CO326 Project :Smart Building
Group D- Energy

Group Members:

E/17/012 Amarasinghe R. A. A. U.
E/17/038 Chandrasekara C.M.A
E/17/040 Chandrasena M.M.D
E/17/044 Coralage D.T.S.
E/17/101 Gunathilaka S.P.A.U.
E/17/242 Perera C.R.M.
E/17/252 Perera U.A.K.K.
E/17/356 Upekha H.P.S
E/17/407 Wljesooriya H.D

Content

Introduction

Energy used in buildings (residential and commercial) accounts for a significant percentage of a country's total energy consumption. This percentage depends greatly on the degree of electrification, the level of urbanization,the amount of building area per capita, the prevailing climate, as well as national and local policies to promote efficiency.

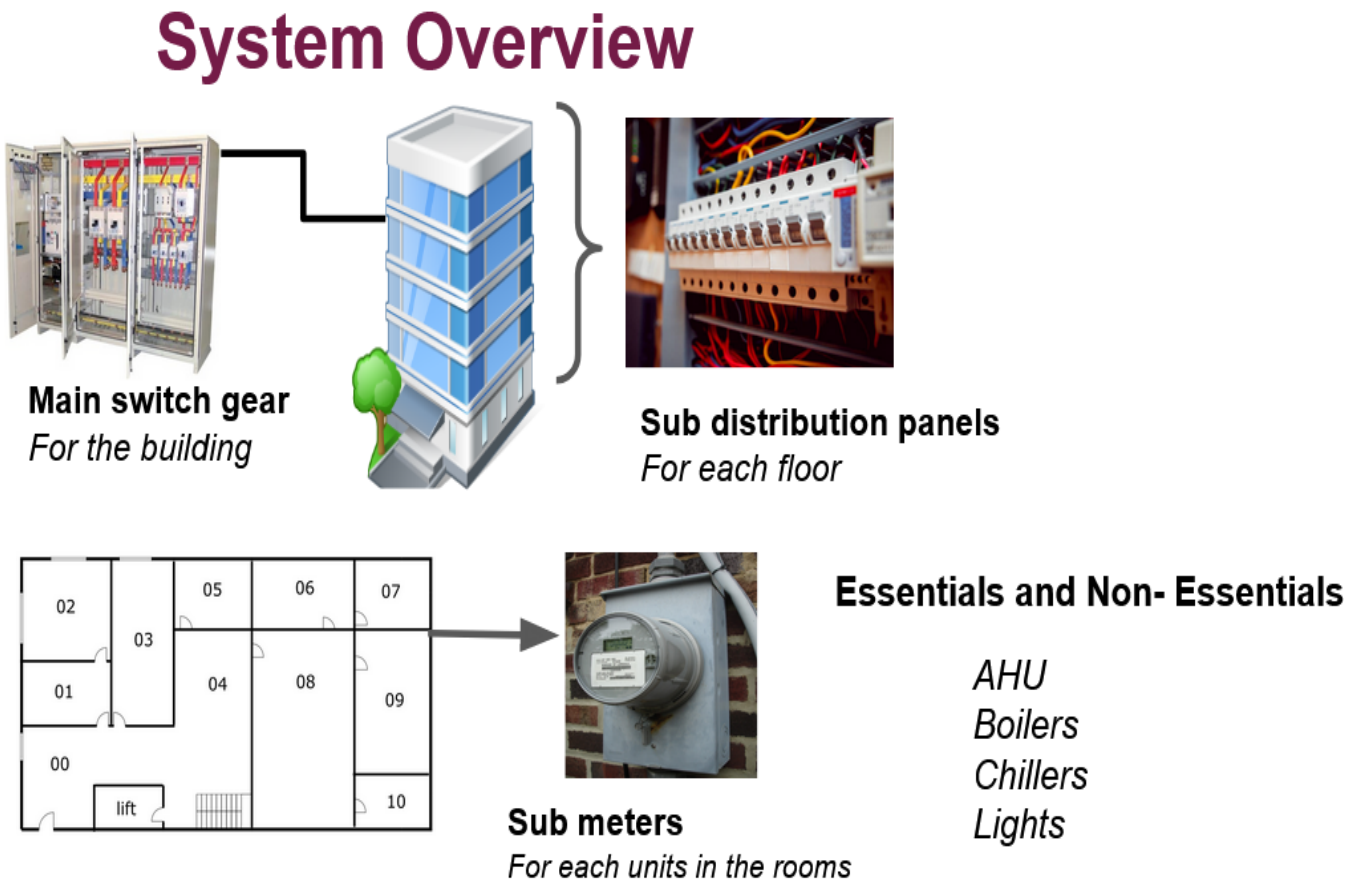
Energy efficiency is more than simply reducing the amount of electricity used in the building.It is about the appropriate use of electricity to generate the most production for the least amount of energy and cost.An energy management system (EMS) supports the reliable functionality ,maximizes the penetration of renewable energy, and optimizes the cost and economic efficiency in the associated smart building

The main benefit from measures to improve energy efficiency buildings is lower energy costs but there are usually other benefits to be considered too. Energy efficiency measures are meant to reduce the amount of energy consumed while maintaining or improving the quality of services provided in the building. Among the benefits likely to arise from energy efficiency investments in buildings are

- Reducing energy use for space heating and/or cooling and water heating;
- Reduced electricity use for lighting, office machinery, and domestic type
- appliances;
- Lower maintenance requirements;
- Improved comfort;
- Enhanced property value

★ **Since Energy is a broader area, in this project we will consider only Electrical energy.**

Design Overview



Major components

Main Switch Gear:

In the system, switchgear is composed of electrical disconnect switches, fuses or circuit breakers used to control, protect and isolate electrical equipment. Switchgear is used both to de-energize equipment to allow work to be done and to clear faults downstream.

Sub-meters for each units in the floor:

Submeters placed on a subpanel or particular electrical circuit may measure consumption in a building subsection like a floor, or equipment

Sub distribution panels for each floor:

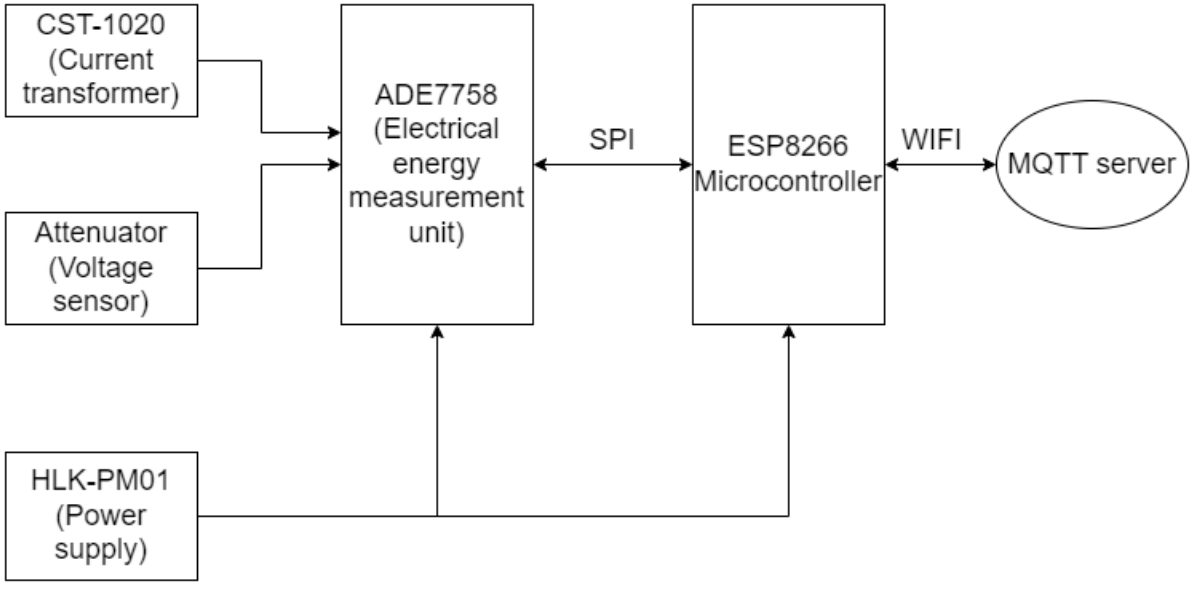
part of distribution panels which consists of some breaker components and or some controls to distribute electrical surge/power to the electrical circuit installation in the area.

Main unit of the system

We are replacing the traditional electrical outlets with a **smart energy meter with load control**. This measures the energy consumption at device level and allows on/off switching of equipment connected to it.

The meter unit is built around the ADE7758 as an energy measurement unit, the ESP8266 microcontroller, the CST-1020 current transformer, a resistive attenuator for the voltage input and an integrated power supply HLK-PM01

Architecture of the smart meter



Energy measurement unit: ADE7758

A high accuracy three-phase electrical energy measurement IC (but can also be

implemented for single-phase systems). We interface it with the microcontroller using the SPI serial communication interface. It incorporates a second-order Delta-sigma type ADC, a digital integrator, reference circuits, a temperature sensor and also the algorithms to determine the active, reactive and apparent power, active and reactive energy and rms voltage and current calculations. Has six analog inputs divided into two sets for current and voltage measurement.

Micro controller: ESP8266

Built in WIFI module easily facilitates the communication of the sensor readings to the servers.

Current sense input: CST-1020

Traditional electrical outlets typically handle upto 15A. Mount transformer has a turns ratio of 1000:1 and is capable of handling 20 A and able to operate at 50Hz.

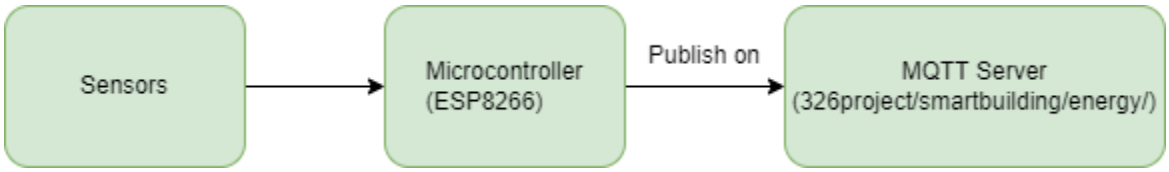
Voltage sense input: Attenuator

The line voltage is attenuated using a simple resistor divider network.

Communication of sensor data to servers

Sensor readings will be communicated to other layers via a MQTT enabled IoT device connected to the sensor. This section will publish under the topic:

co326project/smartbuilding/energy/



Sensor data(power Consumption) will be read for each floor, room and each unit of rooms.

Units of rooms include:

- AHU
- Boiler
- Chiller
- Lighting
    - Essential line
    - Non-essential line

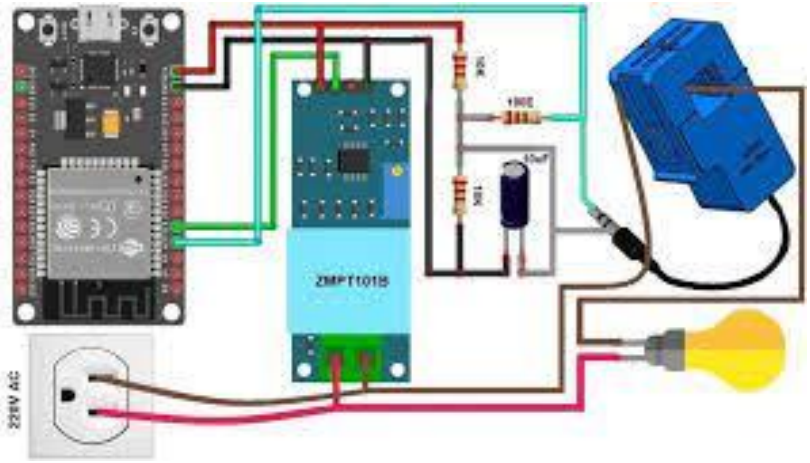
Sub topic	Data format
Monitoring power consumption readings	

floorX/power	{
	“floorno”
	“unit”
	“power”: power consumption
	“timestamp”: time of sensor reading
	}

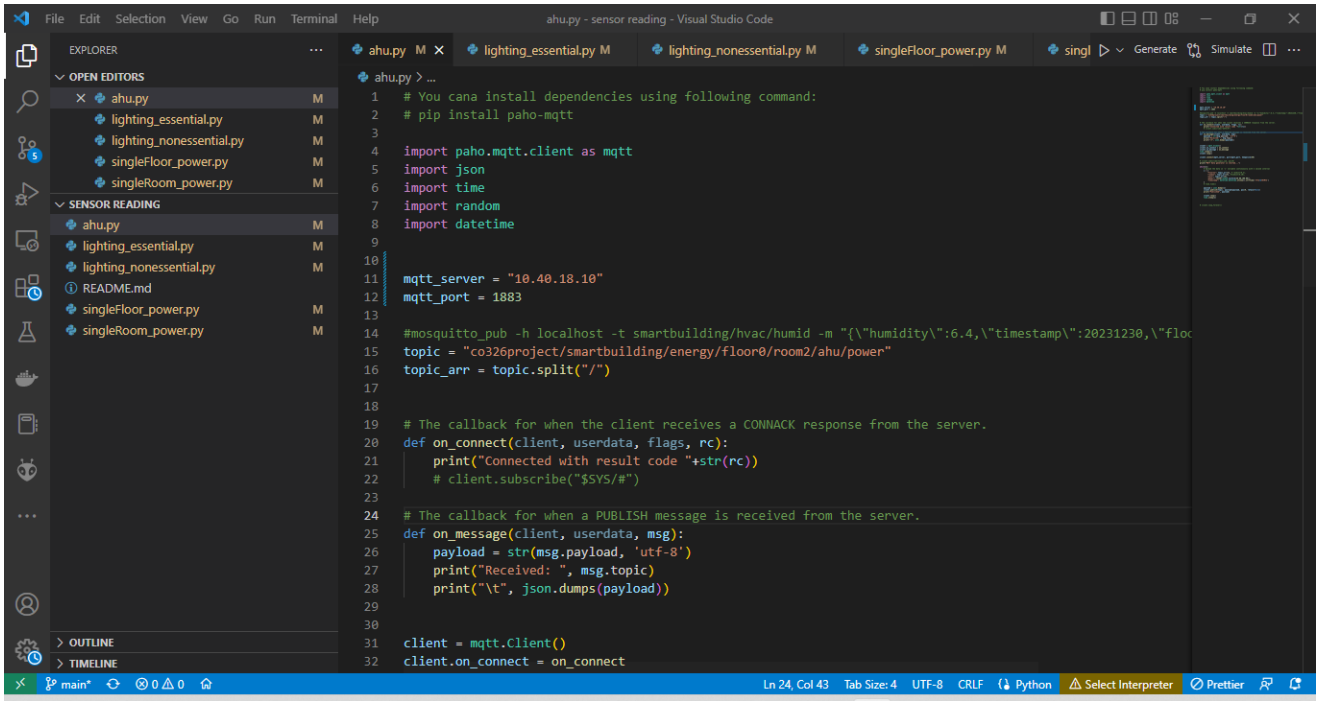
floorX/roomX/power	{
	“floorno”
	“roomno”
	“power”: power consumption
	“timestamp”: time of sensor reading
	}

floorX/roomX/<unit>/power
 	
Unit:
ahu, b, c ,lighting/essential, lighting/nonessential
	{
	“floorno”
	“roomno”
	“unit”
	“line”: for lighting essential/nonessential
	“power”: power consumption
	“timestamp”: time of sensor reading
	}

