

Making H105 a Smart Classroom

Detailed report

Sensors

Temperature and Humidity Sensor

Devices Used:

For connection with server : ESP32

For measuring Physical parameter: SHT21

Arduino Code for sensors:

```
#include <Wire.h>
#include <SoftwareSerial.h>
#include "WiFi.h"
#include "HTTPClient.h"
#include "ThingSpeak.h"
// sda 21
// scl 22
char* wifi_ssid = "<wifi_name>"; //Line 1
char* wifi_pwd = "<wifi_password>"; //Line 2
WiFiClient client;
unsigned long myChannelNumber = <THINGSPEAK_CHANNEL_NUMBER>; //Line 3
const char * myWriteAPIKey = "<write_API_key_from_thngspeak_account>"; //Line 4
#define si7021Addr 0x40
String tempStr;
String tempMeasure()
{
    unsigned int data[2];
    Wire.beginTransmission(si7021Addr);
    //Send humidity measurement command
    Wire.write(0xF5);
    Wire.endTransmission();
    // Request 2 bytes of data
    Wire.requestFrom(si7021Addr, 2);
    // Read 2 bytes of data to get humidity
    if(Wire.available() == 2)
    {
        data[0] = Wire.read();
```

```

        data[1] = Wire.read();
    }
    // Convert the data
    float humidity = ((data[0] * 256.0) + data[1]);
    humidity = ((125 * humidity) / 65536.0) - 6;
    Wire.beginTransaction(si7021Addr);
    // Send temperature measurement command
    Wire.write(0xF3);
    Wire.endTransmission();
    // Request 2 bytes of data
    Wire.requestFrom(si7021Addr, 2);
    // Read 2 bytes of data for temperature
    if(Wire.available() == 2)
    {
        data[0] = Wire.read();
        data[1] = Wire.read();
    }
    // Convert the data
    float temp = ((data[0] * 256.0) + data[1]);
    float celsTemp = ((175.72 * temp) / 65536.0) - 46.85;
    float fahrTemp = celsTemp * 1.8 + 32;

    ThingSpeak.setField(1, String(celsTemp));
    ThingSpeak.setField(2, String(humidity));
    int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    Serial.println("Posted:" + String(x));

    if(x == 200){
        Serial.println("Channel update successful.");
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " + String(x));
    }
    return (String(humidity) + " " + String(celsTemp));
}

void setup() {
    Serial.begin(9600);
    Wire.begin();
    Wire.beginTransaction(si7021Addr);
    Wire.endTransmission();
    delay(1000);
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);
    WiFi.begin(wifi_ssid, wifi_pwd);
    while (WiFi.status() != WL_CONNECTED) {

```

```

        delay(500);
        Serial.println("Connecting to WiFi..");
    }

    Serial.println("Connected to the WiFi network");
    ThingSpeak.begin(client); // Initialize ThingSpeak
    Serial.println("Setup done");
}

void loop()
{
    if(WiFi.status() != WL_CONNECTED) {
        WiFi.begin(wifi_ssid, wifi_pwd);
        while(WiFi.status() != WL_CONNECTED) {
            Serial.print('/');
        }
        Serial.println("Connected!");
    }
    Serial.println(tempMeasure());
    delay(30000);
}

```

Lines to edit while deploying:

- **Line 1:** Here enter your Wi-Fi name in place of <wifi_name>. Ensure that you have the ESP32' MAC Address added to that wifi if that is protected one of campus.
- **Line 2:** Enter your Wi-Fi password in place of <wifi_password>.
- **Line 3:** Enter your Thingspeak Channel number which can be found in your Thingspeak channel where you will be pushing data in place of placeholder <THINGSPEAK_CHANNEL_NUMBER>.
- **Line 4:** Enter write API key of Thingspeak channel in place of <write_API_key_from_thngspeak_account>.

