this are the significant improvements toward human communication which made it more convenient for people to communicate with each other, and automation of industrial or residential systems which made work more efficient and scalable (Abdoullaev, 2021).

Regarding the topic of automation, its applications are well spread-out and can be seen being applied especially in industrial sectors. This is because long and tedious processes that were previously manually monitored and handled by humans before are now being automated through machine programming, which takes the heavy load off of workers and making their job easier and more efficient. Another example of its application can be seen in residential dwellings, which are called “Smart Home” systems nowadays, wherein light systems or security have been automated for people’s convenience (Ezlo, 2022). However, in general, these systems have many practical applications aside from automating home illumination and security. People can also remotely control other home appliances or mechanical house compartments, monitor current climate and weather predictions, increase or decrease audio volumes in home speakers, etc. Essentially, home automation provides convenience for residents.

Adapting this very concept, the researchers have conducted a study with the aim of creating a house prototype that has Arduino automated LED lights installed that can be toggled on or off or configured using an original mobile application through Bluetooth connection with a HC-05 Bluetooth module. The house prototype will also consist of other electronic components such as PIR sensors and photoresistors that will affect how certain LED lights will behave. The concept of home automation will be implemented through the usage of Arduino, which is an open-source electronic platform (Arduino, 2018) that will help the researchers create specific algorithms for their prototype’s looping Arduino system.

**Objective of the Study**

The study was conducted by the researchers with the aim of constructing a “Smart Home” house prototype that has Arduino automated LED lights installed. Furthermore, the researchers also aim to create a mobile application that will act as a remote-control device that will control most of the LED lights of the house prototype through Bluetooth connection.

**Significance of the Study**

The study’s documentation of creating a mobile application and house prototype will serve as a project tutorial that can help people understand the concept of home automation. The study is also expected to be significant to the following:

* **Elderly and handicapped people**.Controlling home automations using mobile phones provides convenience for elderly as well as handicapped people when it comes to turning on or off lights and other electrical loads in the house. With the implementation of this system, they will not be required to walk to the actual place where switches are installed just to toggle room lights as the lights will be controlled through the application via Bluetooth connection.
* **Homeowners.** The study provides various types of concepts and ideas that homeowners can integrate to their dwellings, especially when they want to remotely control illumination or other appliances inside the house while avoiding walking around the house to reach designated light switches or appliances. This will also save time for them to prepare for online meetings, making it convenient for them.
* **Electrical engineering practitioners.** In general, this study can serve as a relevant study for electrical engineering practitioners as the prototype involves innovative ways of implementing home lighting automation systems. Furthermore, in proposing electrical plans to certain clients, this study can serve as an innovative idea that can be suggested in order to acquire the clients’ interest as it may be in-line with their preferences.

**Instrumentation and Control**

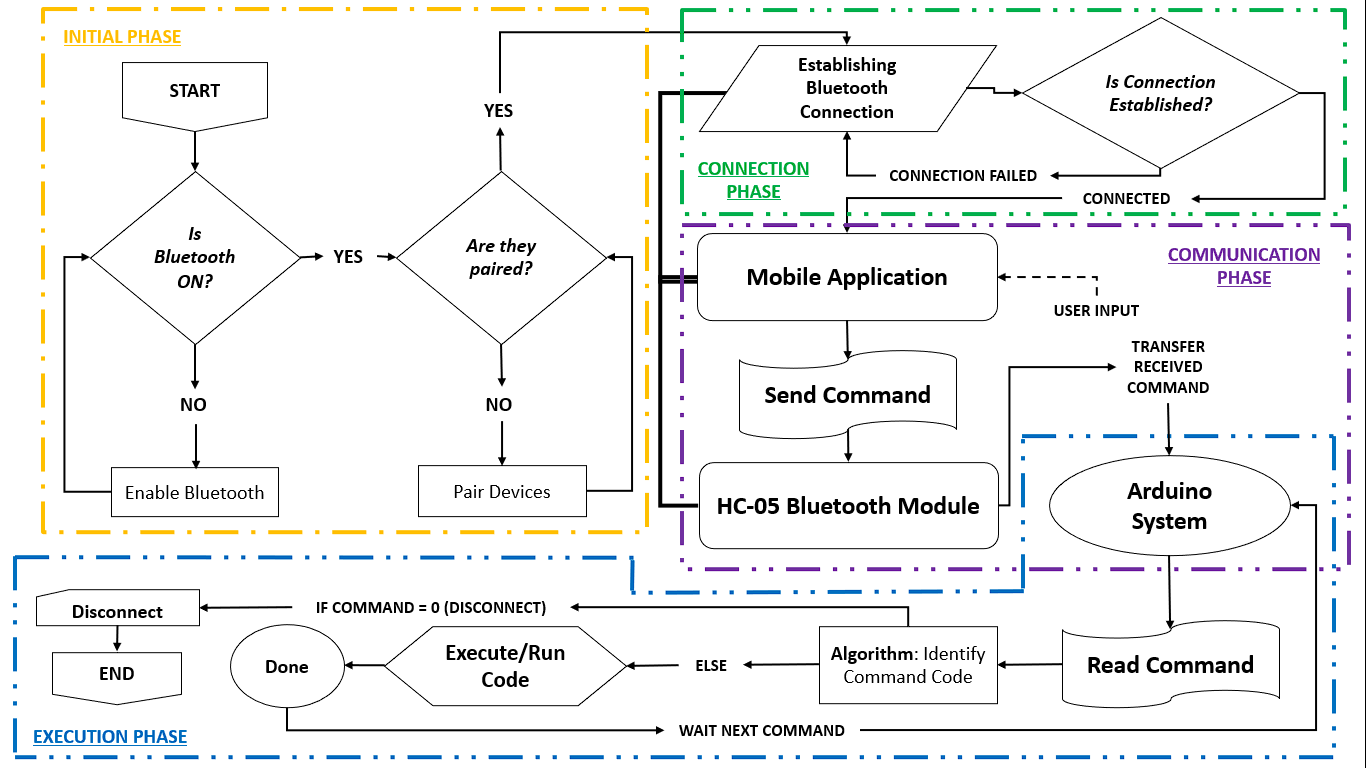
The implementation of instrumentation and control in this study’s prototype is seen through its home automation system where house illumination can be monitored and controlled through the use of a mobile device using this study’s original mobile application. Its implementation can also be observed through the prototype’s automatic outdoor night lights feature and motion sensor-triggered entrance lights.

**Definition of Terms**

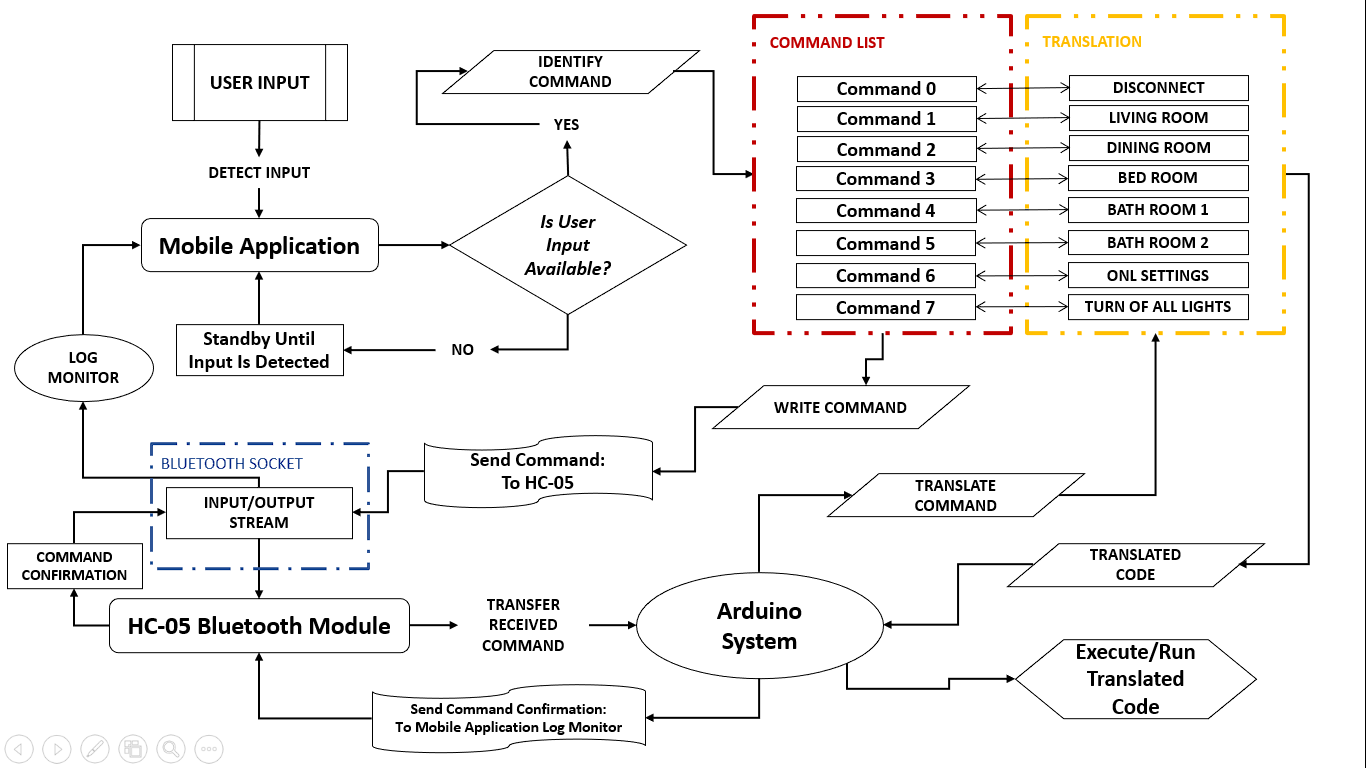
* **Arduino** – an open-source electronics platform for programmers.
* **Arduino board** – a hardware development board used to design electronic circuits.
* **Arduino software** – an Integrated Development Environment (IDE) for Arduino programming language. A software used to develop runnable codes in circuits.
* **Bluetooth** – a wireless technology that uses frequency to share data within a limited distance.
* **Bluetooth socket** – used for incoming and outgoing connections.
* **HC-05 Bluetooth module** – a Bluetooth Serial Port Protocol (SPP) module for allowing Bluetooth connection to be possible between an Arduino board and a remote device.
* **IDE** – Integrated Development Environment (IDE); an application that aids the development of software code efficiently.
* **LED** – Light Emitting Diode; a semi-conductor device which emits light when current passes through.
* **PIR** – Passive Infrared Sensor; an electronic sensor that detects motion.
* **SPP** – Serial Port Protocol; a serial communication interface for the transmission of information.
* **Smart Home** – a residence that implements home automation or the monitoring and controlling of home appliances and systems.

**Design Process**

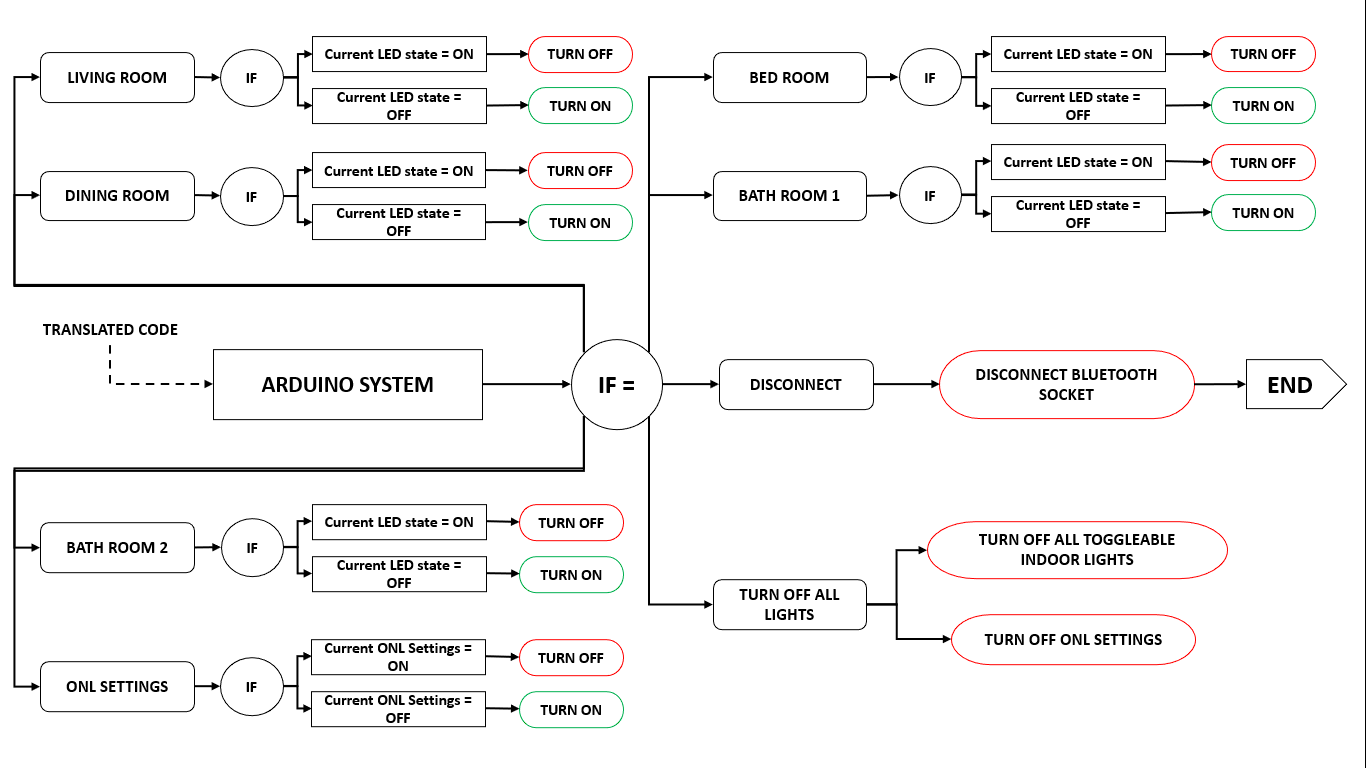
In order to systematically construct the prototype, the researchers have mapped conceptual flowcharts that will guide the designing and construction phase of the house prototype. The flowcharts are segmented into one general flowchart and three specific ones, to provide in-depth analysis. The following are as follows:

General Design

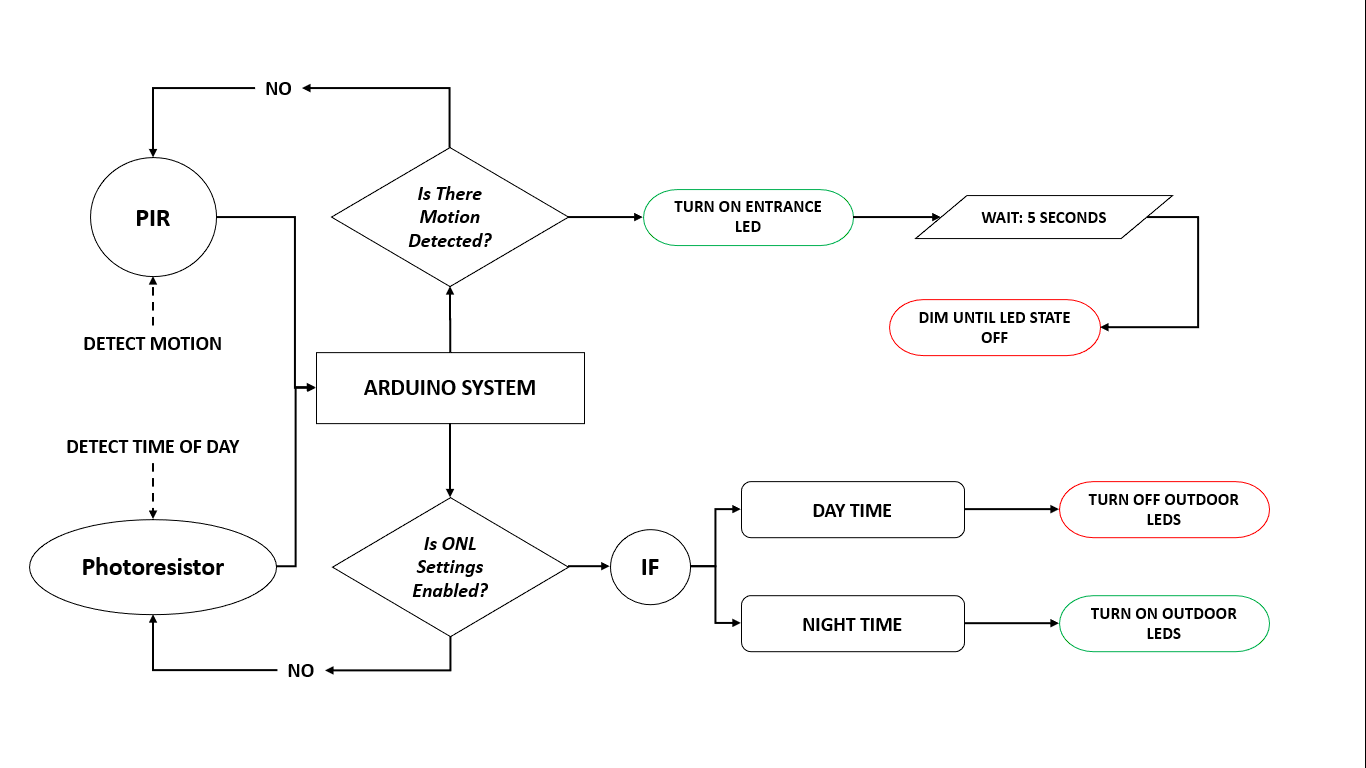
This flowchart illustrates how the prototype’s home automation system will work. The design will involve four phases: initial phase, connection phase, communication phase, and the execution phase. In the initial phase, as the connection between the mobile phone and the HC-05 Bluetooth module will be through Bluetooth connection, the mobile application to be created will be Bluetooth-based which will initially check if the device’s Bluetooth is enabled and is paired with the Bluetooth module before responding to connection requests. The connection phase, on the other hand, is the phase where the actual Bluetooth connection will be established to allow communication between the mobile device and the Bluetooth module. After connection has been successfully established, the mobile device can now send specific commands from the application to the HC-05 Bluetooth module which will transfer these commands to the Arduino system. The final phase is the execution phase where commands transmitted to the Arduino system will be identified and executed until disconnection is prompted.

Communication Phase

This flowchart illustrates an in-depth explanation regarding the transmission of commands during the communication phase. After the connection has been established, the mobile application will then wait until user input has been detected. The said input will be identified and processed as a specific command input that will be sent to the HC-05 Bluetooth module via Bluetooth socket stream. After receiving the command input, the module will then hand it over to the Arduino system which will then identify, translate, and execute this command using its algorithm. The system will also send the command input back to the mobile application via the Bluetooth socket stream, which will then be logged in the mobile application’s log monitor for debugging purposes.

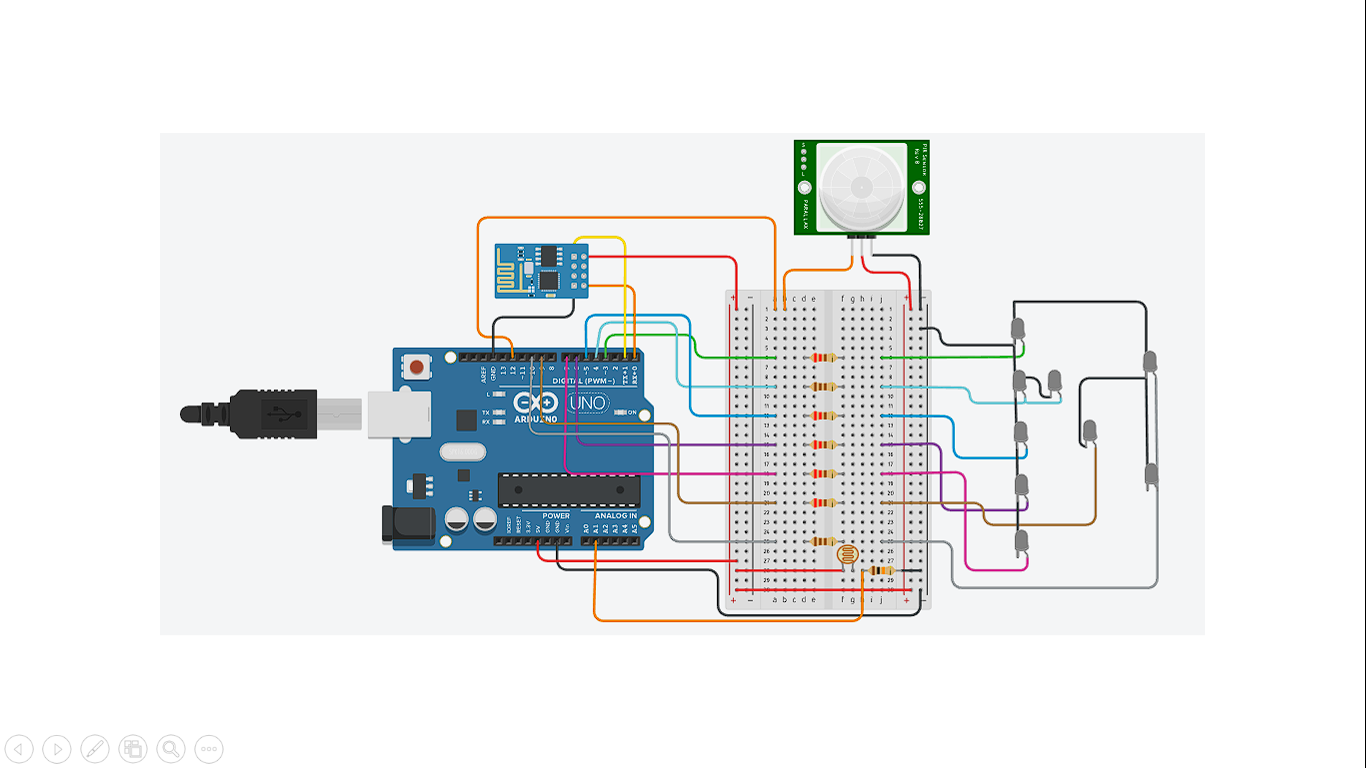
Arduno System Algorithm: Execution of Translated Commands

This flowchart explains how the Arduino system’s algorithm works in terms of identifying what command it received and executing them based on the nature of the command or the current state of the LED light that corresponds to it. There are eight programmed command translations: Living room, Dining room, Bed room, Bath room 1, Bath room 2, ONL settings, Turn off all lights, and Disconnect. The first six commands are toggleable depending on its corresponding state. Turn off all lights command will turn off all LED lights regardless of their states, and will also turn the ONL settings off. Finally, the Disconnect command will terminate the Bluetooth connection that was made between the mobile device and the HC-05 Bluetooth module, disallowing any further commands from the mobile applications from running.

Arduino System Algorithm: Entrance Light & ONL Photoresistor

This flowchart illustrates the different inputs that will be received by the Arduino system aside from commands received from the connected mobile application. The PIR motion sensor component will serve as an input signal after detecting motion that will automatically turn on the entrance light of the house prototype. However, the Arduino system will also be automatically turning it off after some time while also adding a dim effect before the actual turning off of the LED light. Meanwhile, the photoresistor will continuously detect the current time of the day and notifies the Arduino system through input signals that functions like an update. When the system received an update that the photoresistor has detected night time, outdoor LED lights will be automatically turned on, unless ONL settings are currently disabled. However, during day time, these LED lights will be automatically turned off regardless of ONL settings because of the photoresistor.

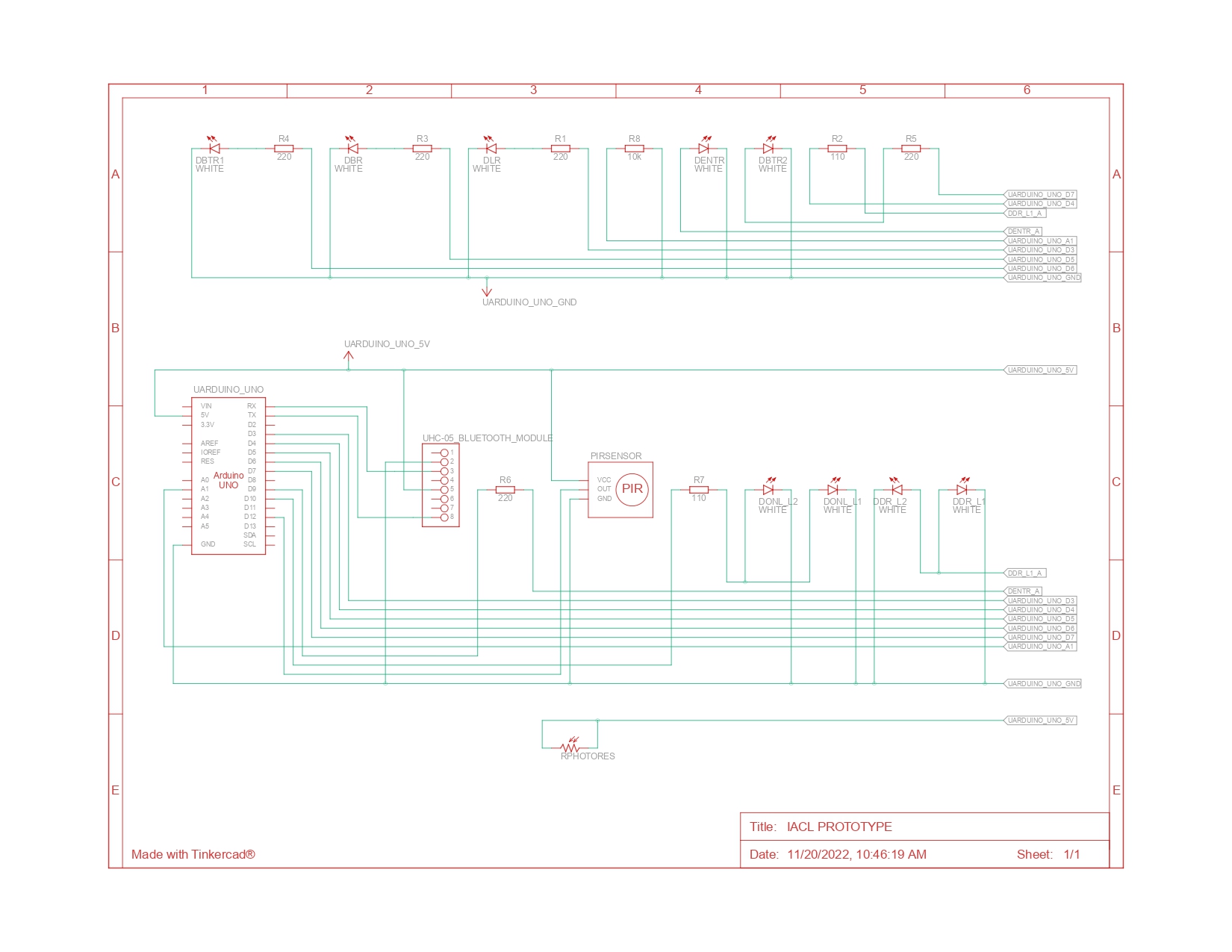
1. **Designing the prototype’s circuit.**