A simple demonstration circuit is given below



Due to the unavailability of sensors python scripts have been used to simulate the sensors




```
# You cana install dependencies using following command:

# pip install paho-mqtt



import paho.mqtt.client as mqtt

import json

import time

import random

import datetime




mqtt_server = "10.40.18.10"
```

```
mqtt_port = 1883



#mosquitto_pub    -h    localhost    -t    smartbuilding/hvac/humid    -m
"{\"humidity\":6.4,\"timestamp\":20231230,\"floorno\":2,\"roomno\":4}"

topic = "co326project/smartbuilding/energy/floor0/room2/ahu/power"

topic_arr = topic.split("/")




# The callback for when the client receives a CONNACK response from the
server.

def on_connect(client, userdata, flags, rc):

    print("Connected with result code "+str(rc))

    # client.subscribe("$SYS/#")



# The callback for when a PUBLISH message is received from the server.

def on_message(client, userdata, msg):

    payload = str(msg.payload, 'utf-8')

    print("Received: ", msg.topic)

    print("\t", json.dumps(payload))




client = mqtt.Client()

client.on_connect = on_connect

client.on_message = on_message
```

```
time.sleep(2)

client.loop()



client.connect(mqtt_server, port=mqtt_port, keepalive=60)



# client.subscribe(topic_sub, qos=0)

print("MQTT Data generator is started...")



while(1):

    # Upload the data in 'x' variable continuously with 2 second interval

    x = {

        "floorno": topic_arr[3], # randint(0,2),

        "roomno": topic_arr[4], #randint(0,10),

        "unit": topic_arr[5],

        "power": round(random.uniform(30.00,100.00)),

        "timestamp": datetime.datetime.utcnow().strftime('%Y%m%d%H%M%S')

    }

    # time.time()



    payload = json.dumps(x)

    client.publish(topic, payload=payload, qos=0, retain=False)

    print("Published", payload)
```

```
    client.loop()

    time.sleep(2)




# client.loop_forever()
```

Get Control Signals from the MQTT Server Control the actuators

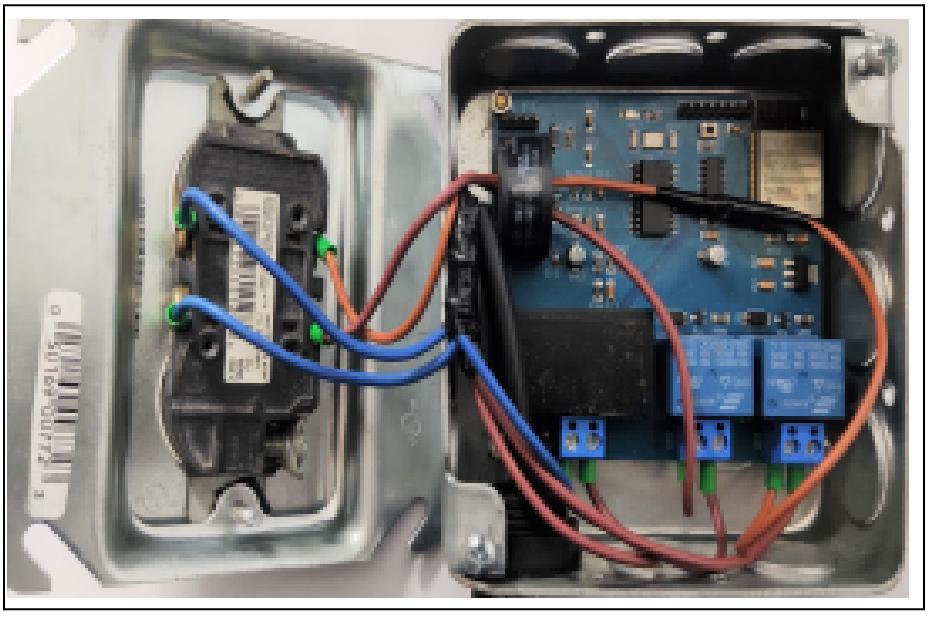
In this energy management system, the control process is done by the actuators. In a traditional energy management system the main actuator is circuit breakers in switch gears and sub distributional panels. But in this smart building system, those actuators should be automated. There are some circuit breakers which can be automated with wifi, but in this case relay elements are used instead of the circuit breakers.

These relay elements can be named as load switches and which contain electromechanical relays to handle the load. Specifically, the SRA-05-VDC-CL relays are used due to their technical specifications. Here, the coil's nominal voltage is 5V and the nominal current is 120mA. This can withstand the maximum current of 20A and that is the main reason why this is used as the relay for this with its size and the performance.

Some other actuators are:

1. Generator starter actuators(choke actuators) - which is used to start the generator automatically at a power failure.

This is an example circuit board where the load switches has been used to control the load automatically.



Basically the control of these actuators done by SCADA system and control signal can be generated by the switches at the SCADA interface manually or the control signal is generated by a written program at the SCADA to handle with respect to the sensing values and processed data with previously extracted sensing data.

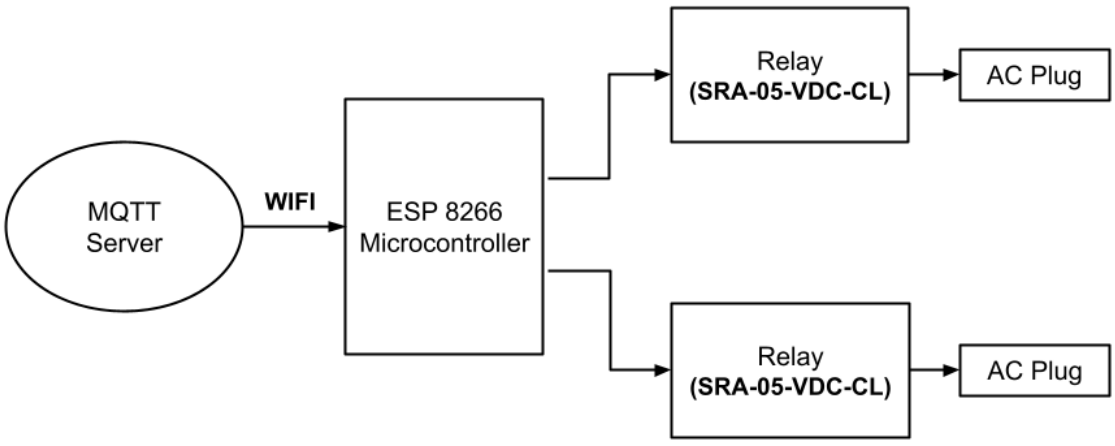
Two methods :

1. Manually by SCADA interface
2. Automatically by implementing a program with respect to the analyzed data previously obtained by sensors.

In these two methods, the priority is given to the switches on the SCADA interface as it should be eligible to handle manually at any time or at an immediate situation.

In both the methods, the control payload is sent via mqtt server and the microcontroller which controls the actuators should subscribe the relevant topics for the relevant actuators which are controlled by itself respectively.

Architecture of Load Control



9

The microcontroller should subscribe to the control topics like the table below.

Sub topic	Data format
Control signal for acutators	

floorX/power/control	{
	“State”: 0/1,
	“Line”: “e”/”n”,
	“Mode”: “m”/”a”
	}

floorX/roomX/power/control	{
	“State”: 0/1,
	“Line”: “e”/”n”,
	“Mode”: “m”/”a”
	}

floorX/roomX/<unit>/power/control
 	
Unit:
ahu, b, c ,lighting/essential, lighting/nonessential
	{
	“State”: 0/1,
	“Line”: “e”/”n”,
	“Mode”: “m”/”a”
	}
	12

The microcontroller should decide which line should be disconnected or connected. For example, if it gets a payload to disconnect the load for the entire floor, then the microcontroller should turn off the relay which handles the load to the floor. The priority of controlling the relays has been given such that highest for the largest. Which means it checks the entire building first. Then the floor wise and room wise next and unit wise at last. That is the same as the hierarchy of the topics it has subscribed to.

Then it will check whether the line is the essential or the non-essential according to the payload. After that it will check the mode whether the payload came from the SCADA interface manually or it is automatically generated by the function implemented on the SCADA.

mqtt1

```
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("connecting..");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  char brightness[length];
  for (int i = 0; i < length; i++) {
    brightness[i] = (char)payload[i];
```

```
char brightness[length];
for (int i = 0; i < length; i++) {
  brightness[i] = (char)payload[i];
  Serial.print(brightness[i]);

}

Serial.print("\n");

if(strcmp(topic, "326project/smartbuilding/energy/floor0/power/control") == 0){
  if ((char)payload[9] == '0') {
    Serial.println((char)payload[9]);
    digitalWrite(LED_ESSENTIAL, HIGH );   // Turn the LED on (Note that LOW is the voltage level
  }
  else if((char)payload[9] == '1'){
    digitalWrite(LED_ESSENTIAL, LOW );   // Turn the LED off (Note that LOW is the voltage level
    Serial.print("HIGH") ;
  }

}

else if(strcmp(topic, "326project/smartbuilding/energy/floor0/room0/power/control") == 0){
  if ((char)payload[9] == '0') {
    Serial.println((char)payload[9]);
    digitalWrite(LED_ESSENTIAL, HIGH );   // Turn the LED on (Note that LOW is the voltage level
  }
  else if((char)payload[9] == '1'){
    digitalWrite(LED_ESSENTIAL, LOW );   // Turn the LED off (Note that LOW is the voltage level
    Serial.print("HIGH") ;
```

```
else{
  if(strcmp(topic, "326project/smartbuilding/energy/floor0/room2/b/power/control") == 0){
  // Switch on the LED if an 1 was received as first character
    int pin;
    if ((char)payload[19] == 'n') {
      pin = LED_NON_ESSENTIAL;
    }
    if ((char)payload[19] == 'e') {
      pin = LED_ESSENTIAL;
    }
    else{
      pin = LED_NON_ESSENTIAL;
    }

    if ((char)payload[9] == '0') {
      Serial.println((char)payload[9]);
      digitalWrite(pin, HIGH );   // Turn the LED on (Note that LOW is the voltage level
    }
    else if((char)payload[9] == '1'){
      digitalWrite(pin, LOW );   // Turn the LED off (Note that LOW is the voltage level
      Serial.print("HIGH") ;
    }
    // but actually the LED is on; this is because
    // it is active low on the ESP-01)
  }
  if(strcmp(topic, "326project/smartbuilding/energy/floor0/room2/l/power/control") == 0){
  // Switch on the LED if an 1 was received as first character
    if ((char)payload[9] == '0') {
      Serial.println((char)payload[9]);
```

```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.subscribe("co326project/smartbuilding/energy/floor0/room2/ahu/power");
      client.subscribe("326project/smartbuilding/energy/floor0/room2/b/power/control");
      client.subscribe("326project/smartbuilding/energy/floor0/room2/l/power/control");
      client.subscribe("326project/smartbuilding/energy/floor0/power/control");
      client.subscribe("326project/smartbuilding/energy/floor0/room0/power/control");

    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

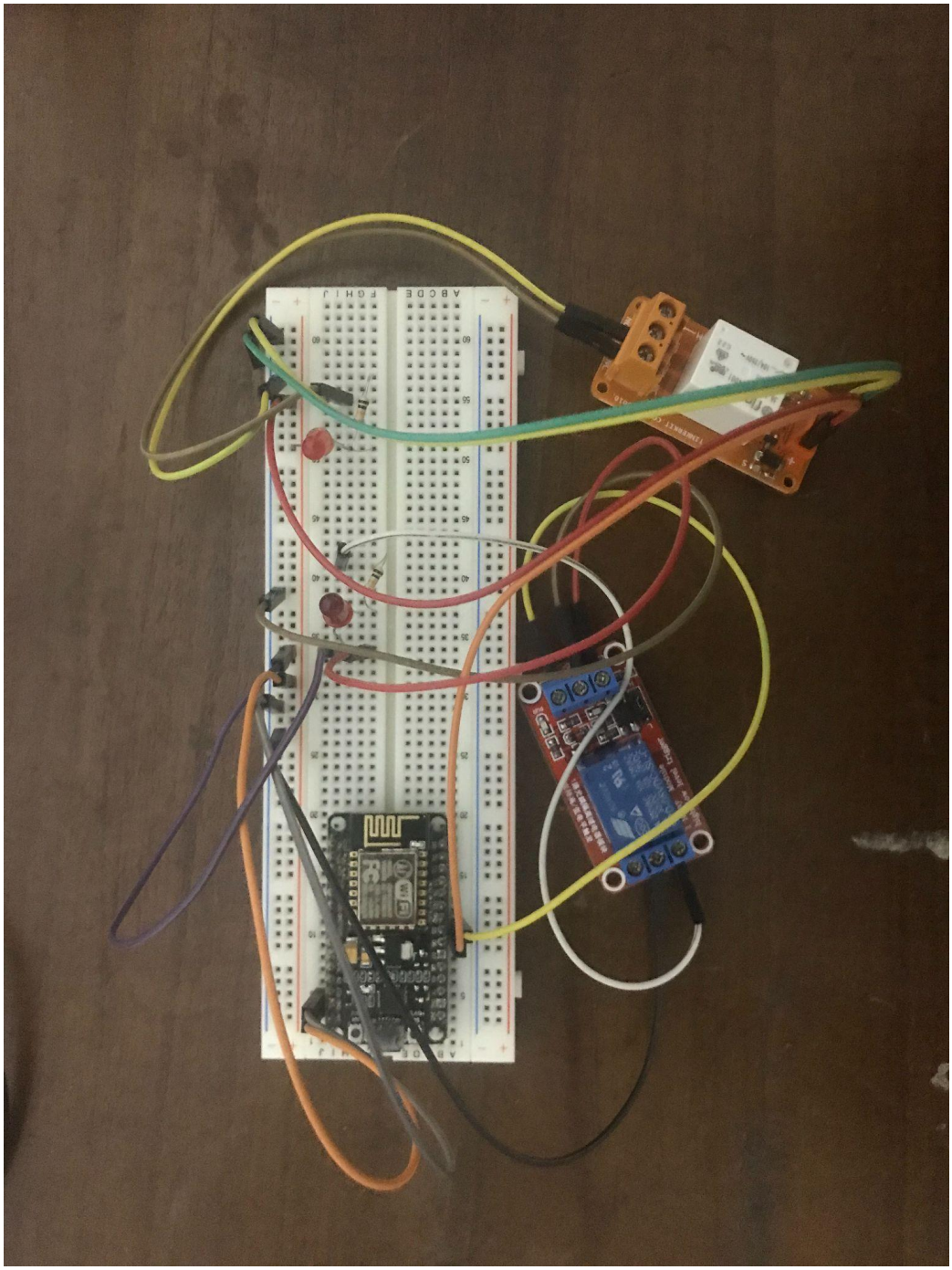
```
void setup() {
  pinMode(D0, OUTPUT);     // Initialize the BUILTIN_LED pin as an output
  pinMode(LED_ESSENTIAL, OUTPUT);
  pinMode(LED_NON_ESSENTIAL, OUTPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883); //8883
  client.subscribe("326project/smartbuilding/energy/floor0/room2/b/power/control");
  client.subscribe("326project/smartbuilding/energy/floor0/room2/l/power/control");
  client.subscribe("326project/smartbuilding/energy/floor0/power/control");
  client.subscribe("326project/smartbuilding/energy/floor0/room0/power/control");