CO2 sensor

Devices Used:

For connection with server : ESP32

For measuring Physical parameter: T6713

Arduino Code for Sensors:

```

#include <Wire.h> //Import Arduino Wire Library
#define T6713_Address 0x15 //T6713 i2C Address

int data [4];
int ppmValue;

void setup(){
  Serial.begin(9600);
  Serial.println("14CORE | T67XX / AN161 CO2 Sensor Test Code");
  Serial.println("Initializing.....");
  delay(2000);
  Serial.println("Starting i2C Communicate ADDR 0x15.....");
  Wire.begin();
  delay(1000);
}

void loop(){
  int ppmValue = readSensor();
  if (ppmValue > 0) {
    Serial.println("CO2-Carbon Dioxide Read Value > ");
    Serial.println(ppmValue);
  } else {
    Serial.println("ERROR | Failed to communicate to the sensor");
  }
  delay(2000);
}

int readSensor(){
  Wire.beginTransmission(T6713_Address);
  Wire.write(0x04);
  Wire.write(0x13);
  Wire.write(0x8B);
  Wire.write(0x00);
  Wire.write(0x01);
  Wire.endTransmission();
  delay(2000);
  Wire.requestFrom(T6713_Address, 4); //Request 4 bytes from the sensor
  data[0] = Wire.read();
  data[1] = Wire.read();
  data[2] = Wire.read();
  data[3] = Wire.read();
  Serial.println("FUNCTION CODE >");
  Serial.println(data[0], HEX);
  Serial.println("");
  Serial.println("BYTE COUNT > ");
  Serial.println(data[1], HEX);
  Serial.println("");
}

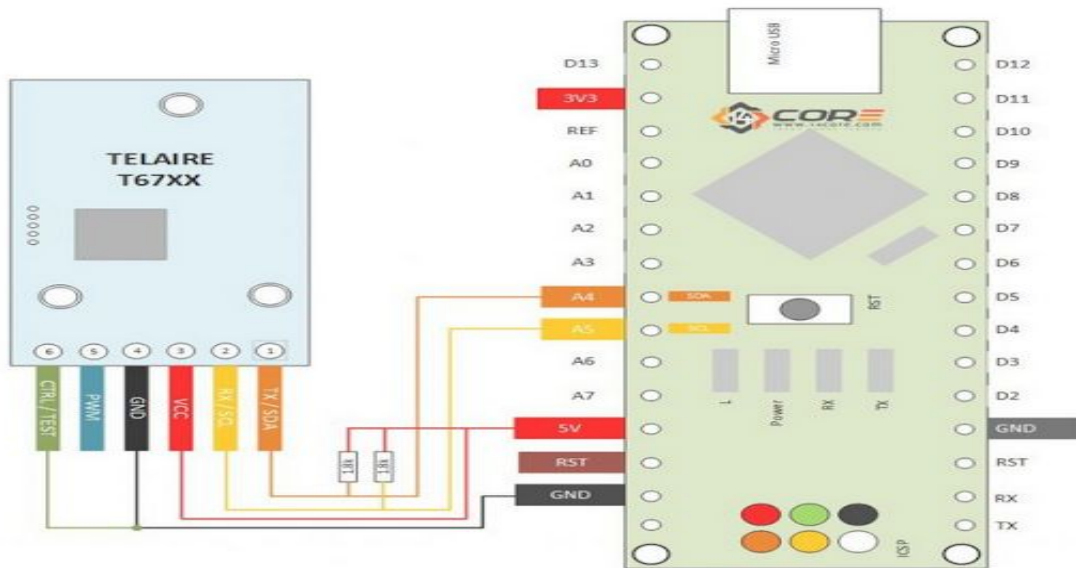
```

```

Serial.println("MOST SIGNIFICANT BIT > 0x");
Serial.println(data[2],HEX);
Serial.println("");
Serial.println("LEAST SIGNIFICANT BIT > 0x");
Serial.println(data[3],HEX);
ppmValue = (((data[2] & 0x3F ) << 8) | data[3]);
}

```

Circuit Diagram:

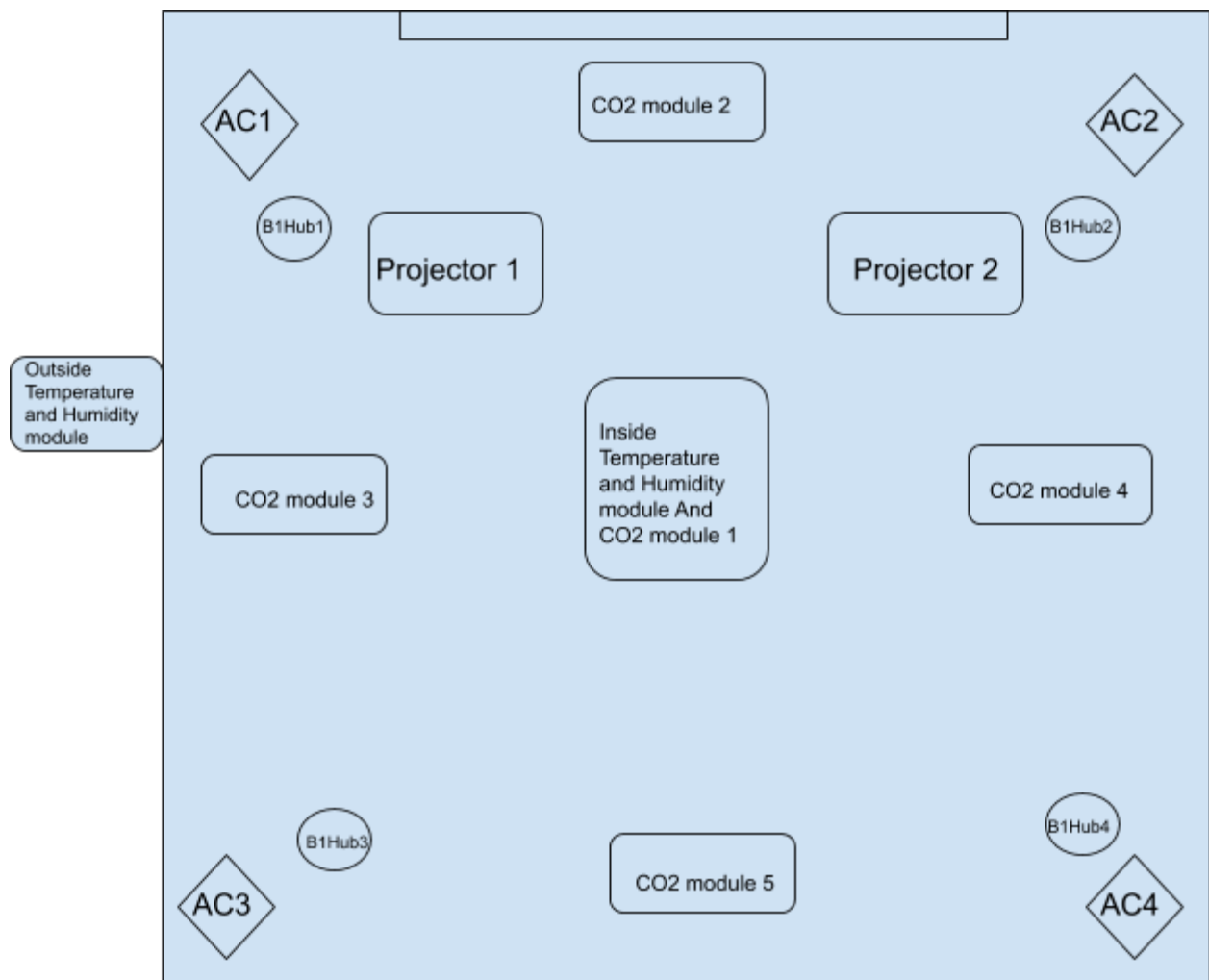


Instructions to connect with server:

- Insert the wifi connection code from Previous sensor code.
- Insert Server details like Write API key, Channel number etc.

Note: Some of Modules are yet to be built with working sensors.
Sensors provided to us were not in working condition.

Sensor module's locations in classroom:



Note: Positions might vary if the infrastructure is changed. Like if the evaporator starts working then positions of devices need to be adjusted. All B1 Hub Positions are tested by the team in presence of B1 Hub experts and mentors.

DETAILED B1 Hub

First of all below is the link to the official website of b1hub,

Link : <https://b1hub.com/>

About our key features



Security

Arm and monitor your Smart Home from anywhere and feel secure. Built for Security, B.One Hub is built on Military Grade Encryption platform with its own battery and cellular network backup. It is always ON and always Connected.



Energy Management and Lighting control

B.One allows you to manage and control all your energy devices using one single app. Know what devices are consuming how much power and make your environment energy efficient.