Circuit Diagram

The circuit diagram was designed using Tinkercad, an online website where users can design their own Arduino-based circuits. The summary of components used are as follows:

|  |  |  |
| --- | --- | --- |
| **Image** | **Component** | **Quantity** |
|  | Arduino Uno R3 | 1 |
| Discrete Devices : Resistor Through-Hole 220 ohm 5 1/4W Pack ... | 220 Ω Resistor | 5 |
| Carbon Film Resistors 5% 1W 110 Ohms – Electronix Express | 110 Ω Resistor | 2 |
| Resistor 10K Ohm 5% 1/4W (25-Pack) - ProtoSupplies | 10 kΩ Resistor | 1 |
| 5mm White LED – Mindsets Online | White LED | 9 |
| SunRobotics LDR Photoresistor Light Sensitive Resistor Light Dependent  Resistor 10mm 5 pcs : Amazon.in: Industrial & Scientific | Photoresistor | 1 |
| Breadboard-friendly Mini PIR Motion Sensor with 3 Pin Header : ID 4871 :  $3.95 : Adafruit Industries, Unique & fun DIY electronics and kits | PIR Sensor | 1 |
| HC-05 HC-06 Bluetooth Wireless RF-Transceiver module RS232 Serial TTL –  AZ-Delivery | HC-05 Bluetooth Module | 1 |

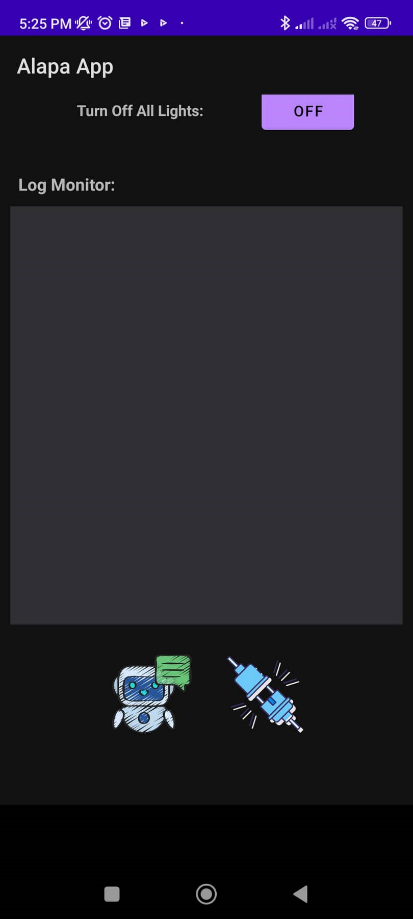
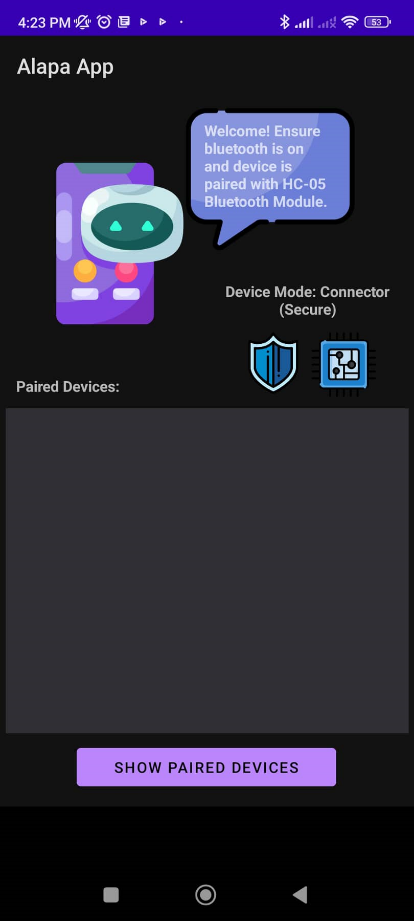
Schematic Diagram



The following diagram illustrates how wiring connections were made. The summary are as follows:

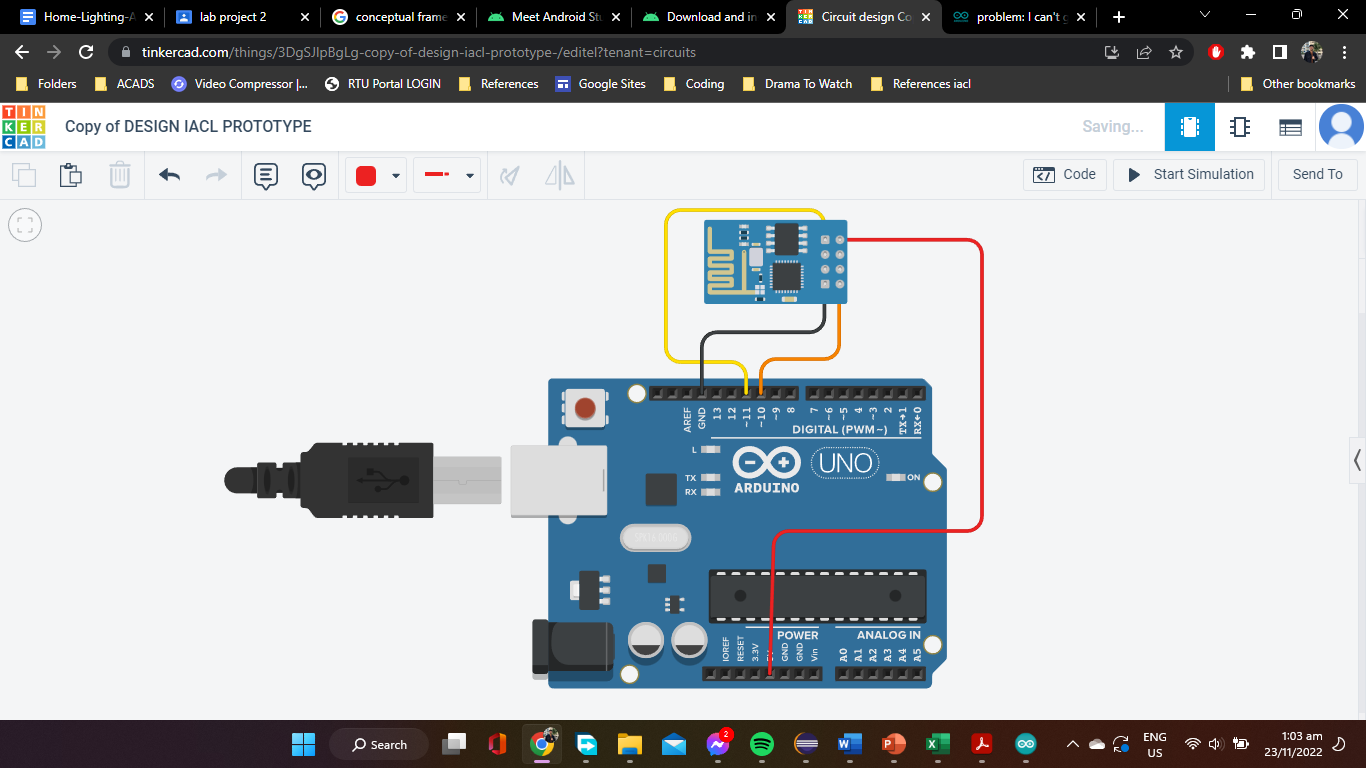
|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Component Name** | **Wiring Connection** | |
| **Component part/s** | **Connected to:** |
| Arduino Uno R3 | UArduino Uno | Power port | a 9V battery |
| 220 Ω Resistor | R1 (Living Room) | Left-end | UArduino Uno pin 3 |
| Right-end | DLR anode |
| R3 (Bed Room) | Left-end | UArduino Uno pin 5 |
| Right-end | DBR anode |
| R4 (Bath Room 1) | Left-end | UArduino Uno pin 6 |
| Right-end | DLR anode |
| R5 (Bath Room 2) | Left-end | UArduino Uno pin 7 |
| Right-end | DBTR1 anode |
| R6 (Entrance) | Left-end | UArduino Uno pin 9 |
| Right-end | DENTR anode |
| 110 Ω Resistor | R2 (Dining Room) | Left-end | UArduino Uno pin 4 |
| Right-end | DDR L1 & DDR L2 anodes |
| R7 (Outdoor Night Lights) | Left-end | UArduino Uno pin 10 |
| Right-end | DONL L1 & DONL L2 anodes |
| 10 kΩ Resistor | R8 | Left-end | RphotoRes right-end & UArduino Uno pin A1 |
| Right-end | UArduino Uno GND |
| Photoresistor | RphotoRes | Left-end | UArduino Uno 5V |
| Right-end | R8 left-end & UArduino Uno pin A1 |
| White LED | DLR | Anode | R1 right -end |
| Cathode | UArduino Uno GND |
| DDR L1 | Anode | R2 right-end |
| Cathode | UArduino Uno GND |
| DDR L2 | Anode | R2 right-end |
| Cathode | UArduino Uno GND |
| DBR | Anode | R3 right-end |
| Cathode | UArduino Uno GND |
| DBTR1 | Anode | R4 right-end |
| Cathode | UArduino Uno GND |
| DBTR2 | Anode | R5 right-end |
| Cathode | UArduino Uno GND |
| DENTR | Anode | R6 right-end |
| Cathode | UArduino Uno GND |
| DONL L1 | Anode | R7 right-end |
| Cathode | UArduino Uno GND |
| DONL L2 | Anode | R7 right-end |
| Cathode | UArduino Uno GND |
| PIRsensor | PIR Sensor | Power | UArduino Uno 5V |
| Ground | UArduino Uno GND |
| Output | UArduino Uno pin 12 |
| HC-05 Bluetooth Module | UHC-05 Bluetooth Module | RX | UArduino Uno TX |
| TX | UArduino Uno RX |
| Power | UArduino Uno 5V |
| Ground | UArduino Uno GND |

1. **Creating a mobile application.**

In order to create a mobile application, the researchers have used Android Studio, the official Integrated Development Environment (IDE) for Android app development. It was developed and distributed by Google, containing the essential tools and environment for android programming. The prototype’s mobile application was designed to have the basic functions of connecting to another device and sending commands through Bluetooth connection, as the Arduino board can only be accessed wirelessly through means of Bluetooth connection with the help of a HC-05 Bluetooth module. The app’s user interface (UI) design is relatively simple, containing relevant texts, buttons, and list views. The images below present the app’s UI:

1. **Configuring HC-05 Bluetooth Module.**

The researchers have configured the settings of the HC-05 Bluetooth module in order to designate its purpose to only accommodate the prototype. In order to gain access to the module’s settings, the researchers used AT commands. The steps the researchers followed to access its AT commands are as follows:

Step 1: Connecting the wirings of the circuit.

The RX pin of the HC-05 Bluetooth module was connected to the Arduino Uno’s pin 11, while its TX pin was connected to pin 11. Meanwhile, its ground pin was connected to the Arduino Uno’s GND, while its power pin was connected to the board’s 5V pin.

Step 2: Connect the Arduino to pc or laptop.

Before connecting the Arduino Uno, the researchers removed the wire connecting the HC-05 bluetooth module’s power pin to the board’s 5V pin. This is to ensure that the power going into the Arduino doesn’t automatically start-up the module. After connection, the researchers copied the following codes to the Arduino Software and deployed them to the Arduino board:

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

void setup() {

pinMode(9, OUTPUT); /\* this pin will pull the HC-05 pin 34 (KEY pin) HIGH to switch module to AT mode \*/

digitalWrite(9, HIGH);

Serial.begin(38400);

Serial.println("Enter AT Commands:");

BTSerial.begin(38400); // HC-05 default speed in AT command mode

}

void loop() {

//The code below allows for commands and messages to be sent from COMPUTER (serial monitor) -> HC-05

if (Serial.available()) // Keep reading from Arduino Serial Monitor

BTSerial.write(Serial.read()); // and send to HC-05

//The code below allows for commands and messages to be sent from HC-05 -> COMPUTER (serial monitor)

if (BTSerial.available()) // Keep reading from HC-05 and send to Arduino

Serial.write(BTSerial.read()); // Serial Monitor

}

This will configure the algorithm of the board to access the HC-05 Bluetooth module’s AT Commands. Afterwards, the researchers held the button beside the pins of the Bluetooth module and finally connected the aforementioned wire earlier to its designated 5V pin before releasing the button.