  client.setCallback(callback);
}

void loop() {

  short reading;
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
```

This is the circuit implemented to check whether actuators are working according to the control signals sent by the SCADA via mqtt.



Read data from the MQTT Server display the status of the system on SCADA

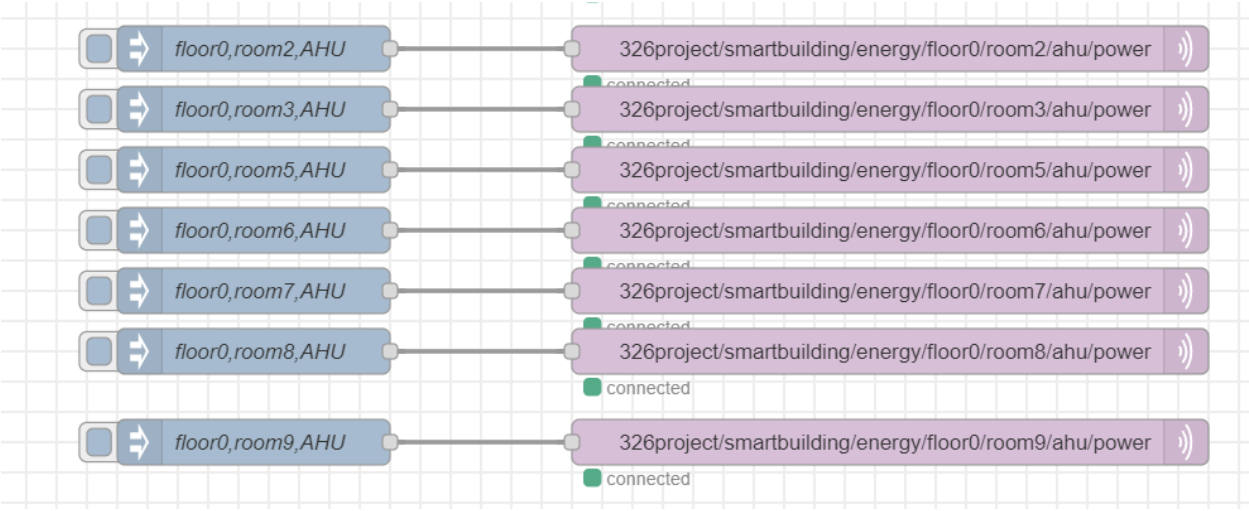
In order to exchange the data, the SCADA system should have been subscribed to the same topics as the other sensors. As the sensors publish their data, those data will be published to the SCADA system by the MQTT broker. Since the SCADA system has subscribed to that topics it will be able to retrieve data from the MQTT broker.

The following information will be published to the SCADA system and they will also be displayed on the Node-RED dashboard.

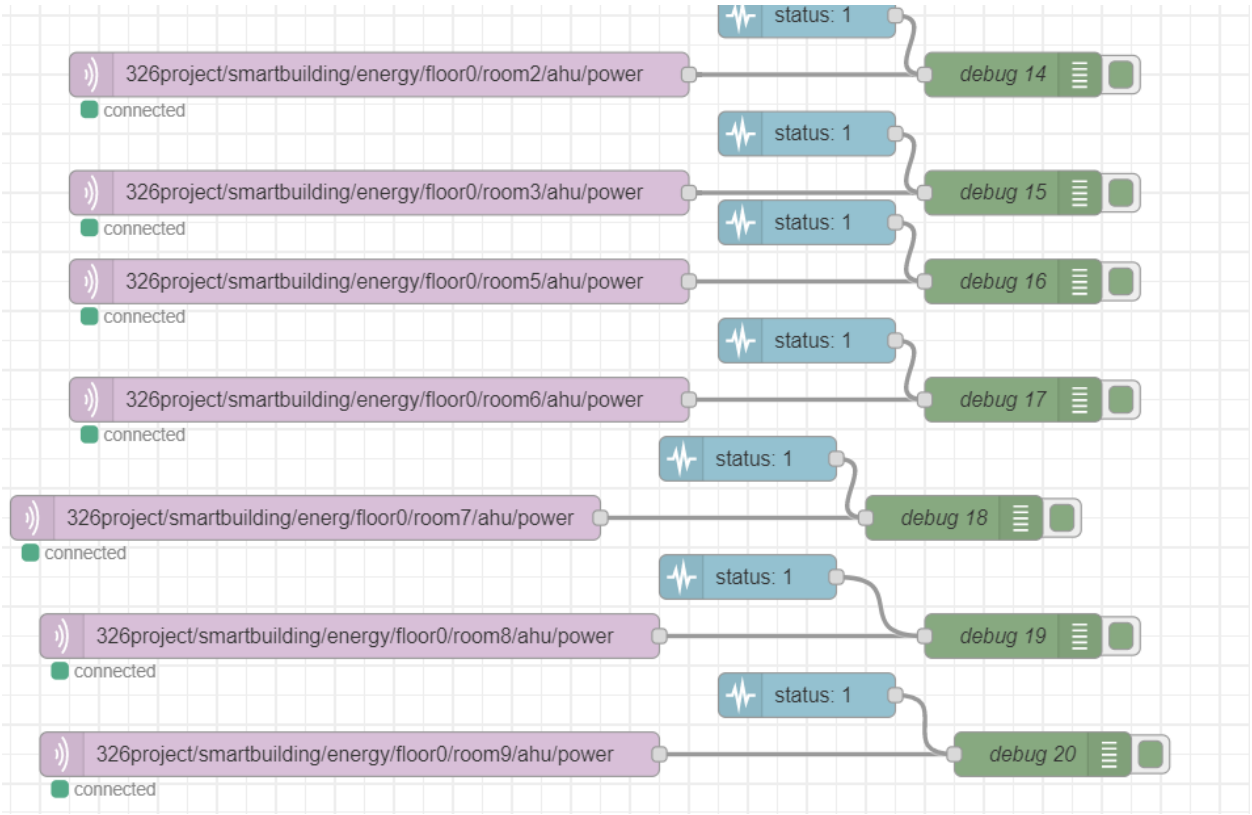
- Power consumption of the lights in each room
- Power consumption of the air handling units in rooms
- Power consumption of the boilers in rooms
- Power consumption of the chillers in rooms
- Total power consumption of each room

Our Node-RED implementations include:

- MQTT ‘publish’ topics correspond to the air handling units of the floor0 - sensors will publish the power consumed by the air handling units of the floor0, to these topics.

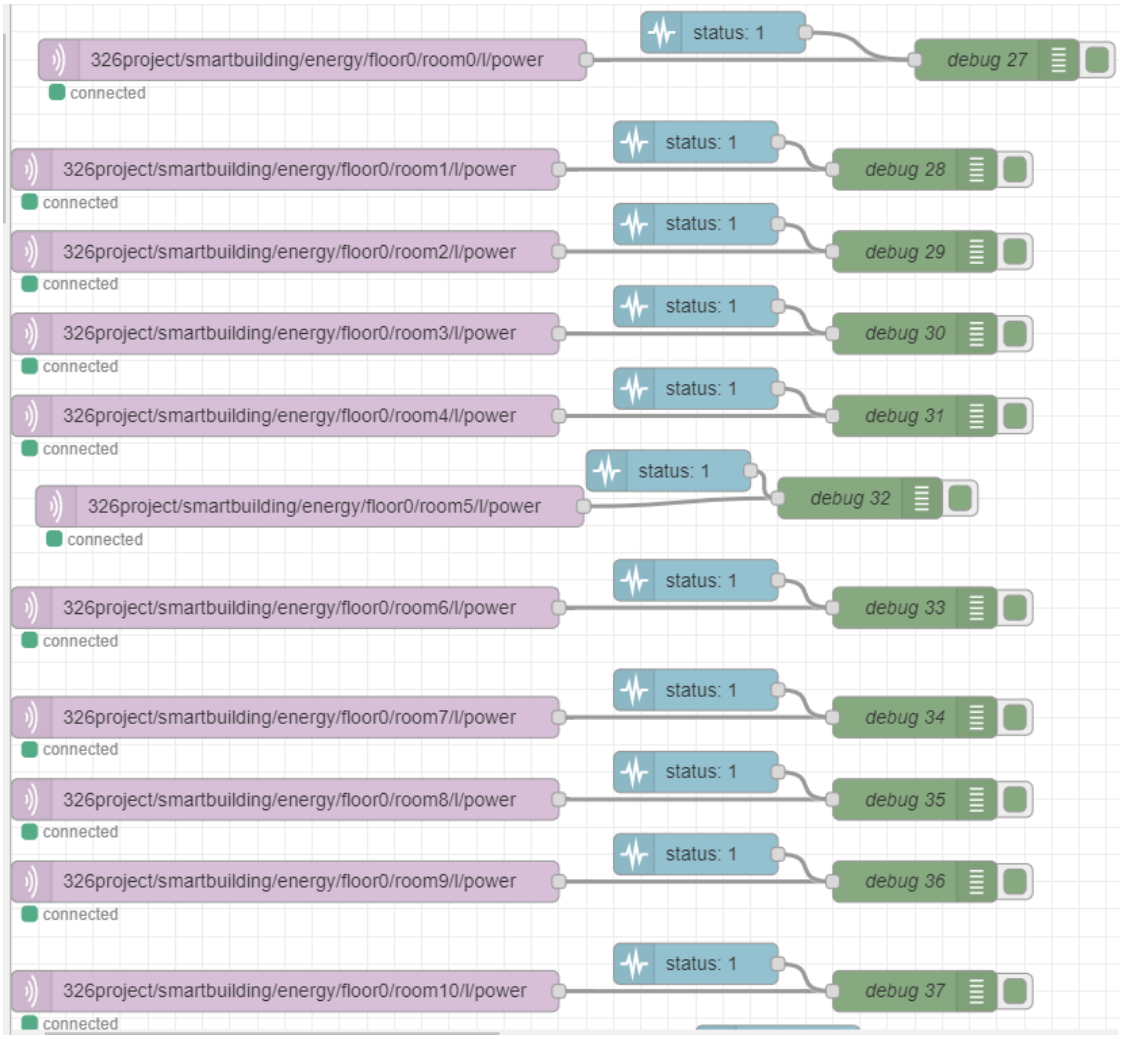


- MQTT ‘subscribe’ topics correspond to the air handling units of the floor0 - clients have to subscribe to these topics.



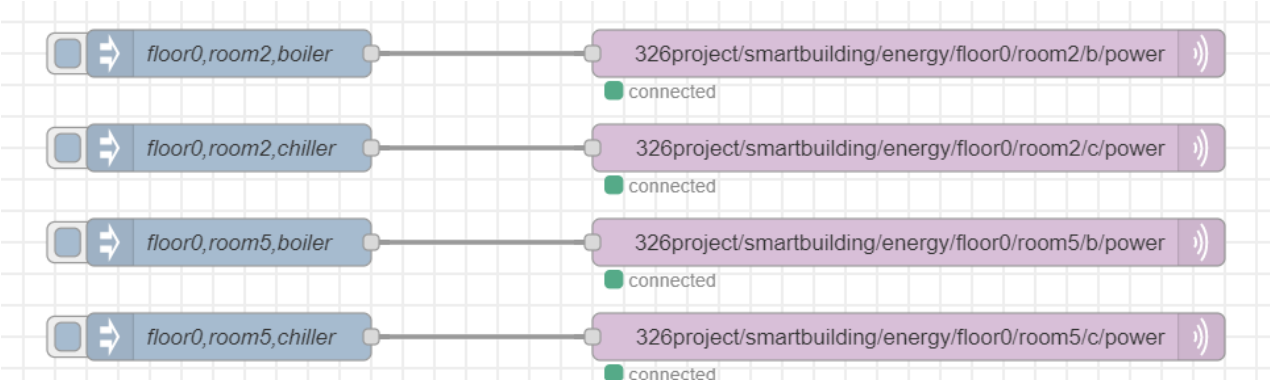
- MQTT ‘publish’ topics correspond to the lights of the floor0 - sensors will publish the power consumed by lights of the floor0, to these topics.

- MQTT ‘subscribe’ topics correspond to the lights of the floor0 - clients have to subscribe to these topics.

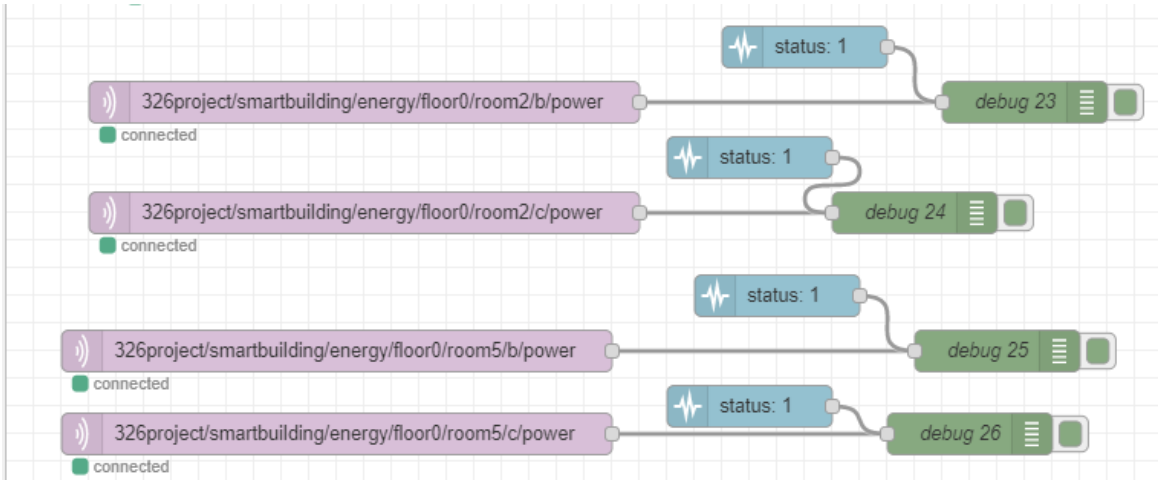


Power consumption of the lights has been divided into two categories, essential and nonessential.

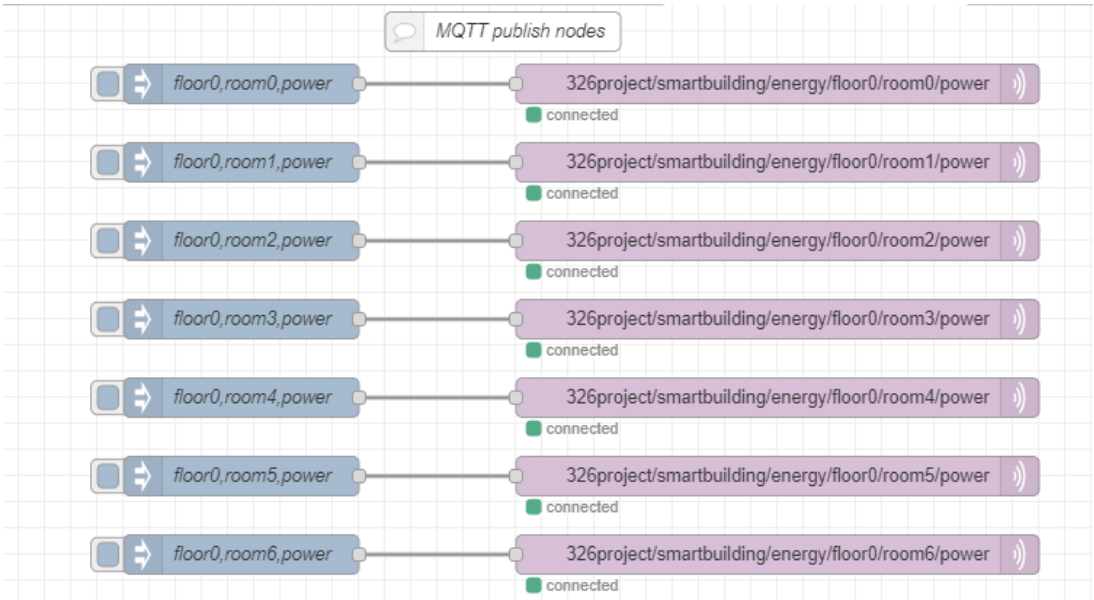
- MQTT ‘publish’ topics correspond to the boilers and chillers of the floor0 - sensors will publish the power consumed by the boilers and chillers of the floor0, to these topics.



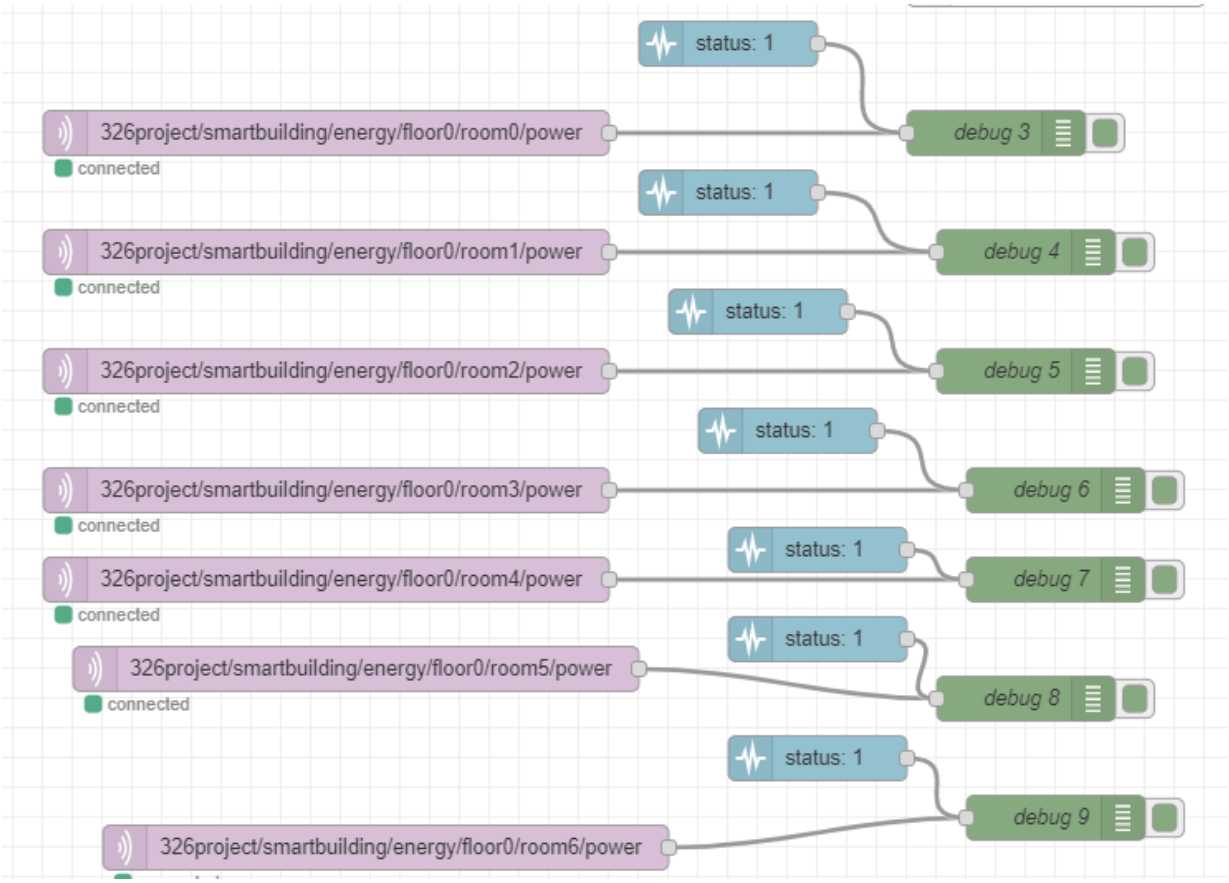
- MQTT ‘subscribe’ topics correspond to the boilers and chillers of the floor0 - clients have to subscribe to these topics.



- MQTT ‘publish’ topics correspond to the total power consumption of the floor0

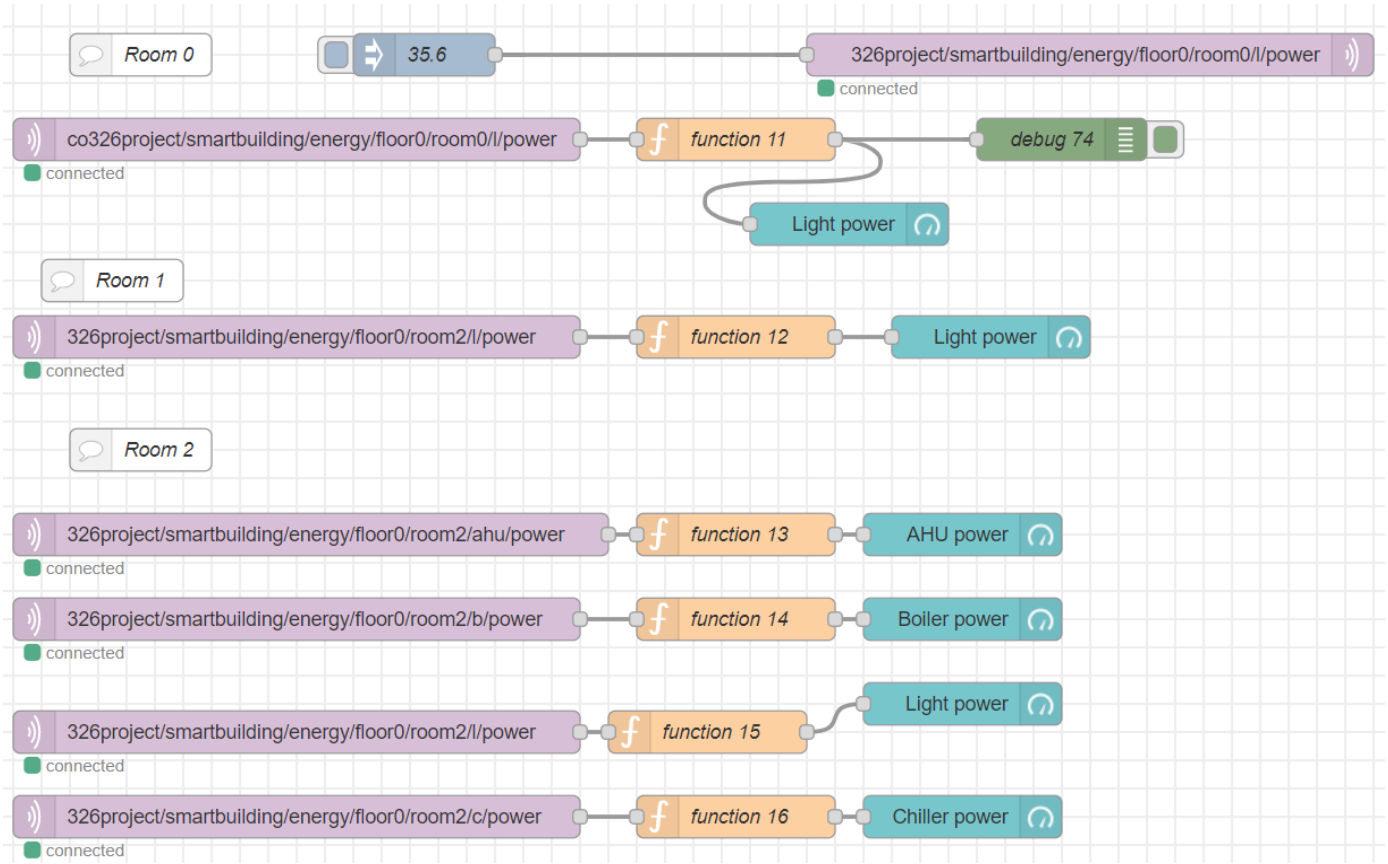


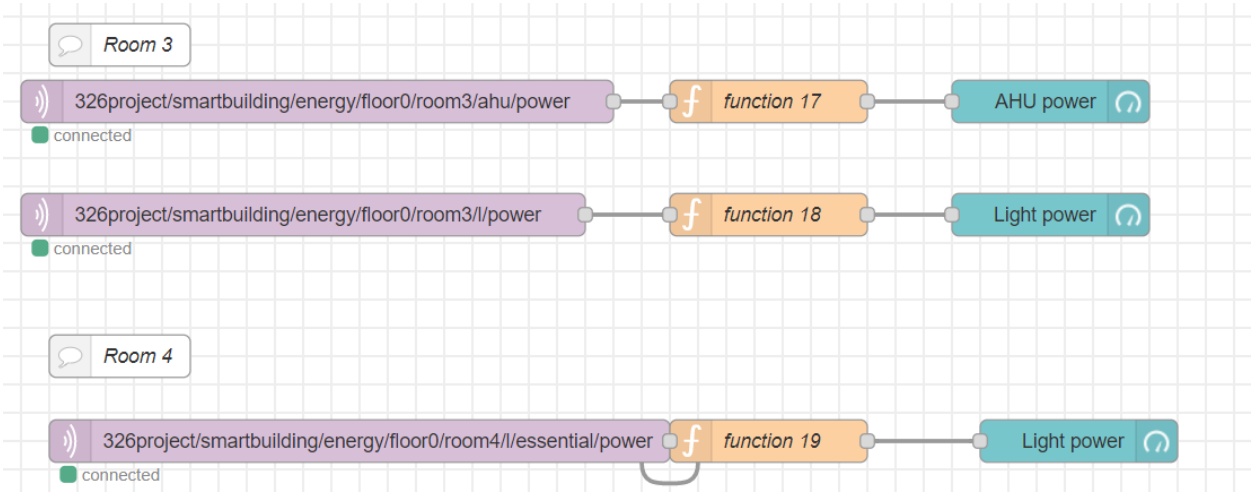