Comfort and Entertainment

B.One Hub monitors your Smart Home Thermostats, Humidity Sensors, CO Sensors and Smoke Sensors. B.One Hub has an on board Infrared Remote Chip with which you can manage all your IR remote controlled devices from one App. It is truly a Universal Remote control that works with both old and new Smart devices.



Actions

"Actions" are a series of highly customizable and configurable BOTs, based on user preferences, which can be activated at a touch of a button.



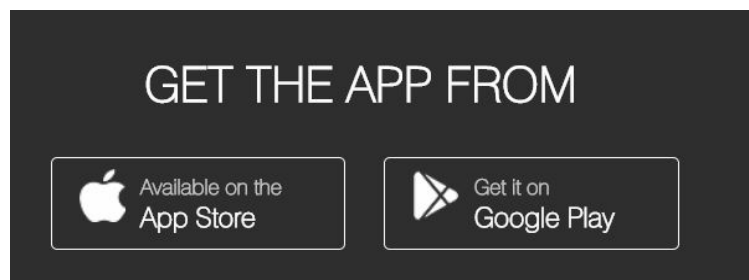
Syncs With[®] Eco-System

Supports over 60 devices and all prominent Wireless Protocol standards. We are rapidly increasing our ecosystem and aiming at reaching over 200 devices.



So Let's explore it more which will help you a lot while setting it up and in your project.

So you have to first install it's app " B.one " which is both available in the [PlayStore\(for Android\)](#) and [AppStore\(for IOS\)](#).



For a smooth working of an app you must have a good internet connection.

Tip : Try to avoid connecting the b1hub with different networks i.e try to avoid the change of Access point as it takes time to reconnect and it is only the phase where tension rises. Also Try to avoid power off the b1hub as if they will be connected to uninterrupted wifi network it will be easy for you to reconnect the app very fast and easily.

I know that you will have to power ON and OFF b1hub for testing purposes, you can do that and it will also work for you. The “ tip ” was just an advice only to reduce your difficulties when you will be working on this project specifically on b1hub.



We are controlling the Lights, projector, Ac's of H105 by it.

API Details :

- http://cbs.iiit.ac.in/?code=c2ed25ac5f00e643a08ed829cf8d978b8ae8da08&client_id=76dca191-4ade-498b-8655-9fcfd8efc295&hub_id=0CB2B706BFF5&scope=remote
- https://eazyapi.b1hub.com/OpenApi/authorize?client_id=76dca191-4ade-498b-8655-9fcfd8efc295&client_secret=0d4215e6-3da7-4728-9a25-4cd305f89e79&redirect_url=http://cbs.iiit.ac.in&response_type=code

In these details you will get the details for API such as **client_id**, **hub_id**, **client_secret** etc.

Note :- No need to open the link only for information purposes.

These are used for backend purposes i.e control the b1hub from the app or post request. We can also control it easily by the b.one app but in this project we have to make an android app for H105 which will work according to our needs. I hope you get it.

To setup B1 hub you just need to follow the instructions given in the app.

More important when you change your access point i.e try to connect to another person's mobile app or using hotspot from your friends phone network while testing the B1 hub. You must follow the steps given below :

Settings -> Hub Settings -> Change Access Point.

Then follow the instructions of the app to change the access point.

That's an easy task then ;).