Step 3: Using AT Commands

The researchers accessed the AT commands of the Bluetooth module through the Arduino’s Serial Monitor. Before typing in the commands, the researchers ensured that the Serial Monitor’s configuration was set to “Both NL & CR” and its baud rate was set to 38400 bauds. The following are the commands typed by the researchers into the Serial Monitor:

AT

AT+NAME=IACL ARDUINO

AT+PSWD=IACLJJRR

AT+RESET

Afterwards, the researchers exited the Serial Monitor and restarted the Arduino circuit but connecting its power port to a 9V battery with the wiring connections between the Bluetooth module’s power pin and the board’s 5V pin connected.

1. **Designing the house prototype model.**

The inspiration behind the design of the house prototype model was acquired from modern house designs with modern color palettes. The house includes 3 enclosed rooms, 2 open rooms, and 1 entrance room. LED lights are installed both indoors and outdoors, with 7 lights inside and 2 lights outside. A PIR sensor is placed above the entrance door, with the aim of detecting door movement and serving as a trigger for automatic entrance room lighting. Meanwhile, a photoresistor is installed on the roof to detect the presence and absence of photons which serve as the trigger for automatic outdoor lights to turn on during night time. Each room inside the house has 1 light installed, with the exception of the dining room, where 2 lights are installed instead of only 1. The prototype’s interior design and furniture were designed to complement the modern exterior of the house. The images below show the initial stages to final output of the house design:

Initial construction of house prototype

**A picture containing indoor, door, white, chest of drawers

Description automatically generated**A picture containing text, indoor

Description automatically generated

Final Output









**Timetable**

|  |  |  |
| --- | --- | --- |
| **WEEK** | **DATE** | **TASK** |
| **1ST Week** | October 3-7, 2022 | 1. Research for different case study regarding Instrumentation and Design 2. Distribution of Tasks |
| **2ND Week** | October 10-14, 2022 | 1. Canvassing of Materials 2. Collection for funds |
| **3RD Week** | October 17-21, 2022 | 1. Testing and completion of materials 2. Start of programming codes 3. Start making of input drafts for paper |
| **4th Week** | October 24-28, 2022 | Start of building the miniature |
| **5th Week** | October 31- Nov. 4, 2022 | Start doing the wirings for the Prototype |
| **6th Week** | November 7-11, 2022 | Adjusting and adding some materials |
| **7th Week** | November 14-18, 2022 | Finalizing the Prototype, testing and paper |
| **Final Week** | November 21-25, 2022 | 1. Final Testing and checking of all inputs and prototype 2. Prototype Presentation |

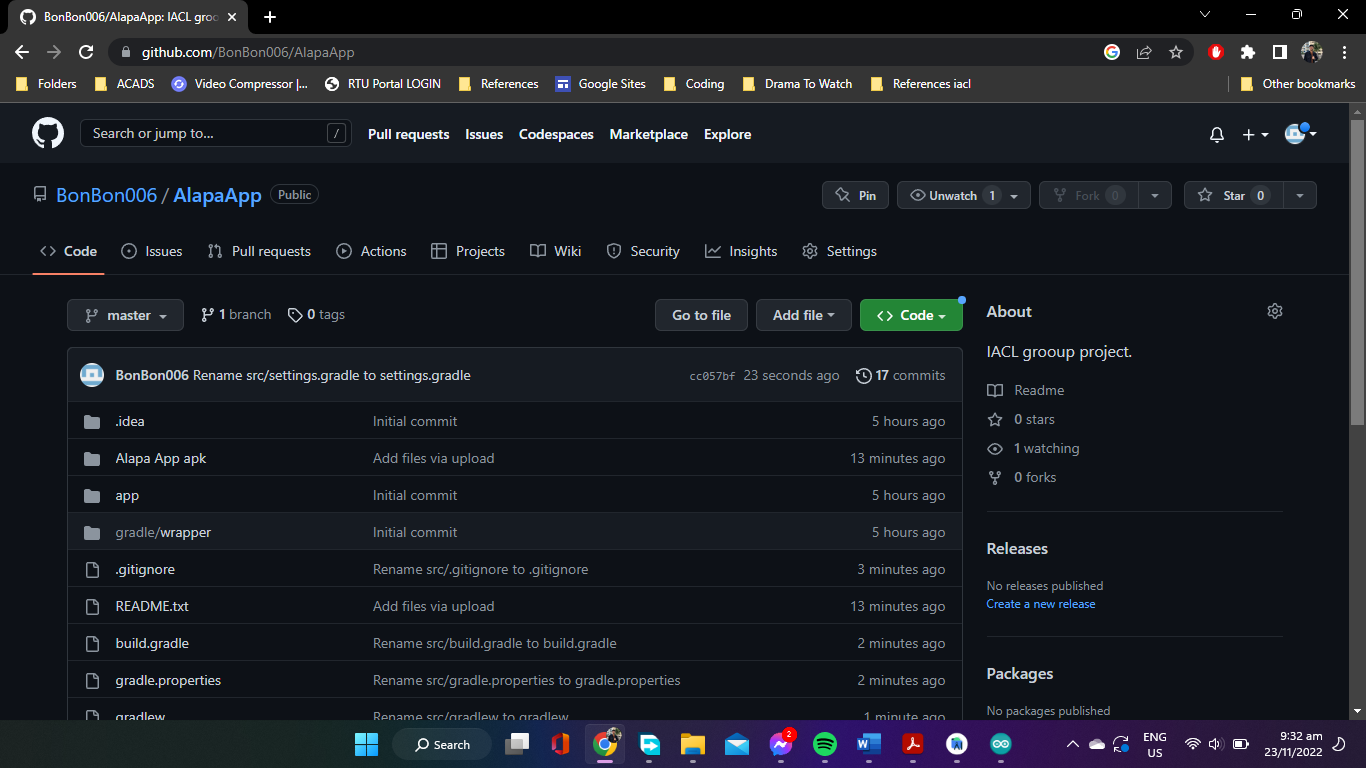
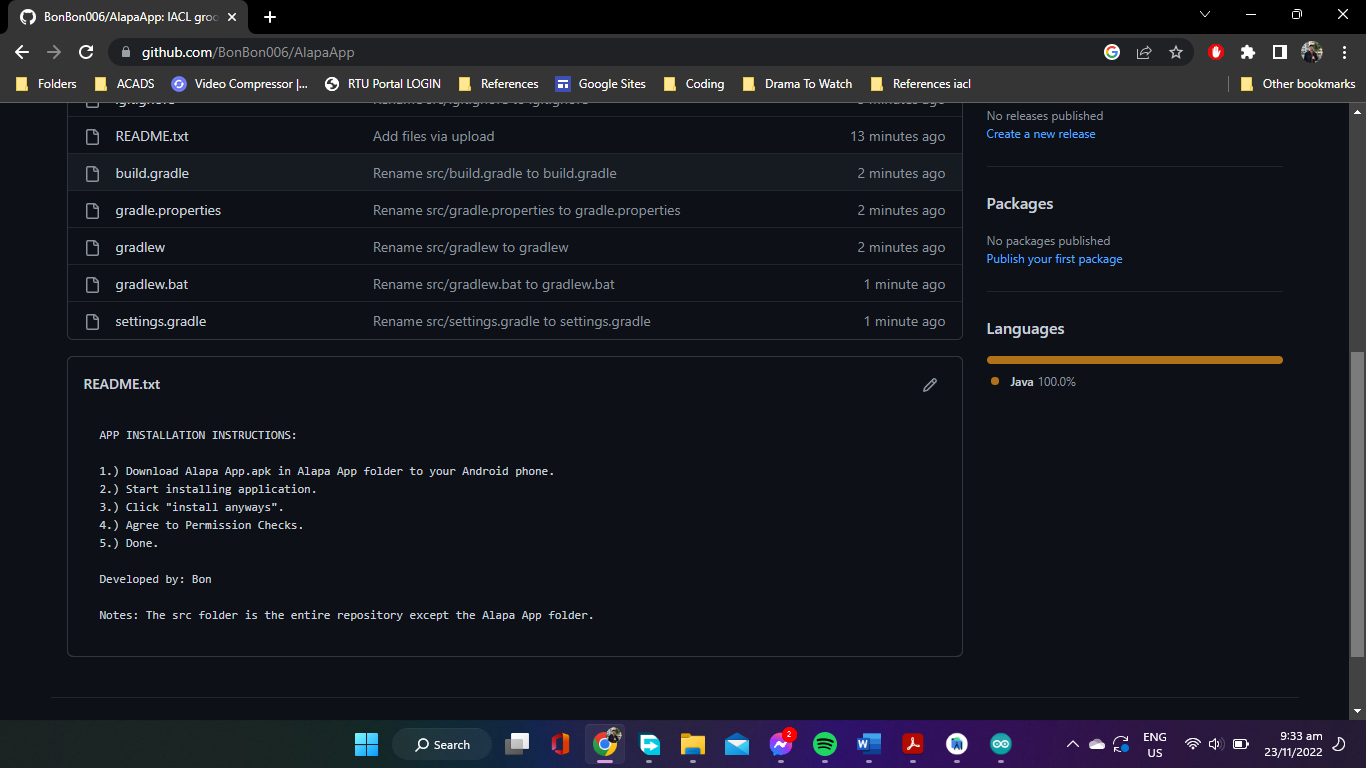
**Cost of Materials**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cost of Materials** | | | |
| **SI No.** | **Name of the item** | **Qty** | **Price** |
| **1** | Arduino Uno Kit | 1 | 715 |
| **2** | ***Tools and Equipment*** |  |  |
|  | Glue Gun Switch | 1 | 165 |
|  | Soldering Wire | 1 | 180 |
|  | Utility Knife | 3 | 164 |
|  | Cutting Board | 1 | 88 |
|  | Ruler Aluminum | 1 | 14 |
| **3** | ***Raw Materials*** |  |  |
|  | Sintra Board 3mm | 3 | 294 |
|  | Sintra Board 2mm | 3 | 345 |
|  | Illustration Board 1/8 | 2 | 36 |
|  | Envelope Clear | 1 | 33 |
|  | Clear Folder | 3 | 39 |
|  | Worx Fine Paper Black | 1 | 32 |
|  | Styro foam 1 inch | 1 | 26 |
|  | Styro foam 1/2 inch | 2 | 38 |
|  | Grass Sheet | 1 | 25 |
|  | Popsicle Stick | 3 | 75 |
|  | Worx Board White | 1 | 26 |
|  | Felt Pad Paper | 2 | 38 |
|  | Poster Paint | 4 | 152 |
|  | Paint Brushes | 3 | 109 |
|  | Mediplast Pad 3x3 | 3 | 15 |
|  | Adhesive Glue | 30 | 400 |
|  | Glue Stick | 2 | 32 |
|  | Special Paper | 2 | 70 |
| **4** | ***Electronics*** |  |  |
|  | 9V Battery | 2 | 376 |
|  | PIR Sensor | 2 | 20 |
|  | LED | 10 | 50 |
|  | **Total** | | **4016** |

**Program Codes**

Mobile Application: Alapa App

Due to the nature of Android Studio, the codes related to the application cannot be directly posted to a file as codes will not function without the entire files involved in the app are present. Therefore, the developer of the program has uploaded the source codes to GitHub, an online software development platform for developers. The mobile application can also be downloaded through the developer’s GitHub.

GitHub link: <https://github.com/BonBon006/AlapaApp.git>

Arduino System

// living room -> pin 3 toggle command = '1'

// dining room -> pin 4 toggle command = '2'

// bed room -> pin 5 toggle command = '3'

// bath room 1 -> pin 6 toggle command = '4'

// bath room 2 -> pin 7 toggle command = '5'

// entrance light -> pin 9 pir sensor triggered

// outdoor night lights -> pin 10 photo resistor triggered / toggle command = '6'

// pir sensor -> pin 12

// photo resistor -> pin A1

// turn of all lights -> toggle command = '7' ; turns off all ON lights and toggles ONL to OFF MODE.