- MQTT ‘subscribe’ topics correspond to the total power consumption of the floor0

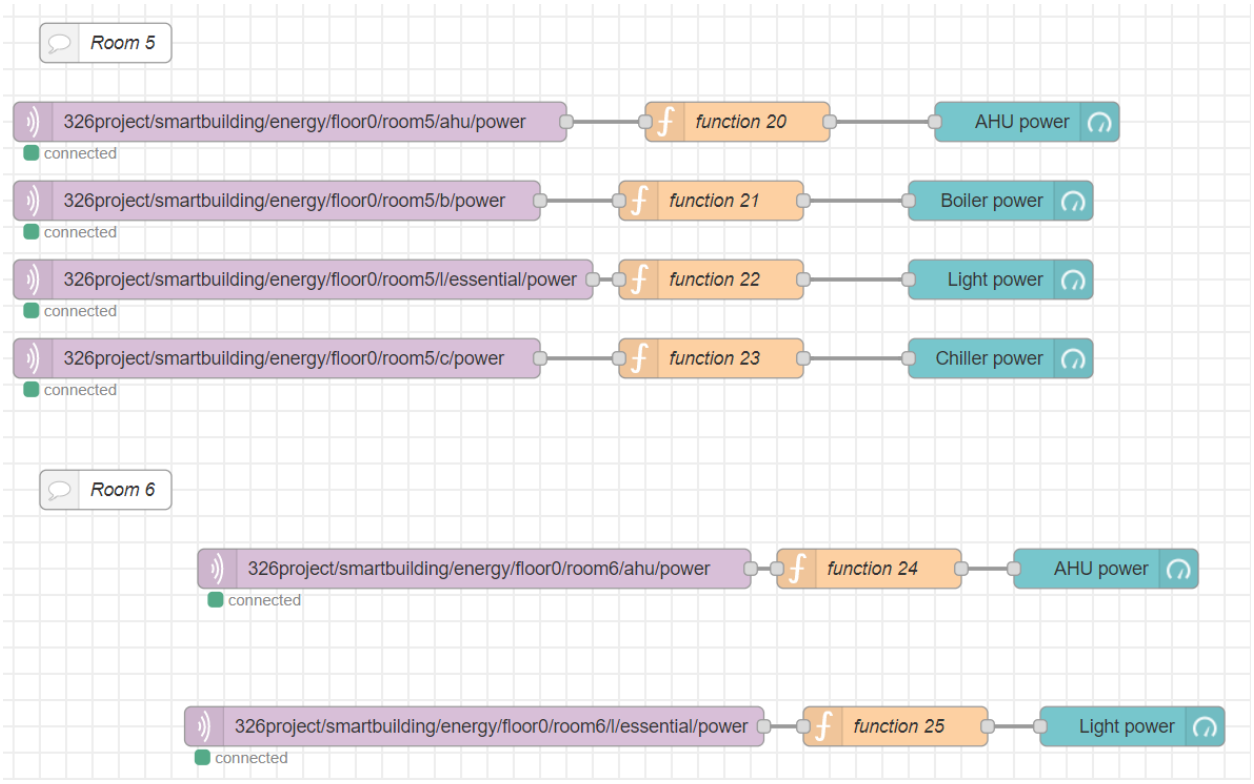


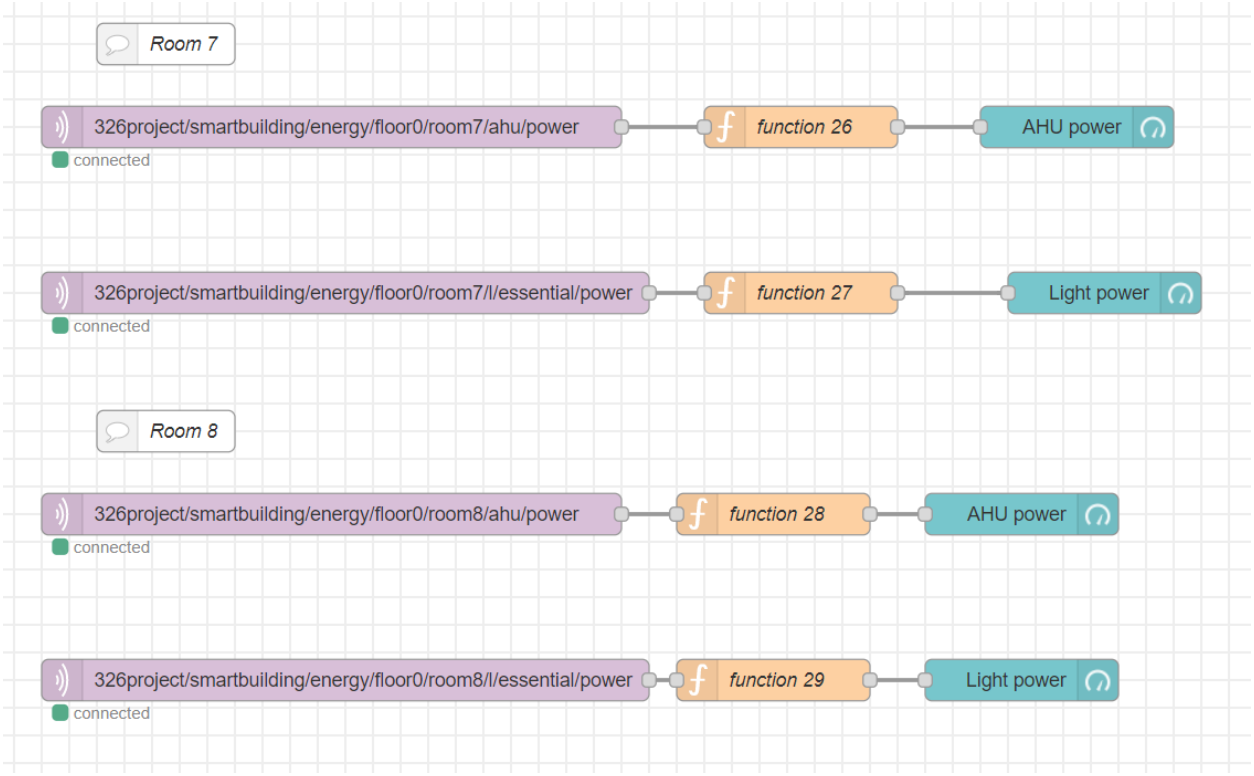
Power Representation of each units of the room in a floor

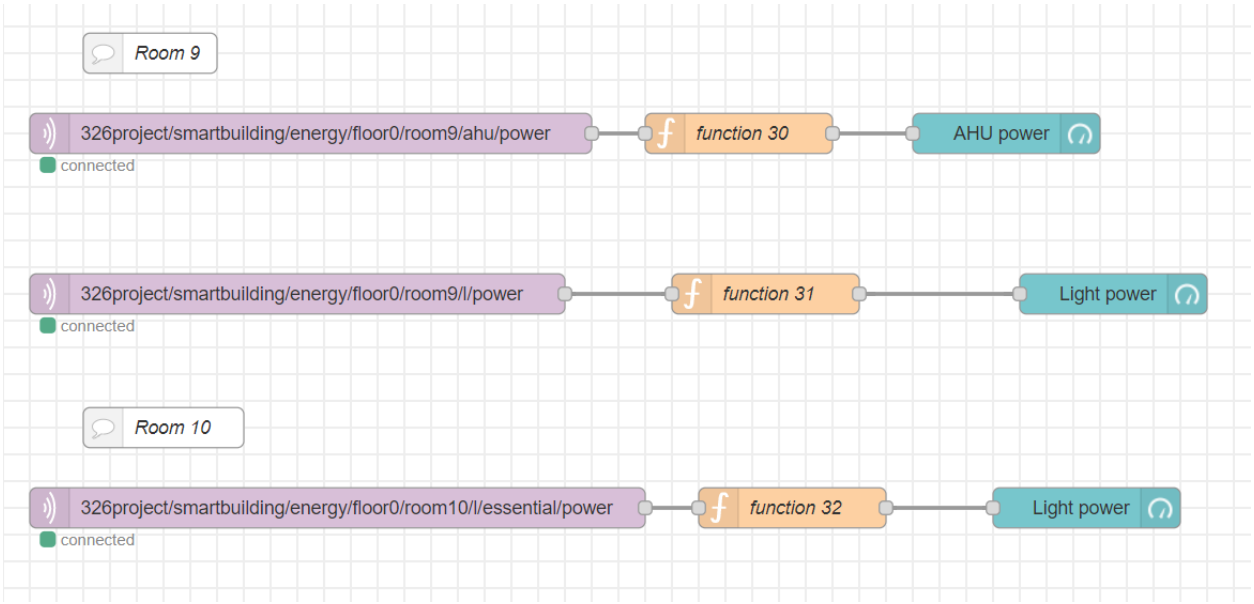
- MQTT ‘publish’ topics correspond to the power of important units like AHU, Chiller,Boiler and Lights of the the room in the floors are given to the gauges to get the readings



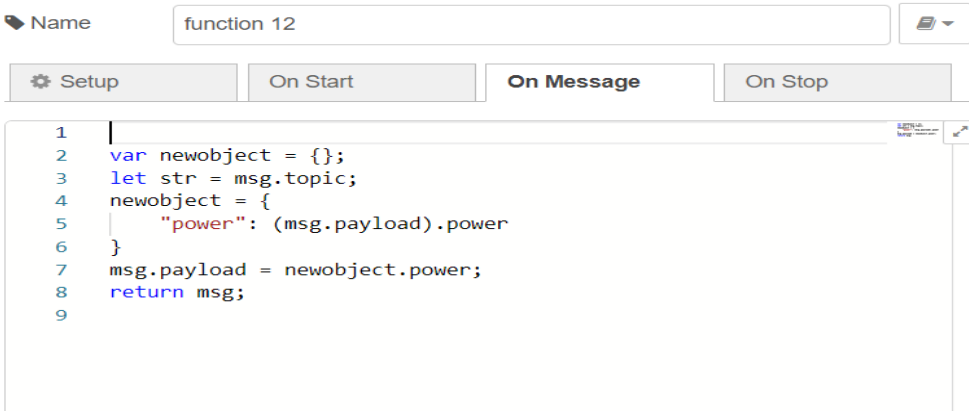



Room 5
326project/smartbuilding/energy/floor0/room5/ahu/power
function 20
AHU power
connected
326project/smartbuilding/energy/floor0/room5/b/power
function 21
Boiler power
connected
326project/smartbuilding/energy/floor0/room5/l/essential/power
function 22
Light power
connected
326project/smartbuilding/energy/floor0/room5/c/power
function 23
Chiller power
connected
Room 6
326project/smartbuilding/energy/floor0/room6/ahu/power
function 24
AHU power
connected
326project/smartbuilding/energy/floor0/room6/l/essential/power
function 25
Light power
connected

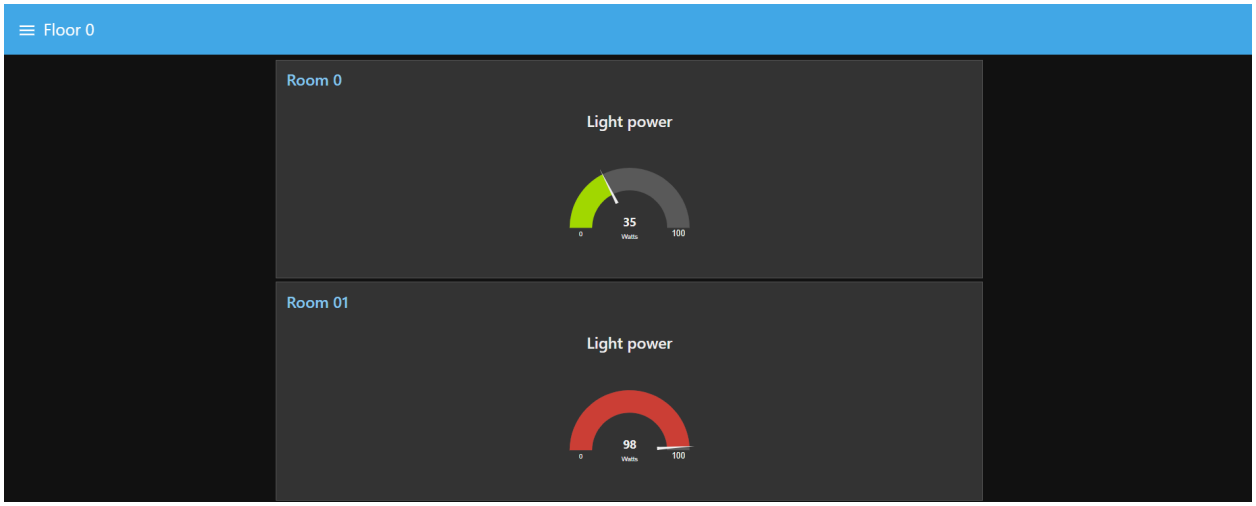

Room 7
326project/smartbuilding/energy/floor0/room7/ahu/power
function 26
AHU power
connected
326project/smartbuilding/energy/floor0/room7/l/essential/power
function 27
Light power
connected
Room 8
326project/smartbuilding/energy/floor0/room8/ahu/power
function 28
AHU power
connected
326project/smartbuilding/energy/floor0/room8/l/essential/power
function 29
Light power
connected

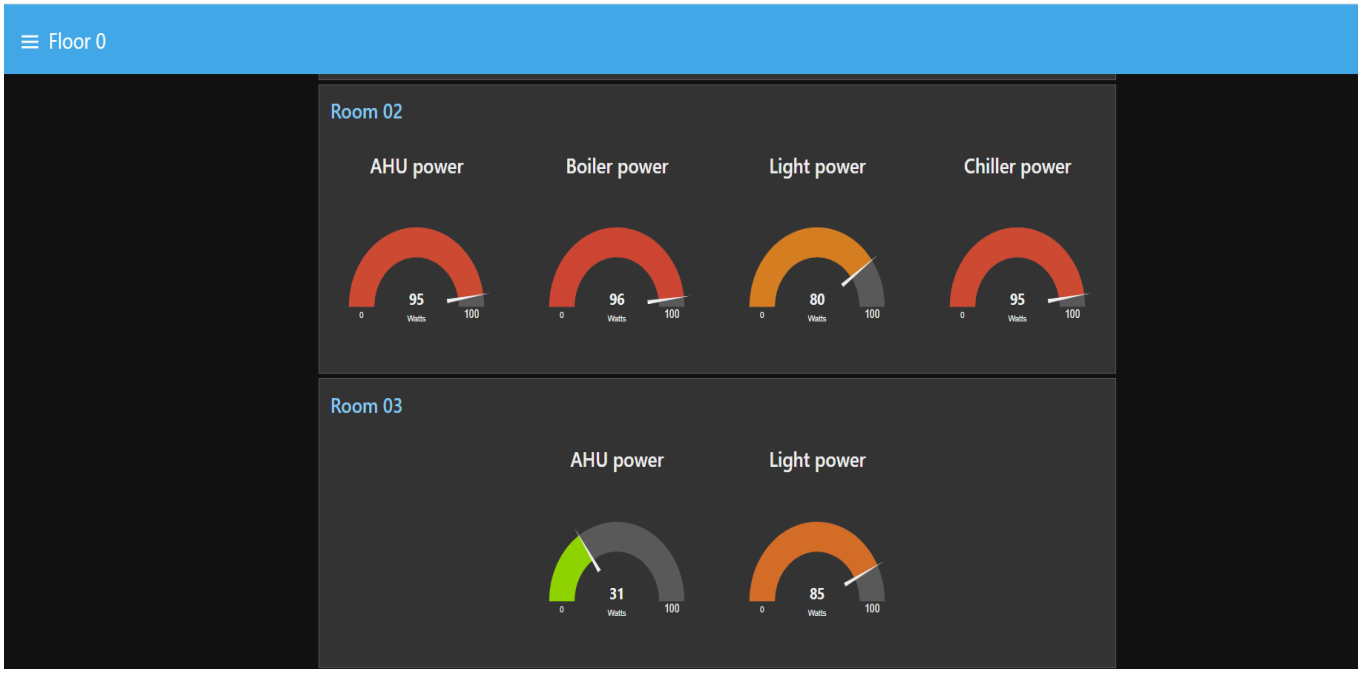


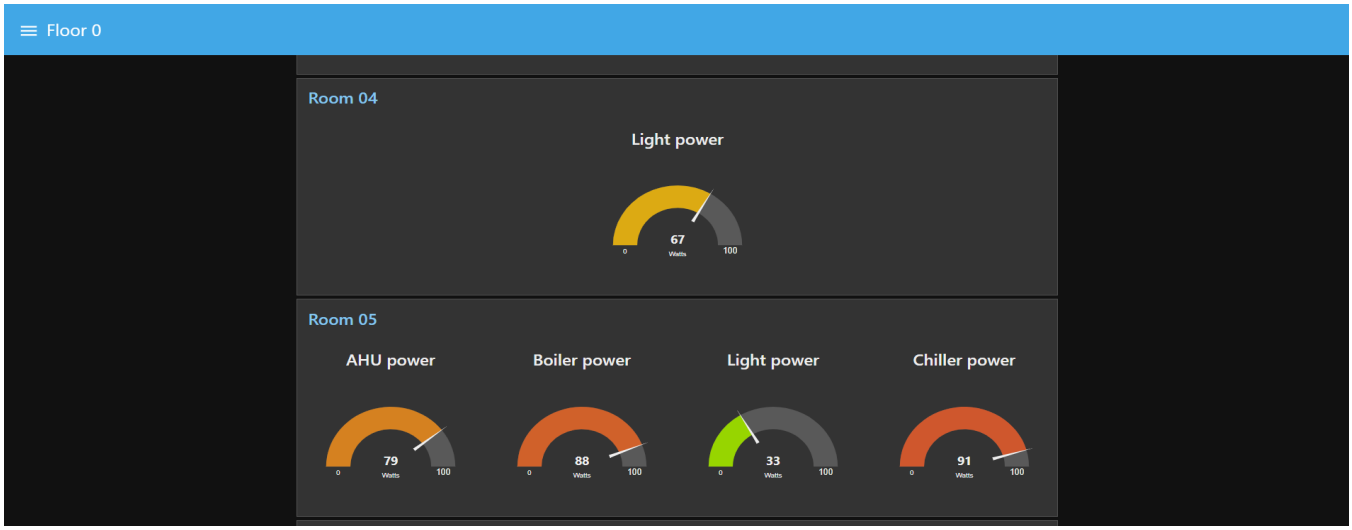
Functions are written to get only the power values from the published values.

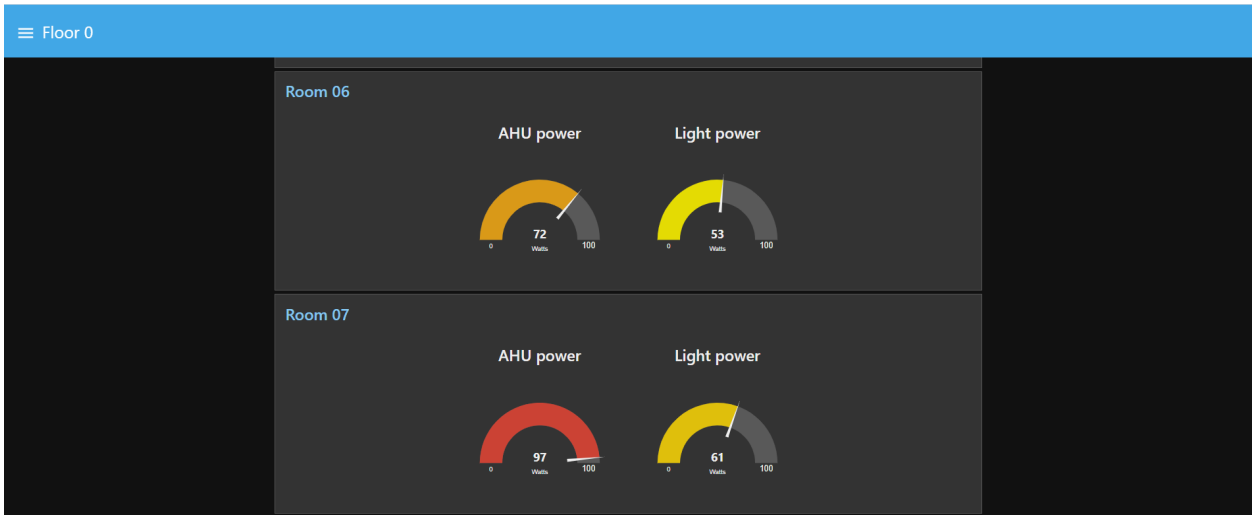


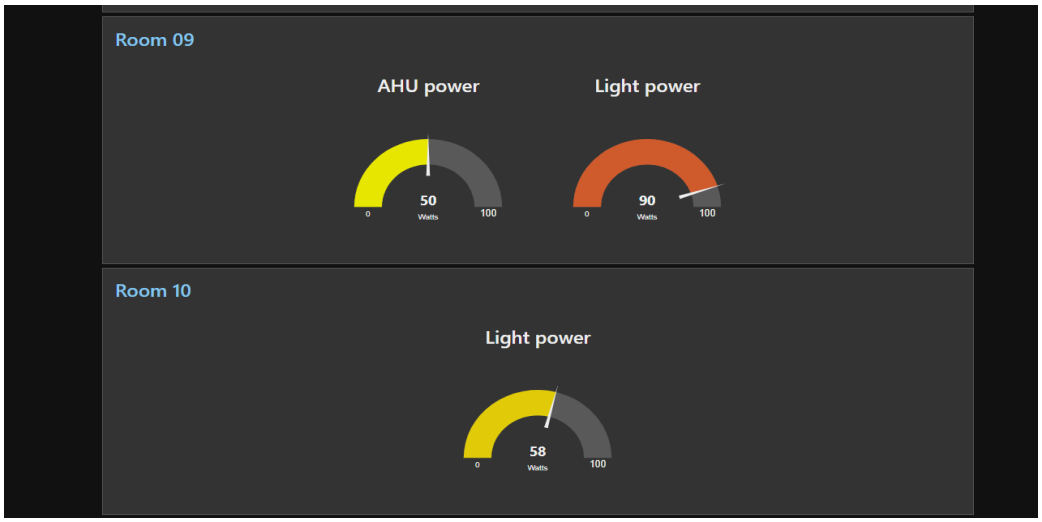
- Readings of the gauges are shown in a dashboard showing them separately corresponding to each room in the floors.


Floor 0
Room 0
Light power
35
Watts
0
100
Room 01
Light power
98
Watts
0
100


Floor 0
Room 02
AHU power
95
Watts
0
100
Boiler power
96
Watts
0
100
Light power
80
Watts
0
100
Chiller power
95
Watts
0
100
Room 03
AHU power
31
Watts
0
100
Light power
85
Watts
0
100


Floor 0
Room 04
Light power
67
0
Watts
100
Room 05
AHU power
79
Boiler power
88
Light power
33
Chiller power
91


Floor 0
Room 06
AHU power
72
Light power
53
Room 07
AHU power
97
Light power
61


Room 09
AHU power
50
Light power
90
Room 10
Light power
58

Sending inputs from SCADA to the MQTT Server

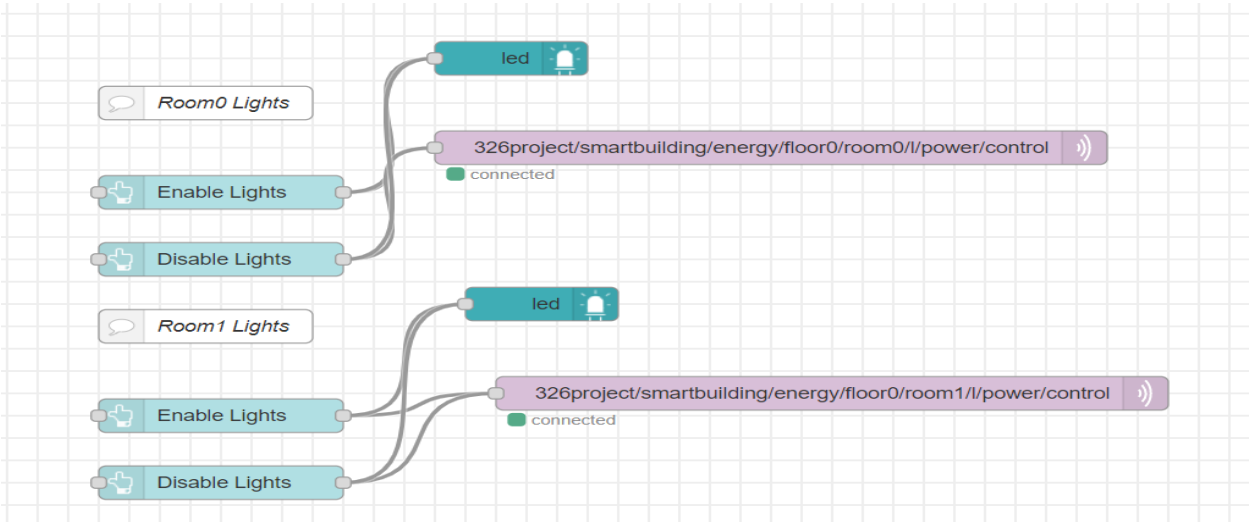
When publishing inputs from SCADA and MQTT server, main parts of the energy sub system are,