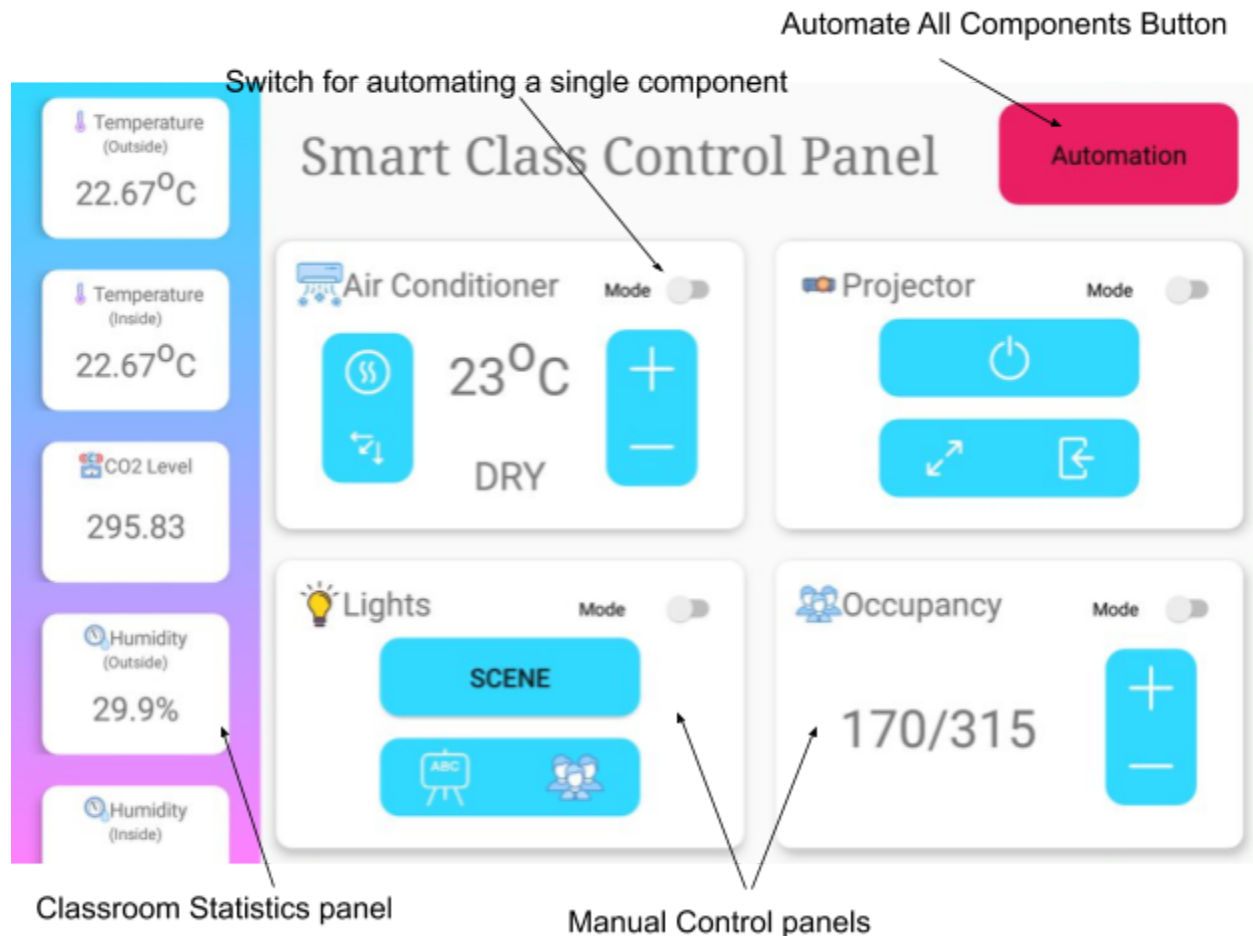
Technical Support :

- After reading this even if you are in trouble in set up of B1hub then you can drop an email to :