// note: if TOAL is toggled, re-toggle ONL to reset to initial settings with ONL "ON"

const int LED[] = {

3, 4, 5, 6, 7, 9, 10

};

const char COMMANDS[] = {

'1', '2', '3', '4', '5', '6', '7'

};

const int PIR = 12;

const int photoRes = A1;

int pirStateCheck;

int timeDifference\_Pir;

int timeDifference\_Dim;

int brightness;

int photoResStateCheck;

int outdoorNightLightsSettings;

const unsigned long shortDelayEvent = 30;

const unsigned long longDelayEvent = 3000;

unsigned long currentTime;

unsigned long pirTimeInterval;

unsigned long dimTimeInterval;

char commandRead = '26';

int pirLedState;

void setup()

{

Serial.begin(9600);

for (int i = 0; i < 7; i++) {

pinMode(LED[i], OUTPUT);

}

pinMode(PIR, INPUT);

pinMode(photoRes, INPUT);

outdoorNightLightsSettings = HIGH;

pirLedState = LOW;

}

void loop()

{

currentTime = millis();

pirStateCheck = digitalRead(PIR);

if (pirStateCheck == HIGH){

pirSensorLoop();

}

currentTime = millis();

timeDifference\_Pir = currentTime - pirTimeInterval;

if ((timeDifference\_Pir >= longDelayEvent && pirLedState == HIGH)){

dim();

digitalWrite(LED[5], LOW);

pirLedState = LOW;

}

if (Serial.available()) {

commandRead = Serial.read();

Serial.println(commandRead);

serialMonitorCommand();

}

outdoorLighting();

}

void pirSensorLoop()

{

currentTime = millis();

pirTimeInterval = millis();

digitalWrite(LED[5], HIGH);

pirLedState = HIGH;

}

void dim()

{

brightness = 255;

dimTimeInterval = millis();

while (brightness > 0){

currentTime = millis();

timeDifference\_Dim = currentTime - dimTimeInterval;

if (timeDifference\_Dim >= shortDelayEvent){

analogWrite(LED[5], brightness);

dimTimeInterval = millis();

brightness -= 5;

}

}

}

void serialMonitorCommand()

{

int i = 0;

while (commandRead != COMMANDS[i]) {

i++;

}

if (i < 5) {

digitalWrite(LED[i], !digitalRead(LED[i]));

}

else if (i == 5) {

toggleNightLights();

}

else {

turnOffAllLights();

}

}

void outdoorLighting()

{

photoResStateCheck = analogRead(photoRes);

if (photoResStateCheck < 900 && outdoorNightLightsSettings == HIGH){

digitalWrite(LED[6], HIGH);

}

else if (photoResStateCheck >= 900 || outdoorNightLightsSettings == LOW){

digitalWrite(LED[6], LOW);

}

}

void disableNightLights()

{

outdoorNightLightsSettings = LOW;

digitalWrite(LED[6], LOW);

}

void enableNightLights()

{

outdoorNightLightsSettings = HIGH;

outdoorLighting();

}

void toggleNightLights()

{

if (outdoorNightLightsSettings == HIGH){

disableNightLights();

}

else if (outdoorNightLightsSettings == LOW){

enableNightLights();

}

}

void turnOffAllLights()

{

for (int a = 0; a < 7; a++) {

digitalWrite(LED[a], LOW);

disableNightLights();

}

}

**User Manual**

This user manual contains general and specific instructions that will guide the user in using “Alapa App”, an Android application created for the sole purpose of controlling the prototype’s LED lights as well as configuring its illumination settings. The manual only includes how to use the prototype and not how to build it.

1. **Application Components**

**Icon List**

|  |  |  |
| --- | --- | --- |
| Image | Name | Description |
|  | RFCOMM | To toggle RFCOMM channels. Most of the time for debugging purposes only. |
|  | Device Mode | To toggle Device Mode. Most of the time for debugging purposes only. |
|  | Developer Notes | To open Developer Notes containing a guide to Log Monitor Commands. Used for debugging. |
|  | Disconnect | To disconnect an established connection with a currently connected device. |

**Button List**

|  |  |  |
| --- | --- | --- |
| Image | Name | Description |
|  | Show Paired Devices | To show paired devices for establishing connection. |
|  | Living Room | To toggle living room light ON/OFF. |
| Dining Room | To toggle dining room lights ON/OFF. |
| Bed Room | To toggle bed room light ON/OFF. |
| Bath Room 1 | To toggle bath room 1 light ON/OFF. |
| Bath Room 2 | To toggle bath room 2 light ON/OFF. |
| Outdoor Night Lights | To toggle ONL Settings ON/OFF. |
| Turn Off All Lights | A universal command which turns of all lights and turns ONL Settings OFF. |

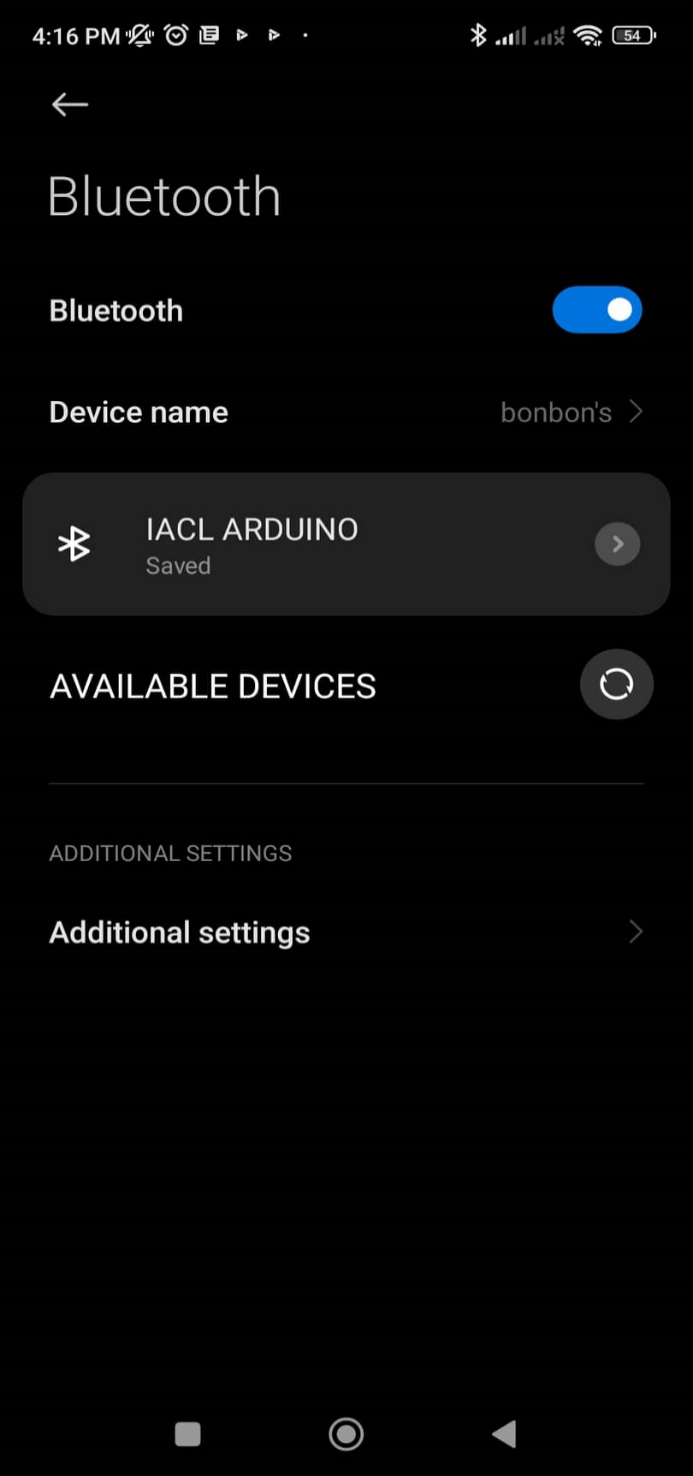
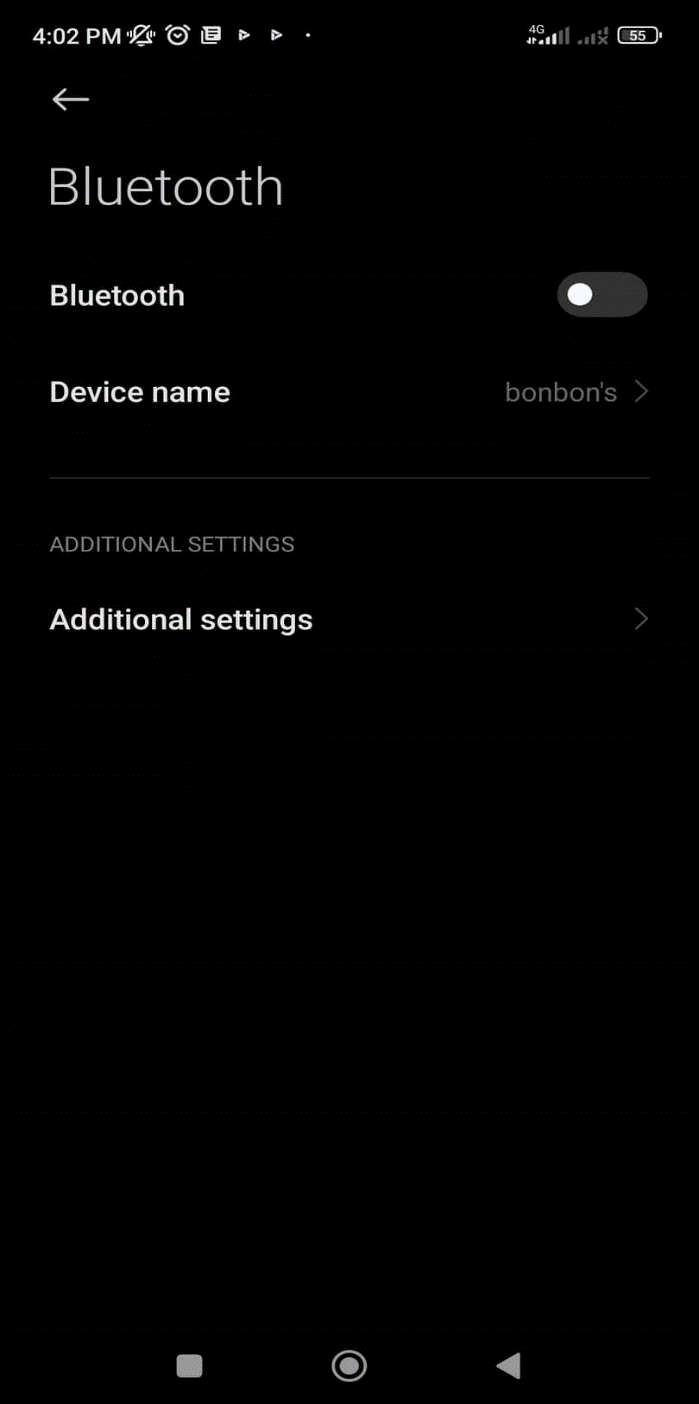
**View Box List**

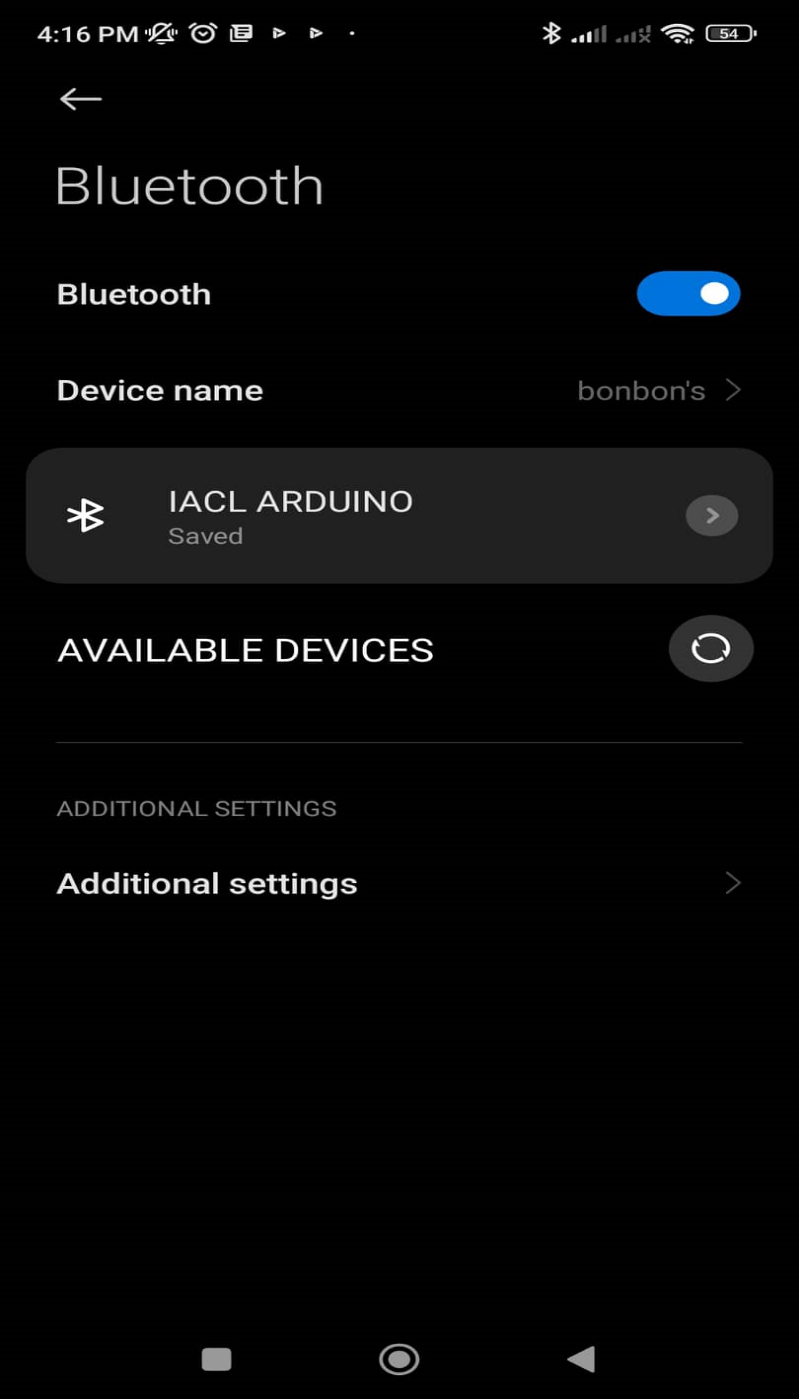
|  |  |  |
| --- | --- | --- |
| Image | Name | Description |
|  | Paired Devices List | Lists and displays your device’s paired devices. |
|  | Log Monitor | Displays Log Commands. For debugging purposes. |

1. **Tutorials**

**Instructions on How to Connect with HC-05 Bluetooth Module**

Connecting with the HC-05 Bluetooth Module is the first important step in using the prototype. Several buttons are also initially disabled until connection with the module has been established. This is because commands from the application will not be remotely received by the Arduino Uno if connection has not been made, resulting to both the application and the house prototype being unresponsive.