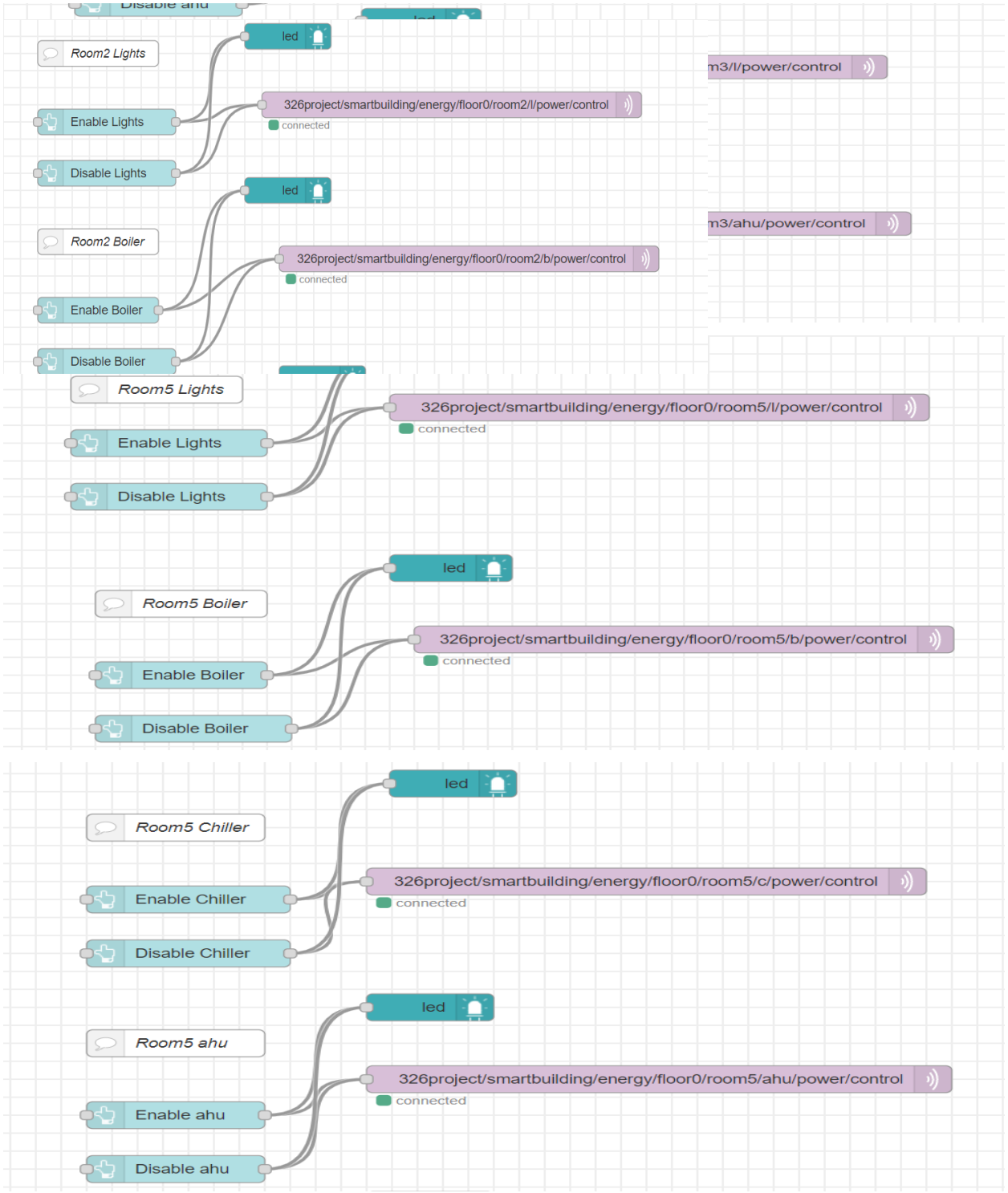
- Lights
- Power
- AHU
- Boiler
- Chiller

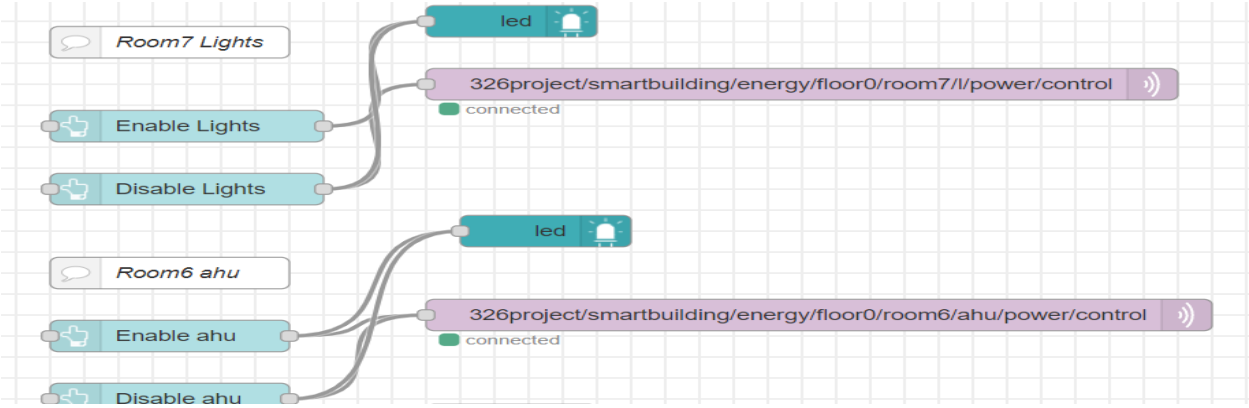
For all the rooms sub parts can be connected and disconnected to the MQTT server. In this process actuators are controlled manually rather than automatically activated by sensor data. We can manually activate the actuators via giving inputs from SCADA.

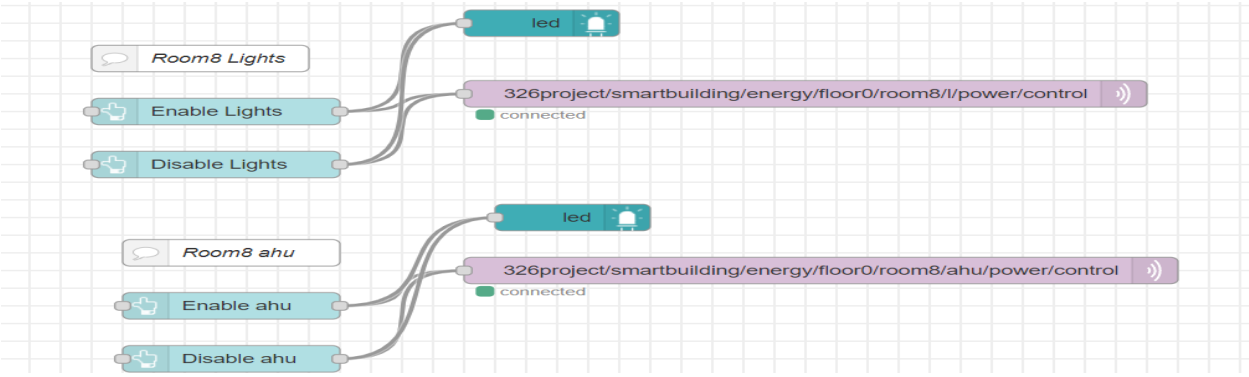
Node-RED implementations:

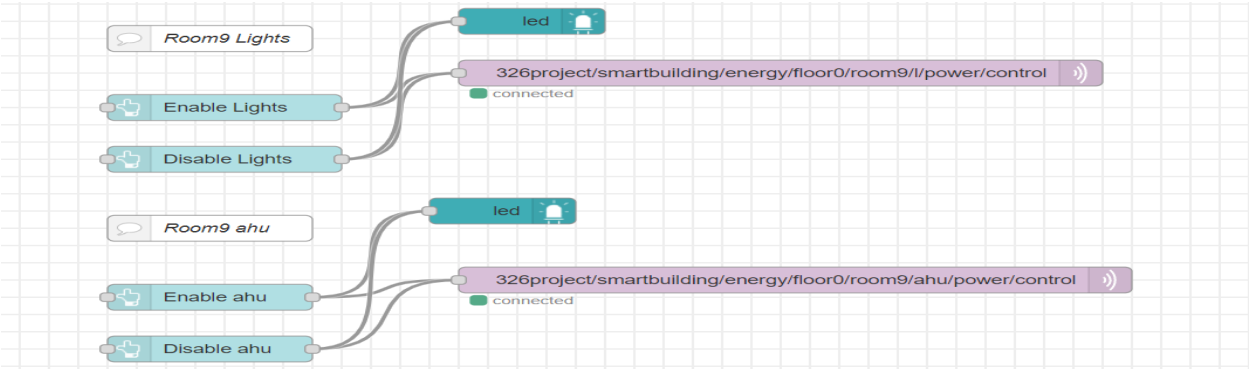
In NODE_RED implementation using enable and disable buttons a payload is published to the MQTT. For enabling the payload is {“state”:1} and for disabling payload is {“state”:0}. Using that payload all the subscribed nodes can be disabled or enabled.

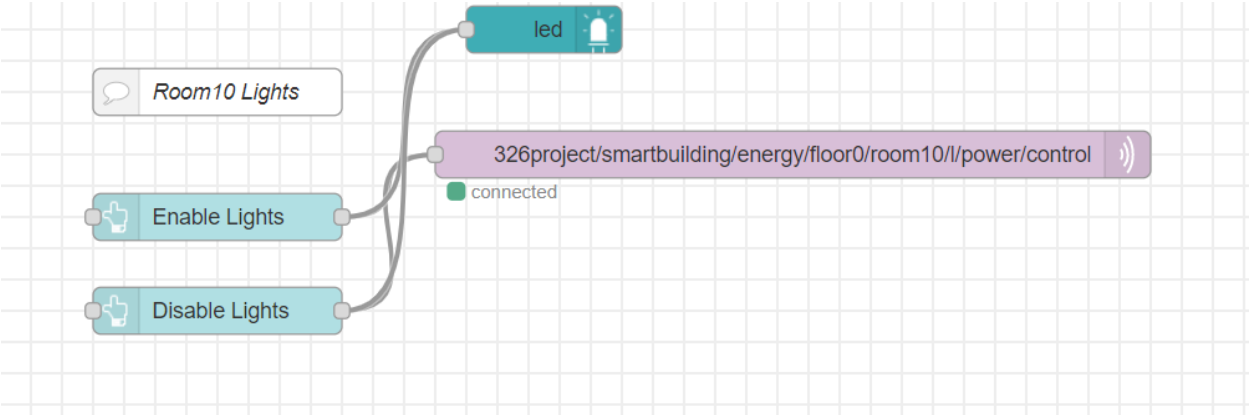



led
Room2 Lights
Enable Lights
Disable Lights
326project/smartbuilding/energy/floor0/room2/l/power/control
connected
led
Room2 Boiler
Enable Boiler
Disable Boiler
326project/smartbuilding/energy/floor0/room2/b/power/control
connected
Room5 Lights
Enable Lights
Disable Lights
326project/smartbuilding/energy/floor0/room5/l/power/control
connected
led
Room5 Boiler
Enable Boiler
Disable Boiler
326project/smartbuilding/energy/floor0/room5/b/power/control
connected
led
Room5 Chiller
Enable Chiller
Disable Chiller
326project/smartbuilding/energy/floor0/room5/c/power/control
connected
led
Room5 ahu
Enable ahu
Disable ahu
326project/smartbuilding/energy/floor0/room5/ahu/power/control
connected


Room7 Lights
Enable Lights
Disable Lights
led
326project/smartbuilding/energy/floor0/room7/l/power/control
connected
Room6 ahu
Enable ahu
Disable ahu
led
326project/smartbuilding/energy/floor0/room6/ahu/power/control
connected


Room8 Lights
Enable Lights
Disable Lights
led
326project/smartbuilding/energy/floor0/room8/l/power/control
connected
Room8 ahu
Enable ahu
Disable ahu
led
326project/smartbuilding/energy/floor0/room8/ahu/power/control
connected

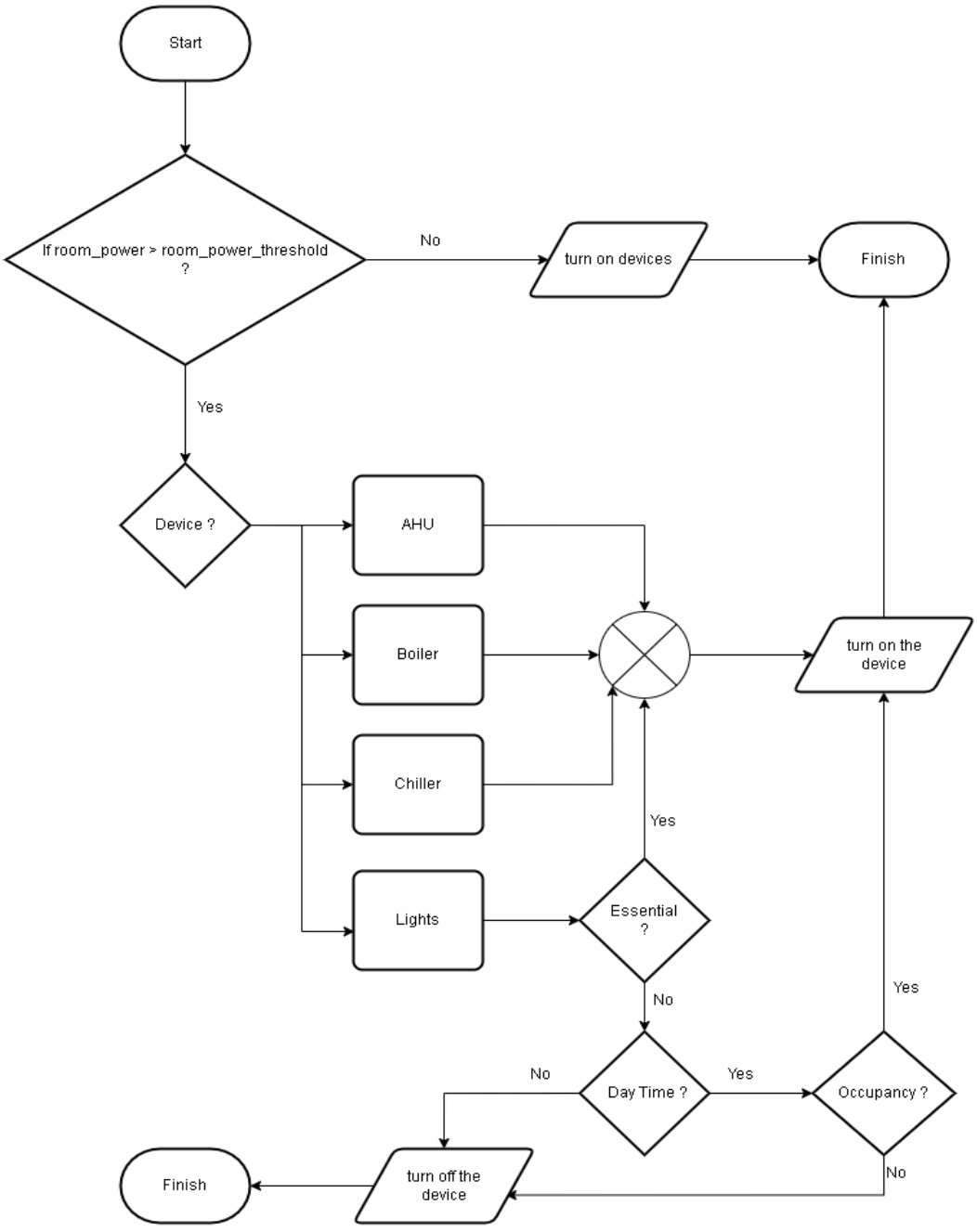

Room9 Lights
Enable Lights
Disable Lights
led
326project/smartbuilding/energy/floor0/room9/l/power/control
connected
Room9 ahu
Enable ahu
Disable ahu
led
326project/smartbuilding/energy/floor0/room9/ahu/power/control
connected


Room10 Lights
Enable Lights
Disable Lights
led
326project/smartbuilding/energy/floor0/room10/l/power/control
connected

Dashboard

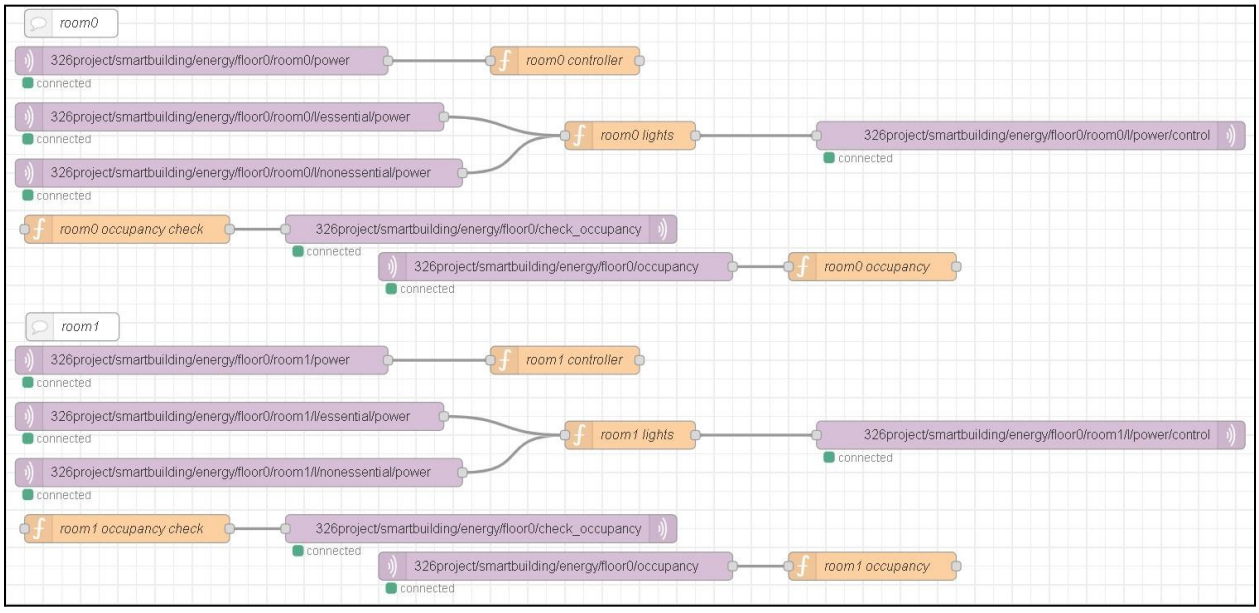
Using this SCADA interface all the inputs are given

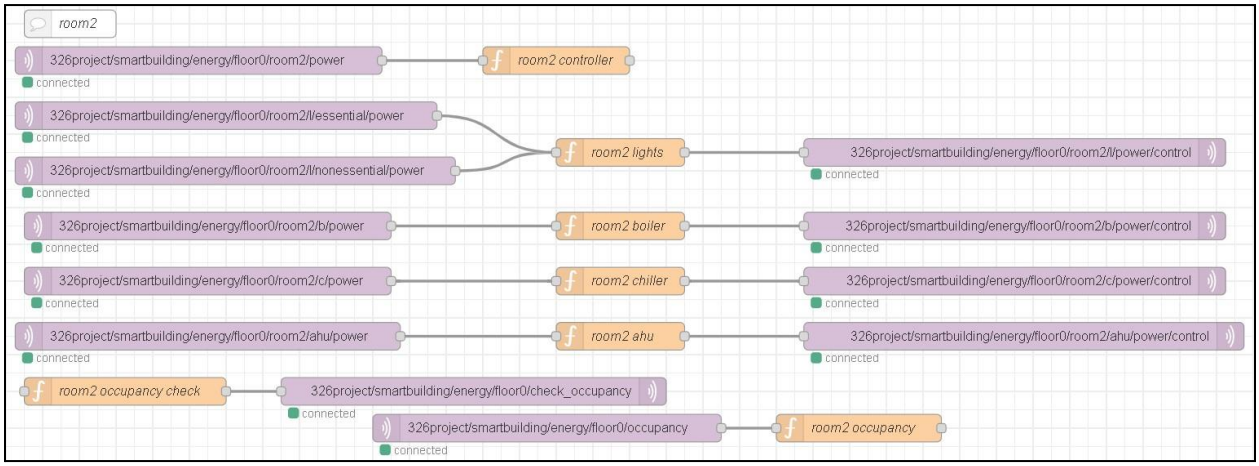
Process controller with operating and optimizing process and algorithms