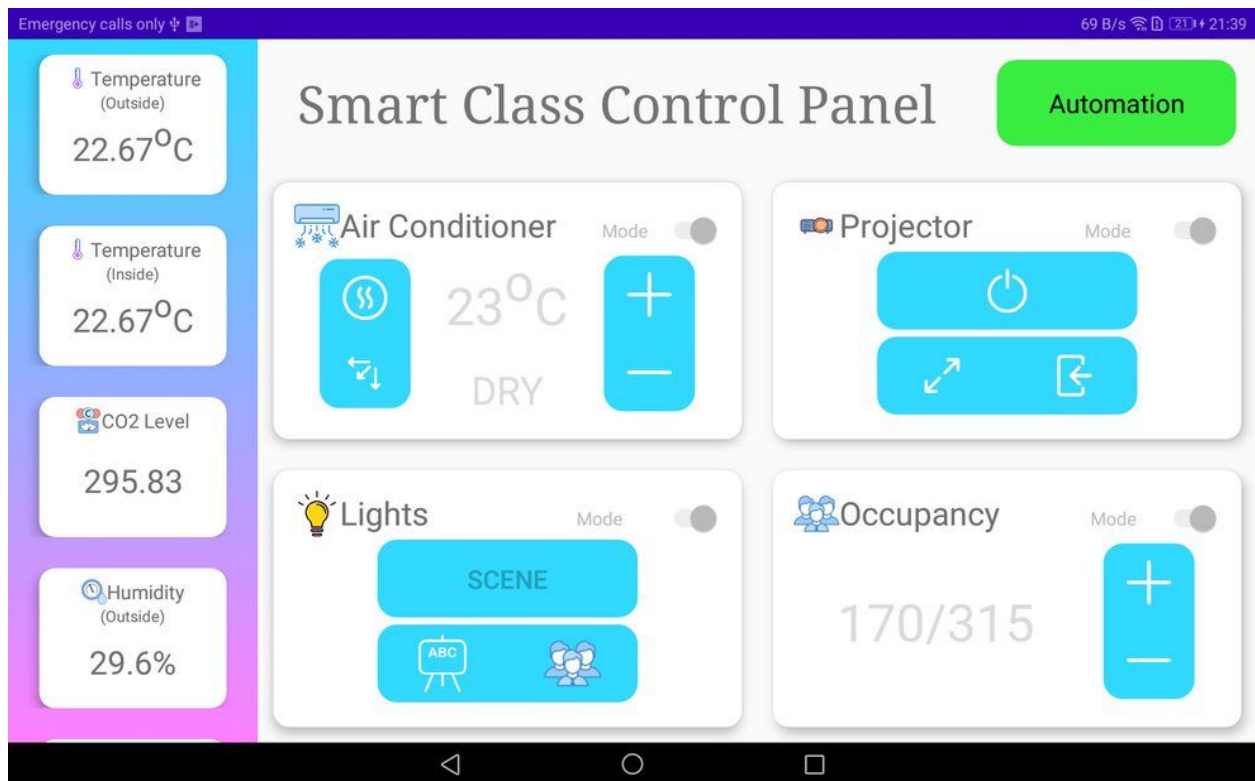
- support@blazeautomation.com
- Or even you can call them through their B1hub app only.
- Even if that doesn't work for you then try to contact Professor Vishal Garg or Simran Singhal as they have direct contact to the officials of B1Hub.

ANDROID APP

The Application Interface





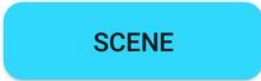





Clicking Automate Button will switch the mode of all components to “ON” state and will disable all the manual controls. The classroom will be set based on the current physical stats and the occupancy and other conditions.



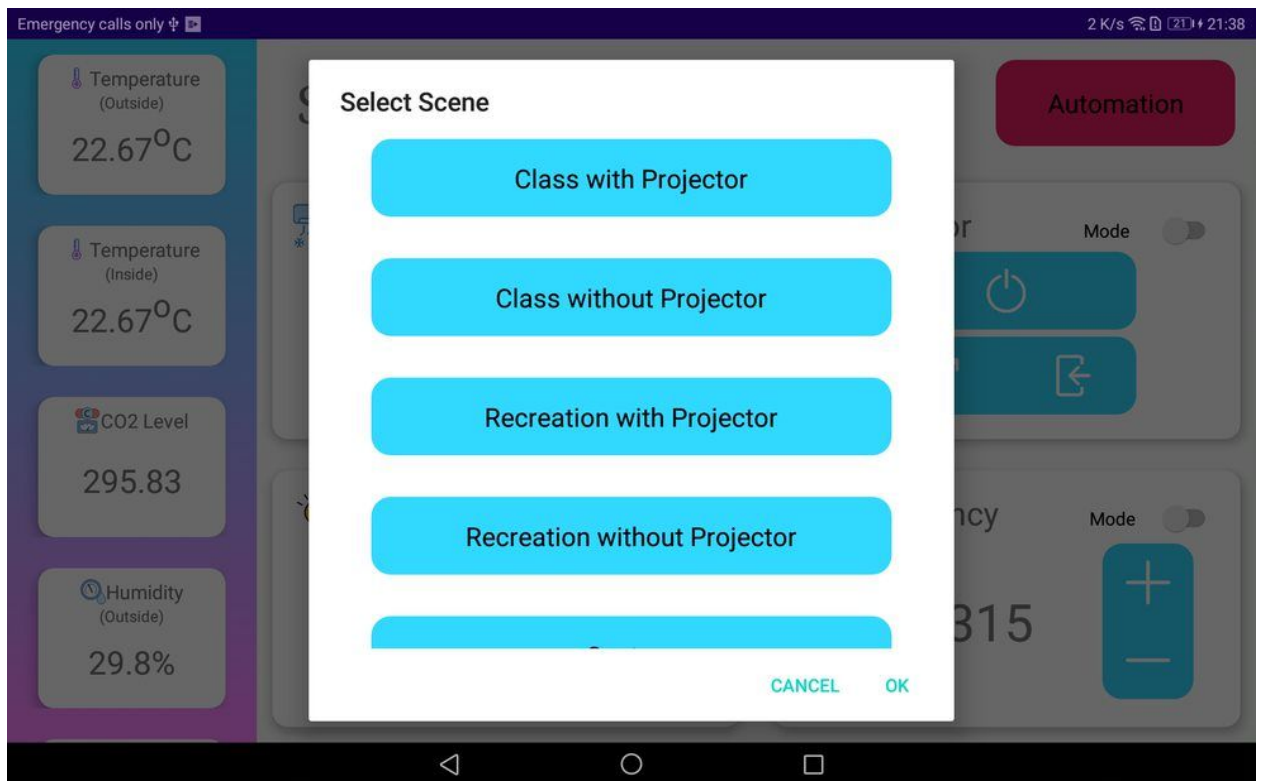
In the above image all the controls are disabled and no manual inputs can be given to the system.