**Step 1: Enable device Bluetooth**

  
This step is important. Bluetooth must be enabled or turned on during the whole duration of using the application, otherwise the device will not be able to connect with the prototype’s HC-05 Bluetooth Module. In an event that the device’s Bluetooth was disabled post-connection phase, controlling the lights will not be possible anymore.

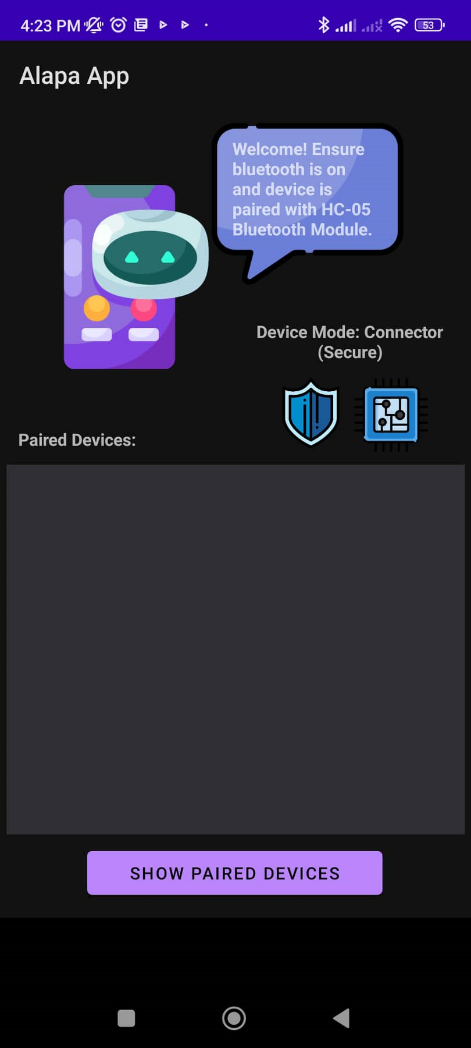
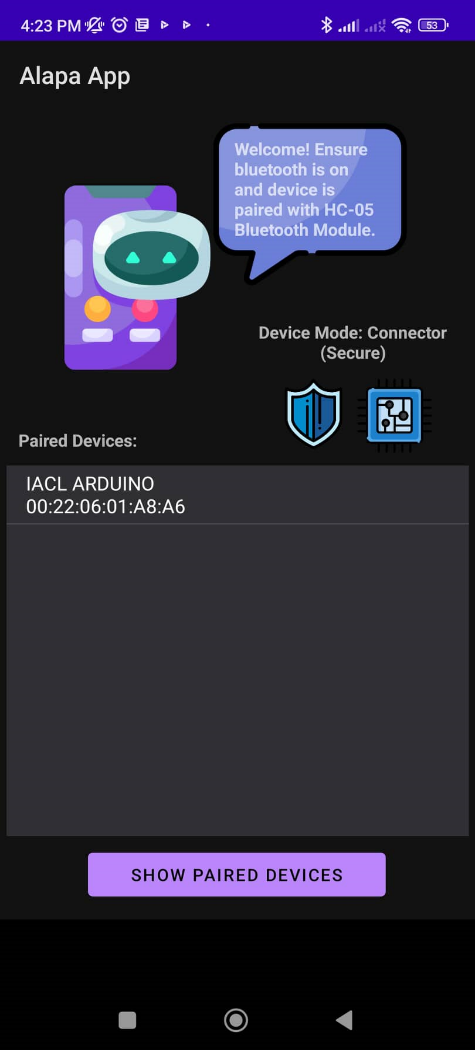
**Step 2: Pair Device with HC-05 Bluetooth Module**

Another important step is pairing the device with the prototype’s HC-05 Bluetooth Module. In order to do so, the user must navigate to their Bluetooth settings which is usually found in the device’s Settings. Establishing connections with the Bluetooth Module will not be possible without first pairing with it.

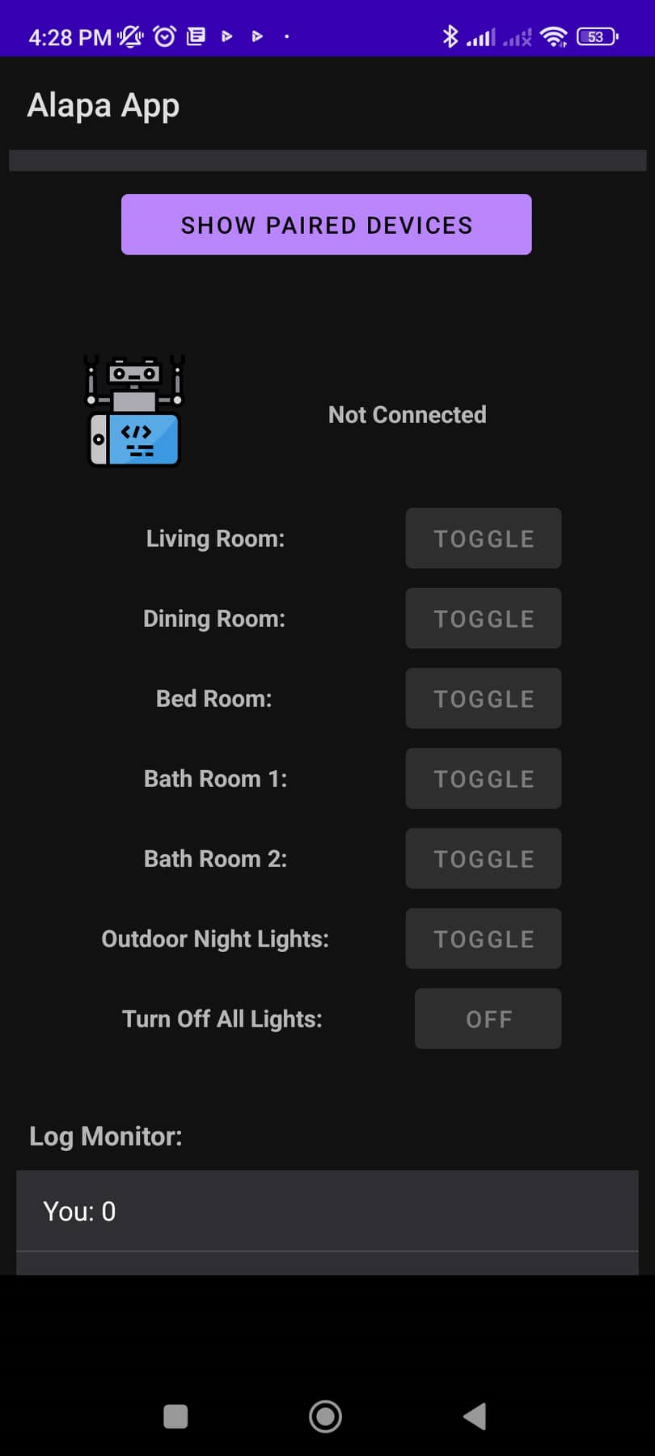
In order to pair with the module, the user must scan for new devices and wait until the name “IACL ARDUINO” pops up on the screen. The user must then tap the said name and type the corresponding password to pair. The following are the HC-05 Bluetooth Module’s information:

**Name:** IACL ARDUINO

**Password:** IACLJJRR

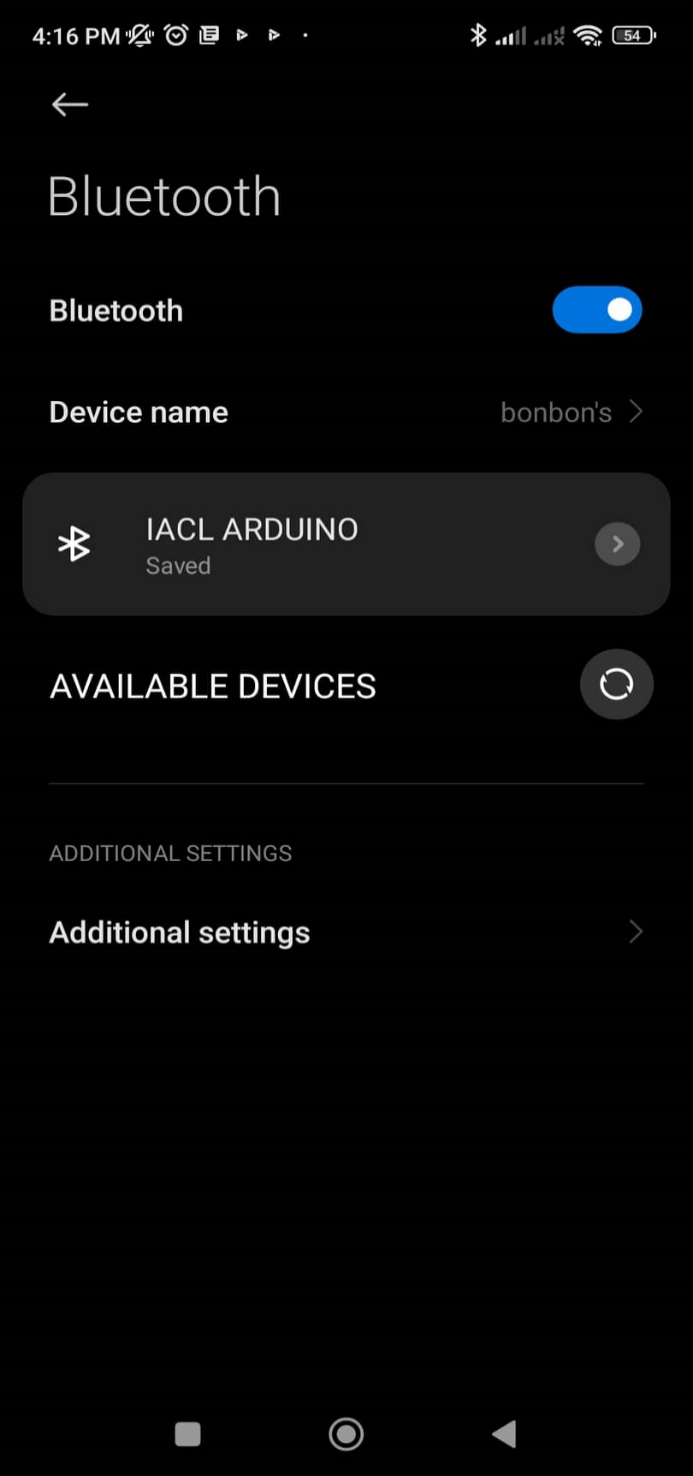
**Step 3: Connect with the HC-05 Bluetooth Module**

Upon ensuring the first two steps are done, open Alapa App. Initially, the application’s control buttons as well as disconnect icon will be disabled automatically until a connection is created. To connect, the user must tap the button named “SHOW PAIRED DEVICES” to display the device’s paired devices. This will show the prototype’s HC-05 Bluetooth Module’s name, “IACL ARDUINO”, with its corresponding address. The user must tap it and wait for a short notification at the bottom side of the screen. If connection was made, the application will notify the user saying “Connected To: IACL ARDUINO”, while on the other hand, saying “Connection failed” upon failing to connect with the Bluetooth Module.

Upon connecting with it, the button for showing paired devices will be disabled to disallow the user from connecting to another device with the same secure or insecure UUID or from creating connection issues with the Bluetooth Module post-connection phase. At this phase, the application’s control buttons will now be enabled, as the application has now detected a connection with the module. From this point onwards, the user is now able to control the prototype’s lights, toggling them on or off, or configure specific lights settings.

**How to Toggle Indoor Lights ON/OFF**

Indoor lights pertain to the toggleable LED lights inside the house prototype, excluding the entrance light, which is automatically turned on by motion sensor inputs and turns off after a few seconds. Indoor lights are as follows: *Living Room*, *Dining Room*, *Bed Room*, *Bath Room 1*, and *Bath Room 2*.