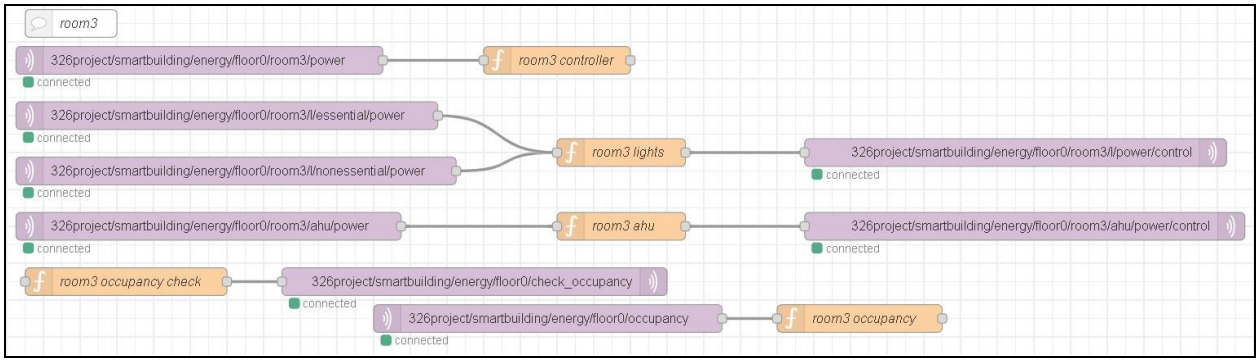


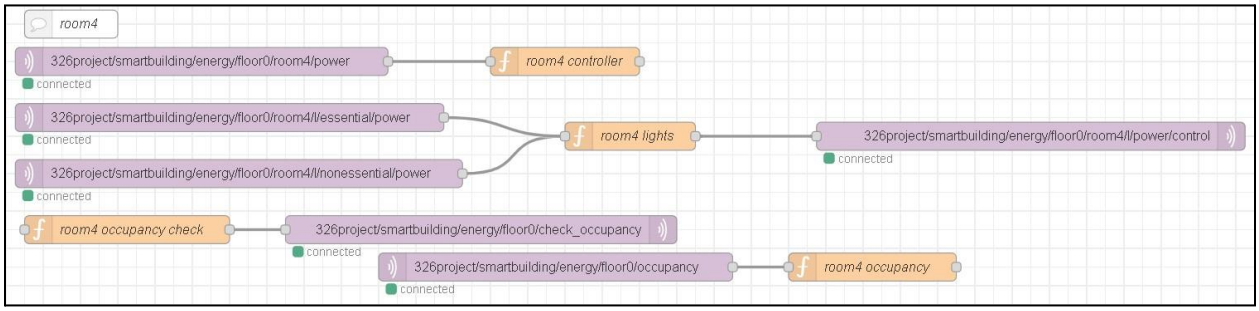
It is expected to reduce the over power consumption in this building using this algorithm. In this case, only floor0 has been considered for demonstration and likewise all the floors in the building should be considered as well. First, for each room, it is checked whether the room power consumption exceeds the threshold value. If it does not exceed, all devices inside the room can be powered on. Otherwise, essential devices are powered on and for non essentials, decisions are taken depending on time and occupancy. When considering non essentials, all non essentials are turned off during night time and at day time they are turned off if nobody is inside the room. Otherwise, if people are inside the room at day time, then non essentials should be turned on.

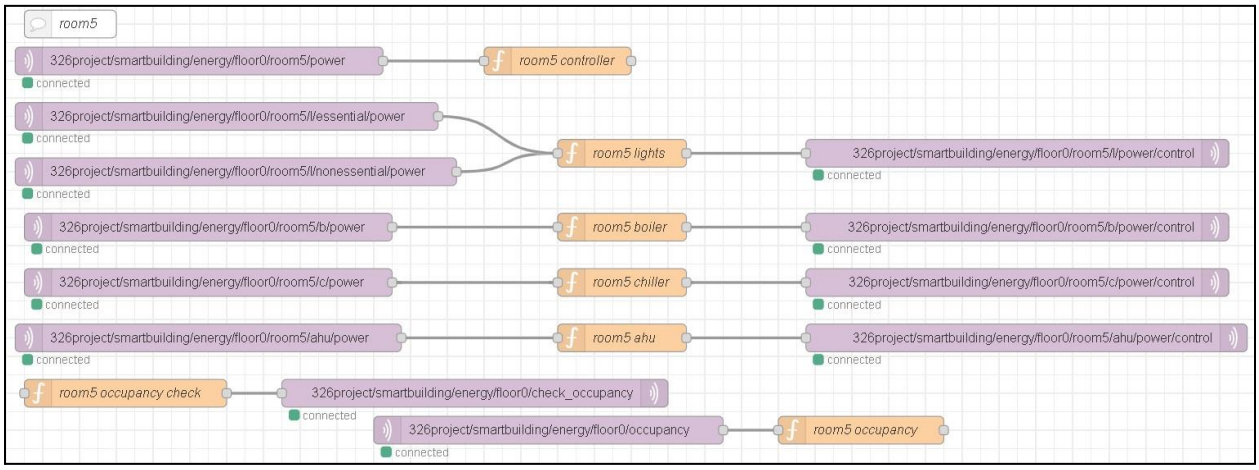
NodeRed Implementation for each room

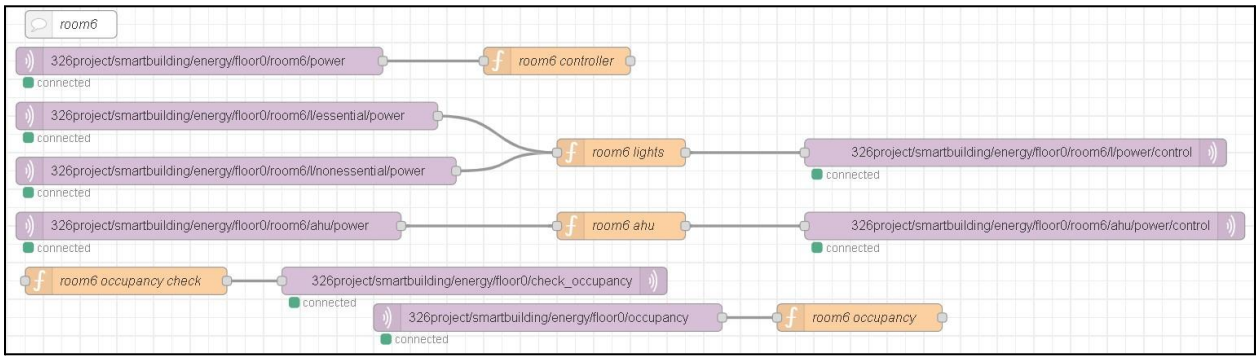


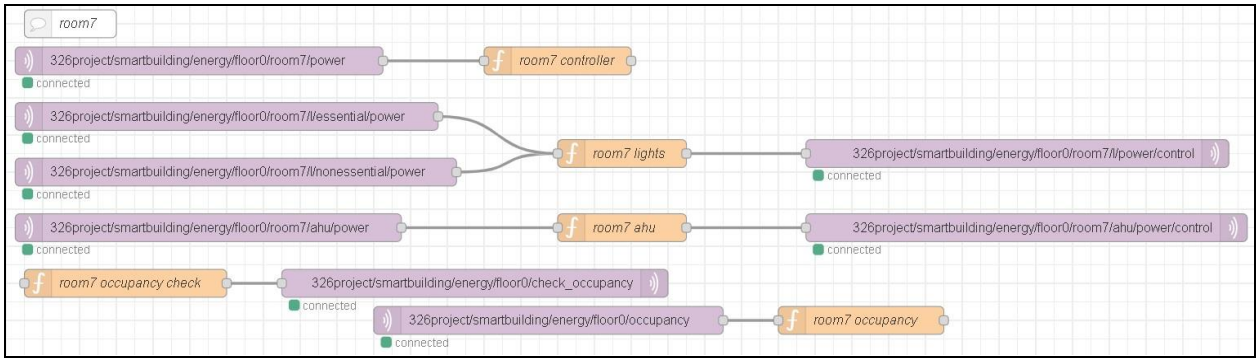


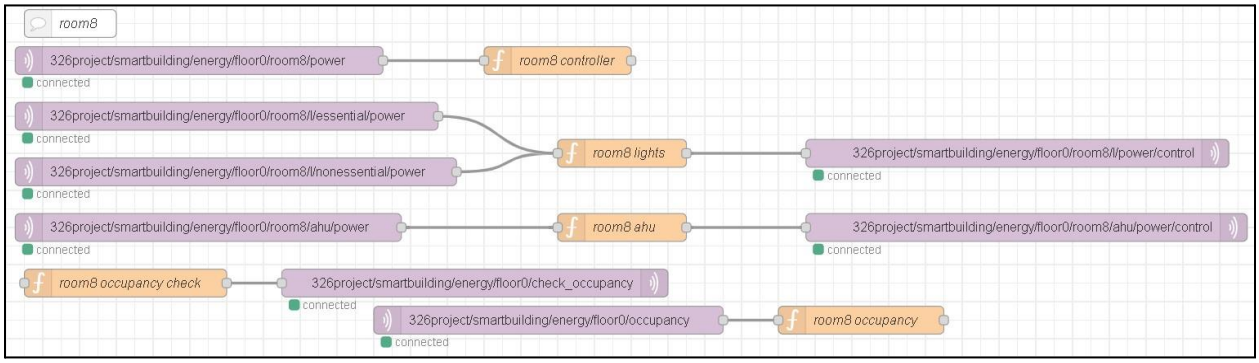

room3
326project/smartbuilding/energy/floor0/room3/power
connected
room3 controller
326project/smartbuilding/energy/floor0/room3/l/essential/power
connected
326project/smartbuilding/energy/floor0/room3/l/nonessential/power
connected
room3 lights
326project/smartbuilding/energy/floor0/room3/l/power/control
connected
326project/smartbuilding/energy/floor0/room3/ahu/power
connected
room3 ahu
326project/smartbuilding/energy/floor0/room3/ahu/power/control
connected
room3 occupancy check
326project/smartbuilding/energy/floor0/check_occupancy
connected
326project/smartbuilding/energy/floor0/occupancy
connected
room3 occupancy

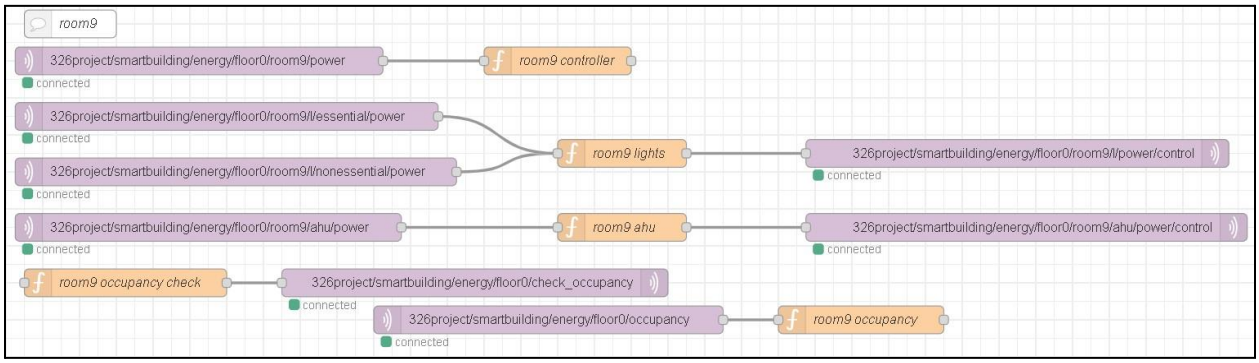

room4
326project/smartbuilding/energy/floor0/room4/power
connected
room4 controller
326project/smartbuilding/energy/floor0/room4/l/essential/power
connected
326project/smartbuilding/energy/floor0/room4/l/nonessential/power
connected
room4 lights
326project/smartbuilding/energy/floor0/room4/l/power/control
connected
room4 occupancy check
326project/smartbuilding/energy/floor0/check_occupancy
connected
326project/smartbuilding/energy/floor0/occupancy
connected
room4 occupancy

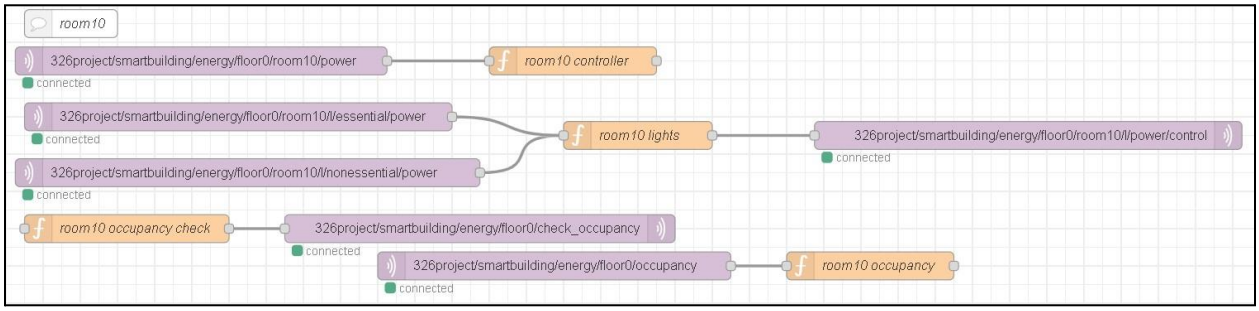

room5
326project/smartbuilding/energy/floor0/room5/power
connected
room5 controller
326project/smartbuilding/energy/floor0/room5/l/essential/power
connected
326project/smartbuilding/energy/floor0/room5/l/nonessential/power
connected
room5 lights