The various buttons and displays on the panel.

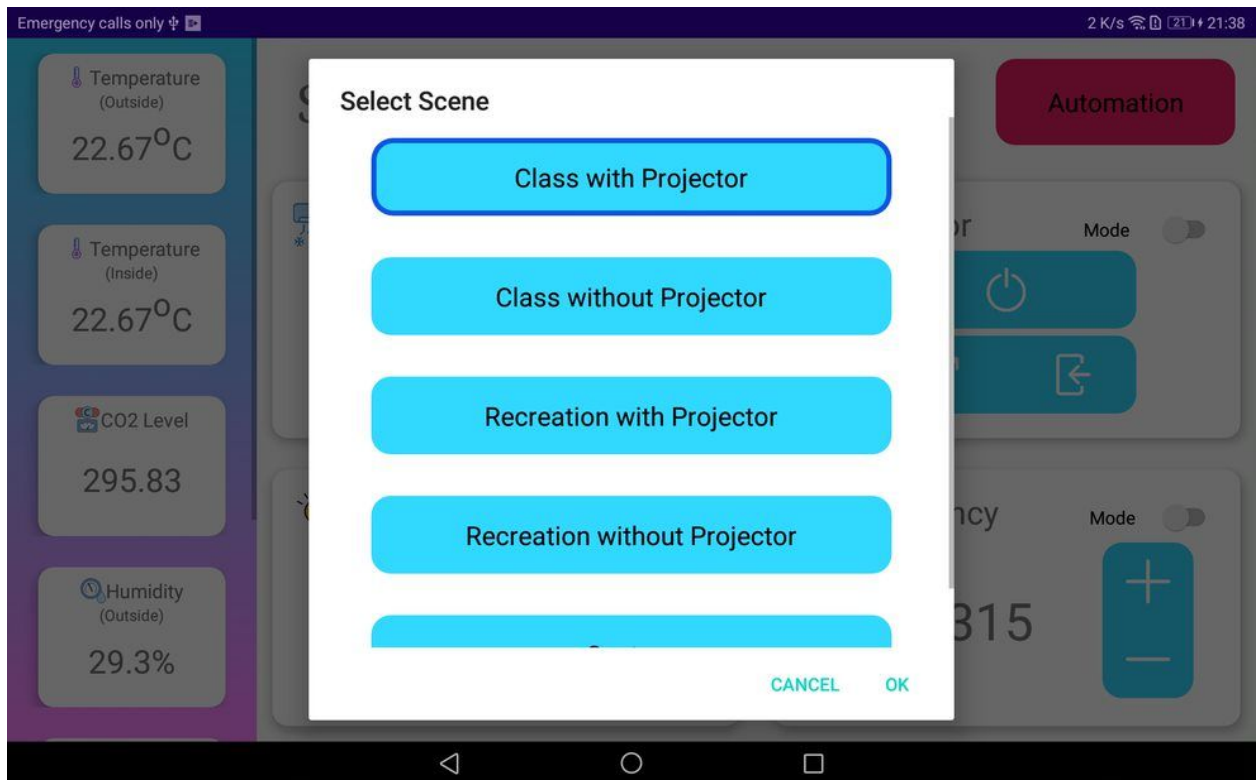
Sno	Button	Function
1		will increase and decrease temperature and occupancy
2		The upper button is for setting AC Mode and the lower button is to toggle swing
3		To turn projector ON and OFF
4		Left button changes the aspect ration and the right button changes the input for projector
5		Used for selecting one of the preset light layouts or custom light layout
6		Left Button is for toggling Green Board Focus Lights and right button is for toggling Audience overhead lights
7		Shows current temperature setting for the ACs and Current Cooling mode for AC
8		Shows Current Occupancy/Max Occupancy