**Step 1: Ensure Bluetooth is on and connection with HC-05 Bluetooth Module has been established.**

Access to the designated buttons that can toggle the lights on or off is disabled until connection with the HC-05 Bluetooth Module has been established. Moreover, device Bluetooth MUST always be on in order for commands from the application to reach the Arduino system.

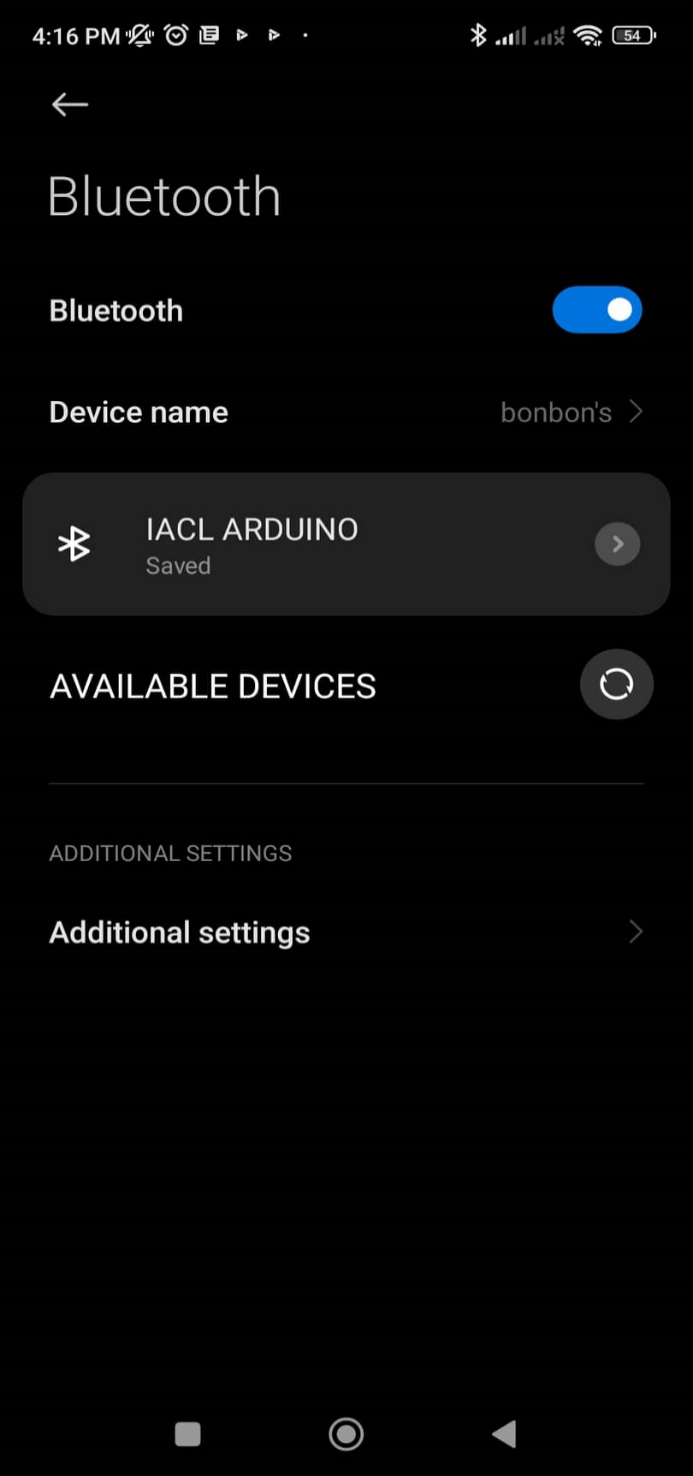
**Step 2: Tap the corresponding button of your choice.**

Upon deciding on which room to toggle light/s with, tap the button which corresponds to your choice. The prototype’s Arduino system has been programmed to toggle the room of your choice on or off, depending on its current lighting state.

**How to Toggle Outdoor Night Lights Setting**

The ONL settings is used for enabling or disabling the outdoor night lights’ automatic lighting algorithm whenever a photoresistor attached to it detects that the surrounding area is experiencing night time. This setting was created for user convenience to minimize power consumption due to the constant lighting of outdoor lights due to its night-detecting system.

By default, ONL settings is set to ON.

**Step 1: Ensure Bluetooth is on and connection with HC-05 Bluetooth Module has been established.**

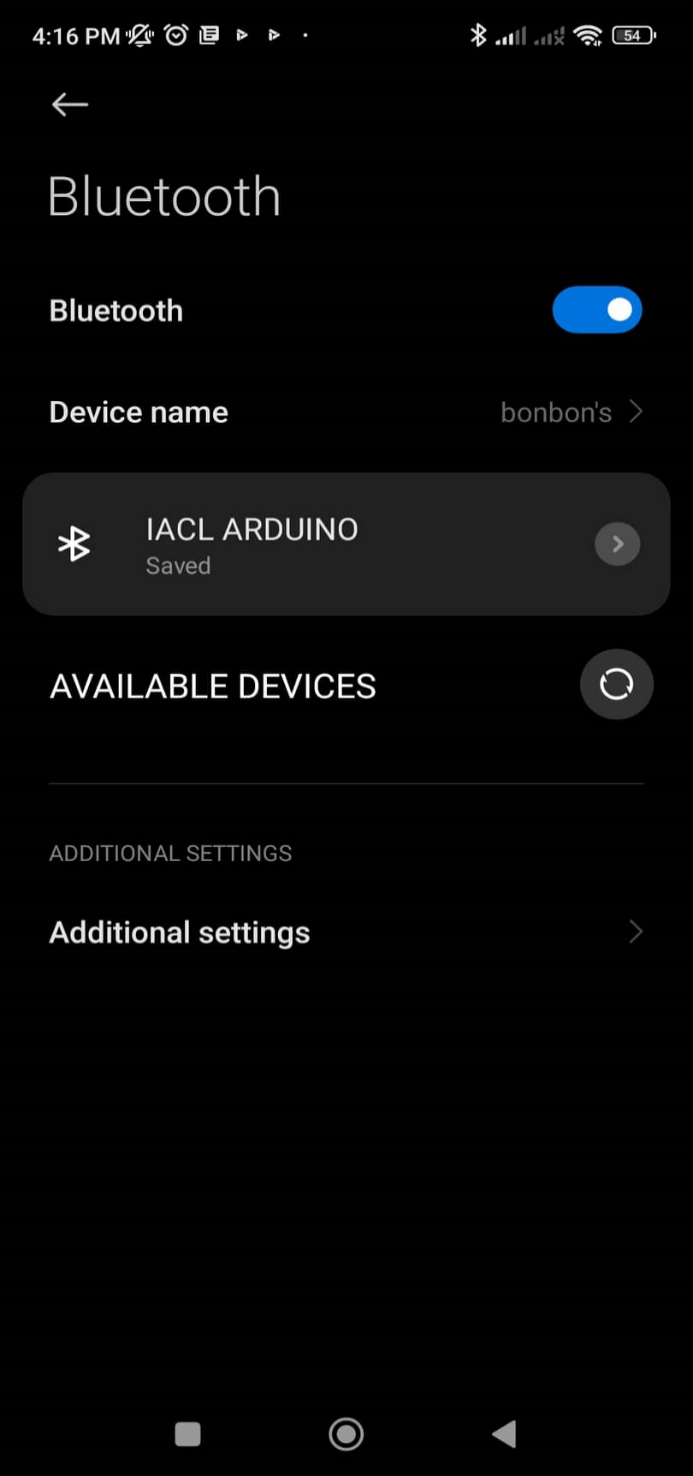
Access to the button for toggling the ONL settings is disabled until connection with the HC-05 Bluetooth Module has been established. Moreover, device Bluetooth MUST always be on in order for the command from the application to reach the Arduino system.

**Step 2: Tap the ONL Settings button.**Upon tapping, the algorithm of the outdoor night lights will be temporarily disabled until enabled by the user again or Arduino Uno has been restart

**How to Use Universal Command: Turn Off All Lights**

TOAL is a universal command that is used to disable all currently enabled indoor lights in the house prototype as well as disable ONL Settings. This feature was created for user convenience of turning off all lights, whether located indoors or outdoors.

**Note:** ONL Settings are disabled on tap. To enable it again, tap the ONL Settings button.

**Step 1: Ensure Bluetooth is on and connection with HC-05 Bluetooth Module has been established.**

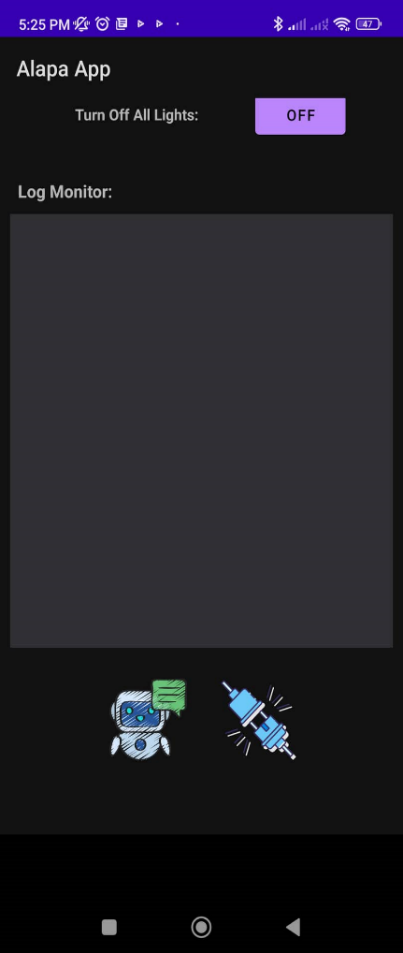
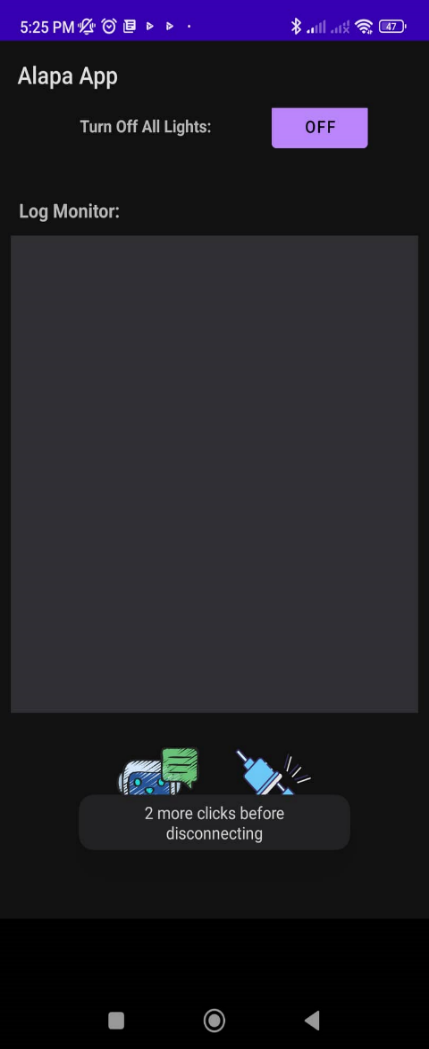
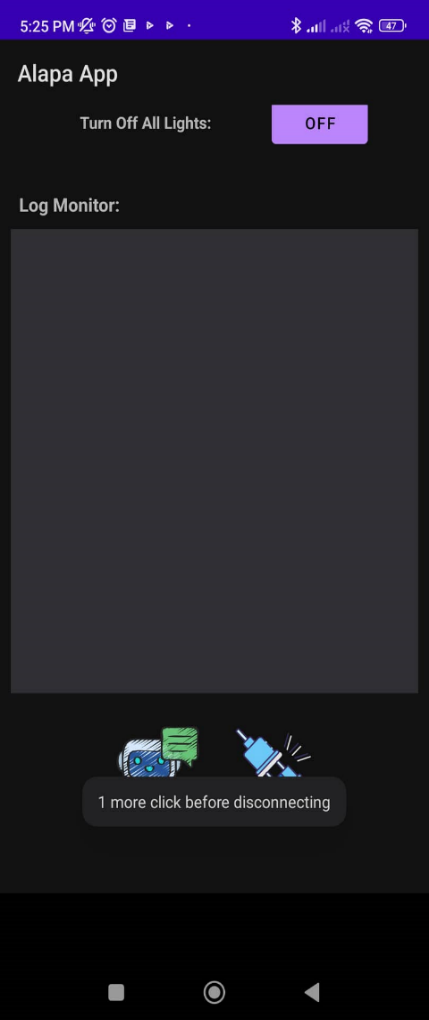
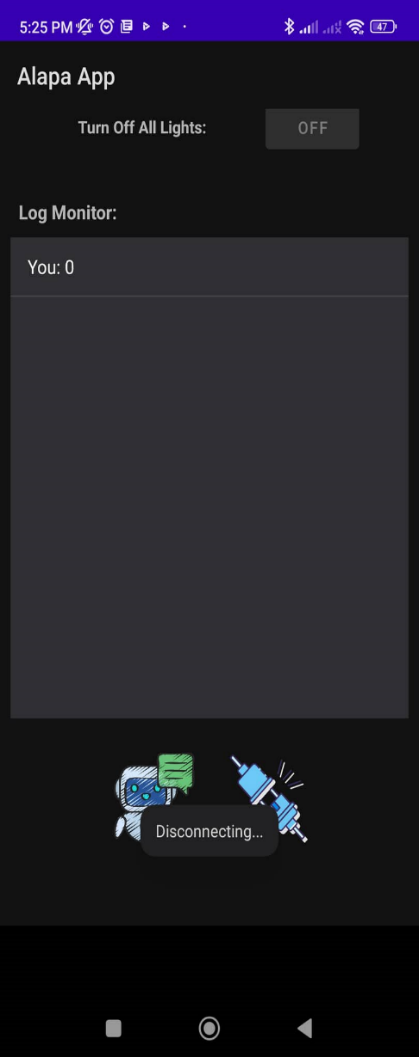
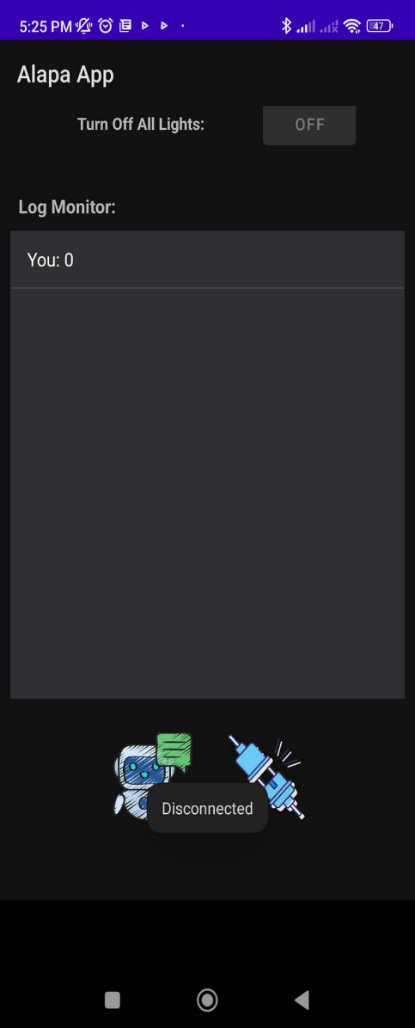
Access to the button for TOAL is disabled until connection with the HC-05 Bluetooth Module has been established. Moreover, device Bluetooth MUST always be on in order for the command from the application to reach the Arduino system.