326project/smartbuilding/energy/floor0/room5/l/power/control
connected
326project/smartbuilding/energy/floor0/room5/b/power
connected
room5 boiler
326project/smartbuilding/energy/floor0/room5/b/power/control
connected
326project/smartbuilding/energy/floor0/room5/c/power
connected
room5 chiller
326project/smartbuilding/energy/floor0/room5/c/power/control
connected
326project/smartbuilding/energy/floor0/room5/ahu/power
connected
room5 ahu
326project/smartbuilding/energy/floor0/room5/ahu/power/control
connected
room5 occupancy check
326project/smartbuilding/energy/floor0/check_occupancy
connected
326project/smartbuilding/energy/floor0/occupancy
connected
room5 occupancy


room6
326project/smartbuilding/energy/floor0/room6/power
connected
room6 controller
326project/smartbuilding/energy/floor0/room6/l/essential/power
connected
326project/smartbuilding/energy/floor0/room6/l/nonessential/power
connected
room6 lights
326project/smartbuilding/energy/floor0/room6/l/power/control
connected
326project/smartbuilding/energy/floor0/room6/ahu/power
connected
room6 ahu
326project/smartbuilding/energy/floor0/room6/ahu/power/control
connected
room6 occupancy check
326project/smartbuilding/energy/floor0/check_occupancy
connected
326project/smartbuilding/energy/floor0/occupancy
connected
room6 occupancy


room7
326project/smartbuilding/energy/floor0/room7/power
connected
room7 controller
326project/smartbuilding/energy/floor0/room7/l/essential/power
connected
326project/smartbuilding/energy/floor0/room7/l/nonessential/power
connected
room7 lights
326project/smartbuilding/energy/floor0/room7/l/power/control
connected
326project/smartbuilding/energy/floor0/room7/ahu/power
connected
room7 ahu
326project/smartbuilding/energy/floor0/room7/ahu/power/control
connected
room7 occupancy check
326project/smartbuilding/energy/floor0/check_occupancy
connected
326project/smartbuilding/energy/floor0/occupancy
connected
room7 occupancy


room8
326project/smartbuilding/energy/floor0/room8/power
connected
room8 controller
326project/smartbuilding/energy/floor0/room8/l/essential/power
connected
326project/smartbuilding/energy/floor0/room8/l/nonessential/power
connected
room8 lights
326project/smartbuilding/energy/floor0/room8/l/power/control
connected
326project/smartbuilding/energy/floor0/room8/ahu/power
connected
room8 ahu
326project/smartbuilding/energy/floor0/room8/ahu/power/control
connected
room8 occupancy check
326project/smartbuilding/energy/floor0/check_occupancy
connected
326project/smartbuilding/energy/floor0/occupancy