How to Select Scenes

1. Click Scene button in the lights control panel. A layout of the following form should open

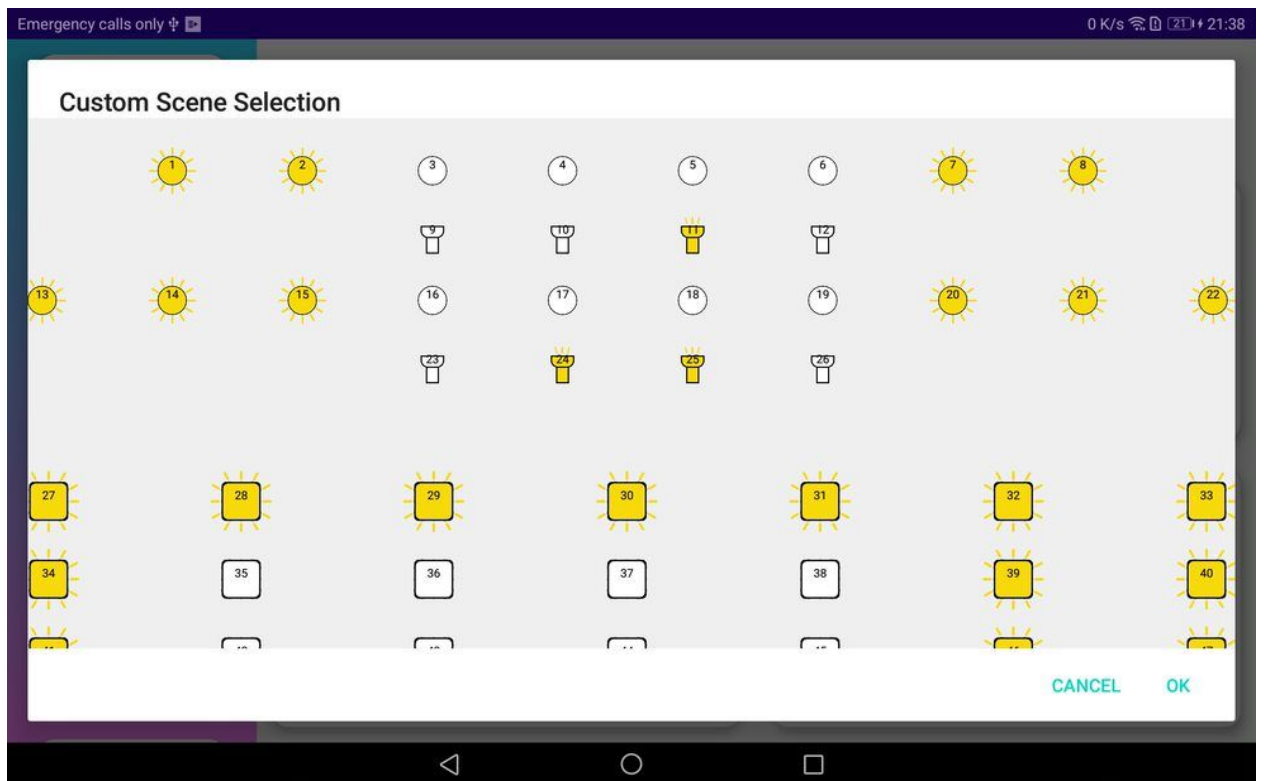


2. Selecting any one of the given setting will highlight the selection



3. Click Ok to Apply

4. There is an option for custom scene. Clicking on that will open a new layout in the following form



5. Selects the lights in the custom scene select dialog box and click OK to apply
6. The scene should now be applied