**Step 2: Tap the TOAL button.**

This will disable all currently enabled lights indoors as well as outdoors. This also disables ONL Settings.

**How to Disconnect with HC-05 Bluetooth Module**

Before exiting the application, ensure that your device has been disconnected with the HC-05 Bluetooth Module as the created Bluetooth socket post-connection will still remain. This may cause issues in reconnecting with the module.

**Step 1: Disconnect with HC-05 Bluetooth Module**

In order to terminate the connection established with the module, the user must navigate downwards on the screen and finding the “Disconnect” icon. However, the user must tap the said icon thrice before the actual disconnection happens as the application is programmed to disallow accidental disconnection to happen. For each tap, the application will be displaying a short notification on the bottom of the screen saying, “\_\_ more click/s before disconnecting” until finally displaying, “Disconnecting...” when the actual disconnection starts.

1. **Component Functions**

**Connection Modes**

The toggleable icons involved can be ignored by the user as default settings will work most of the time. However, when connection failure persists even after following the connection steps properly, toggling them may be a solution.

***RFCOMM: Secure***

By default, RFCOMM is set to “Secure”, which uses a secure UUID when connecting or listening to other devices. This mode is the preferred RFCOMM mode as this prevents 3rd party interference or hacking. Although Bluetooth connections are usually secure, this improves security and is therefore the default RFCOMM settings at application start-up.

***RFCOMM: Insecure***

When RFCOMM is set to “Insecure”, the application will be using an insecure UUID when connecting or listening to other devices. This mode is not recommended as this can allow 3rd party interference or hacking. Although, when secure RFCOMMS do not work, toggling to insecure RFCOMM can be an option in fixing the issue.

***Device Mode: Connector***

By default, Device Mode is set to “Connector”. In Connector, the application will be the one to form a Bluetooth socket with a device to form a connection. When connecting to IACL ARDUINO, Device Mode must be set to Connector as the module itself does not have the capability of creating a Bluetooth socket.

***Device Mode: Listener***

For debugging purposes. This Device Mode sets the device to be the listener and open a Bluetooth Server Socket that will accept any device with the same UUID that is attempting to create a Bluetooth Socket with your device.

**Control Buttons**

***Indoor Lights: Toggle ON/OFF***

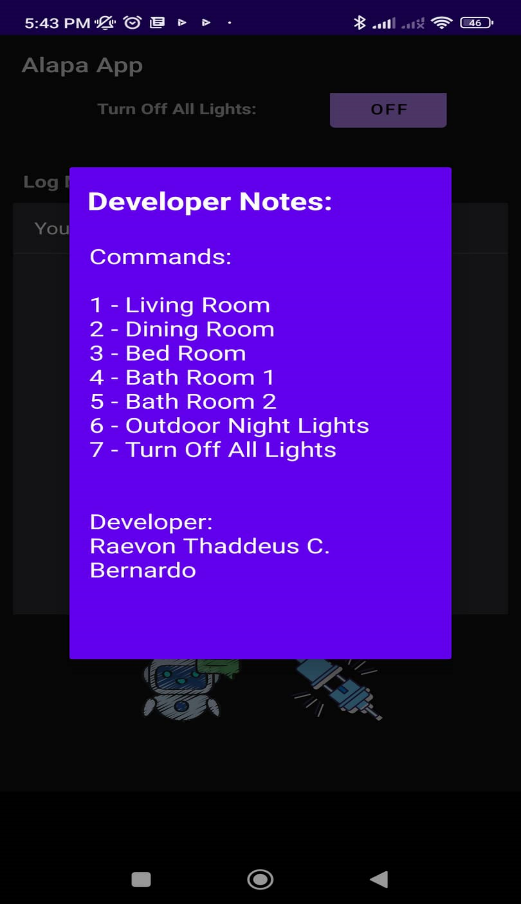
There are 5 indoor lighting buttons that can be toggled by the user via tapping: *Living Room, Dining Room, Bed Room, Bath Room 1,* and *Bath Room 2*. Tapping one of these buttons will toggle the corresponding room’s light/s ON or OFF.

***Outdoor Night Lights: Toggle ONL Settings***

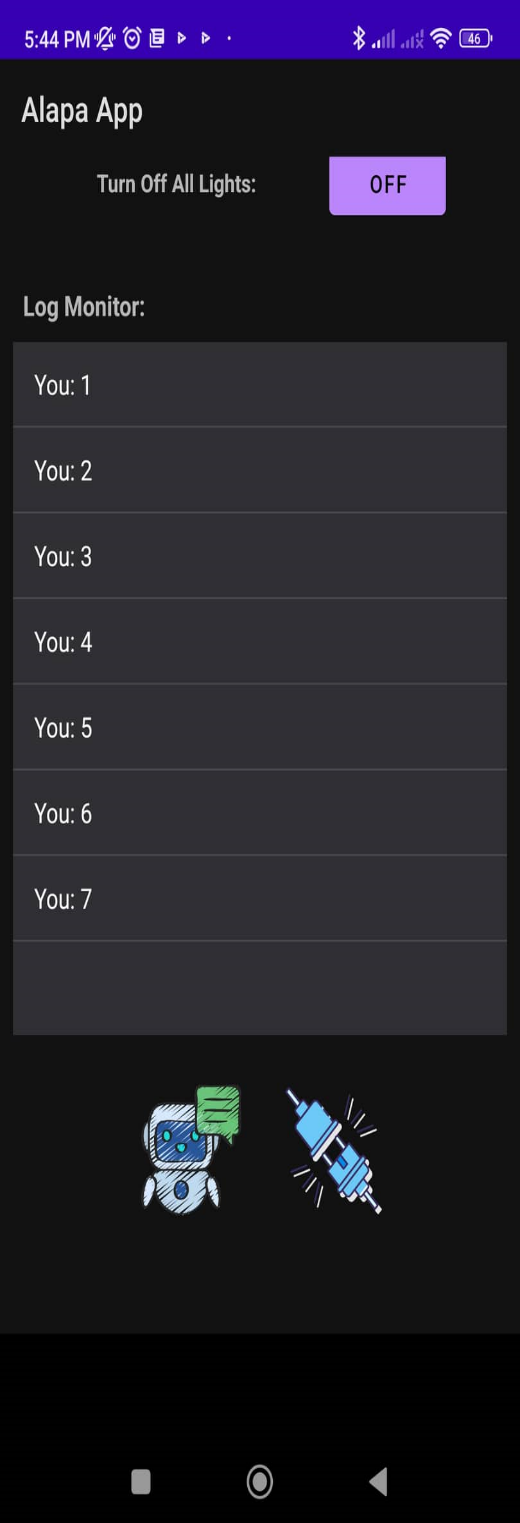
As the prototype’s outdoor lights are programmed to automatically turn on whenever the time of the day proceeds to night time, toggling this button can turn this specific automation setting ON or OFF. By default, ONL Settings is set to ON or enabled. However, disabling it will disallow the Arduino to run its designated automation program related to these outdoor lights’ which simply prevents them from turning on automatically during night time.

***Universal Command: Turn Off All Lights***

This button is non-toggleable. If tapped, all lights that are currently on will be turned off, with the exception of the entrance light as it automatically turns off after some time after being previously turned on. Also, the ONL Settings will be set to OFF or disabled.

**Developer Notes**

***Log Monitor Commands Guide***

****This function is purely for debugging purposes. To view this guide, the user must tap the Developer Notes icon beside the Disconnect icon, found by navigating down. Upon tapping, a pop-up note will display the Log Monitor Commands Guide.

**Debugging**

***Log Monitor***

This displays commands that are communicated between your device and the device it is connected to. This is to know if the actual commands that are sent from the application matches the ones that are accepted by the Arduino algorithm. The way it works is that it displays a command sent from your device, and when the Arduino receives it, will then return the same command back to your device. If the received command differs from what was initially sent, then an issue has been logged. Moreover, to avoid unnecessary memory allocation, if log texts exceed 8 text lines, the Log Monitor deletes all written lines.

The Log Commands can be found in the Developer Notes, found by navigating down the application.

1. **Important Notes**

**Warnings**

***Bluetooth MUST always be enabled***

Enable device Bluetooth always. Do not turn it off during the whole duration of using the application to prevent any connectivity issues.

***Pair with HC-05 Bluetooth Module first***

In order to form a connection with the Bluetooth Module, the user must first attempt to pair with it using the provided password. Connection is not possible if pairing of devices is not accomplished first.

***Disconnect with HC-05 Bluetooth Module first before closing the application***

When closing the application, the user must first disconnect with the Bluetooth Module by tapping the Disconnect icon found in the bottom part of the application’s interface three times. This is to ensure that after closing the application, the socket created with the module is terminated.

**Message from developer**

Should any issues or errors occur, please read this User Manual to find a corresponding solution to the said problem. If it is still not resolved, please contact the Developer of the application thru email. If you want to view the codes used to form the algorithms of the application, please contact the Developer using the email provided.

**References**

Abdoullaev, A. (2021, October 9). *How is Technology Changing the World Today?* How Is Technology Changing the World Today? Retrieved November 22, 2022, from <https://www.bbntimes.com/technology/how-is-technology-changing-the-world-today?fbclid=IwAR1rf9CDehs6dMwJW6JFtxFRqB9AxDGGhljJIc3_a8tmxFe0yMt-auy3OXA>

Grant, M., & Grant, M. (2019, February 5). *How Technology Has Made an Impact in Making Our Lives Better - Thrive Global*. Thrive Global. Retrieved November 22, 2022, from <https://community.thriveglobal.com/how-technology-has-made-an-impact-in-making-our-lives-better/>

Innovations. (2022, February 24). *Home Lighting Automation: Smart Lighting Control Systems*. Ezlo Smart Home. Retrieved November 22, 2022, from <https://www.ezlo.com/home-lighting-automation/>

*What is Arduino?* (n.d.). What Is Arduino? Arduino. Retrieved November 22, 2022, from <https://www.arduino.cc/en/Guide/Introduction>

Technologies, A. (n.d.). *What is Instrumentation and Control Engineering? - AGE Technologies*. What Is Instrumentation and Control Engineering? - AGE Technologies. Retrieved November 22, 2022, from <http://www.age.co.za/Marketing/WhatisInstrumentationandControlEngineering>

*Meet Android Studio | Android Developers*. (n.d.). Android Developers. Retrieved November 23, 2022, from <https://developer.android.com/studio/intro>

T., & Techbitar & Raquo;, M. A. (n.d.). *Modify the HC-05 Bluetooth Module Defaults Using AT Commands*. Instructables. Retrieved November 23, 2022, from <https://www.instructables.com/Modify-The-HC-05-Bluetooth-Module-Defaults-Using-A/>