connected
room8 occupancy


room9
326project/smartbuilding/energy/floor0/room9/power
connected
room9 controller
326project/smartbuilding/energy/floor0/room9/l/essential/power
connected
326project/smartbuilding/energy/floor0/room9/l/nonessential/power
connected
room9 lights
326project/smartbuilding/energy/floor0/room9/l/power/control
connected
326project/smartbuilding/energy/floor0/room9/ahu/power
connected
room9 ahu
326project/smartbuilding/energy/floor0/room9/ahu/power/control
connected
room9 occupancy check
326project/smartbuilding/energy/floor0/check_occupancy
connected
326project/smartbuilding/energy/floor0/occupancy
connected
room9 occupancy


room10
326project/smartbuilding/energy/floor0/room10/power
connected
room10 controller
326project/smartbuilding/energy/floor0/room10/l/essential/power
connected
326project/smartbuilding/energy/floor0/room10/l/nonessential/power
connected
room10 lights
326project/smartbuilding/energy/floor0/room10/l/power/control
connected
room10 occupancy check
326project/smartbuilding/energy/floor0/check_occupancy
connected
326project/smartbuilding/energy/floor0/occupancy
connected
room10 occupancy

Database

Basically there are two sensors in the energy part.Such as Current sensor and voltage sensor.When these sensors measure the current value and voltage value,we can measure the power for a specific actuator by using p=vi function.Then that power value will published to corresponding topic as a object with another details such as sensor_name,floor_no,room_no,time,power.In this database part, that object has been subscribed by the same topic and it has stored in mongoDB database system as a object.Figure 1.0 shows,how we have implemented database and how save sensor data in database using Node-red.

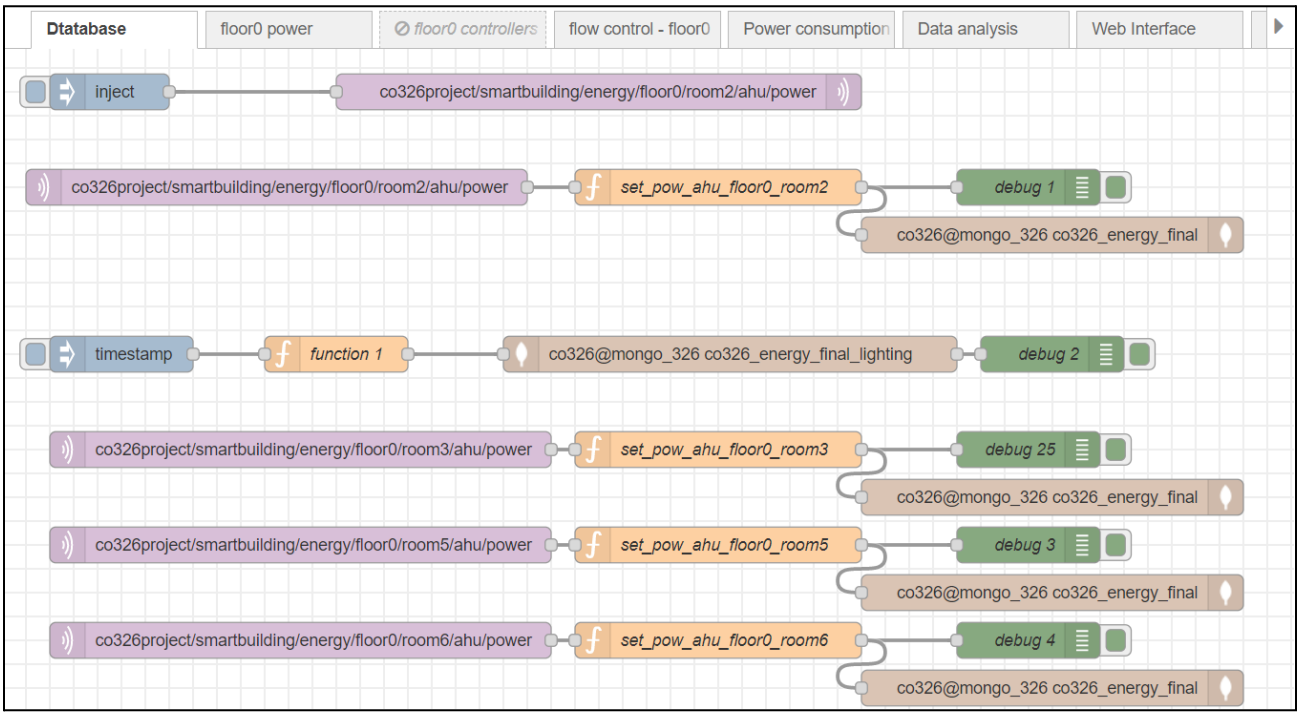


Figure 1.0

There are basically two collections under energy such as co326_energy_final and co326_energy_final_lighting.We have consider boiler,chiller,ahu are essential units and lights may essential or non-essential units.co326_energy_final collection consists with sensor_name,floor_no,room_no,time and power attributes for boiler,chiller and ahu.figure 2.0 shows saving data object and attributes for boiler,chiller and ahu.Also co326_energy_final_lighting collection consists with sensor_name,floor_no,room_no ,state,time and power for lights.Using the state attribute we can identify whether it is essential light or non-essential light.figure 2.1 shows saving data object and attributes for light.

Figure 2.0

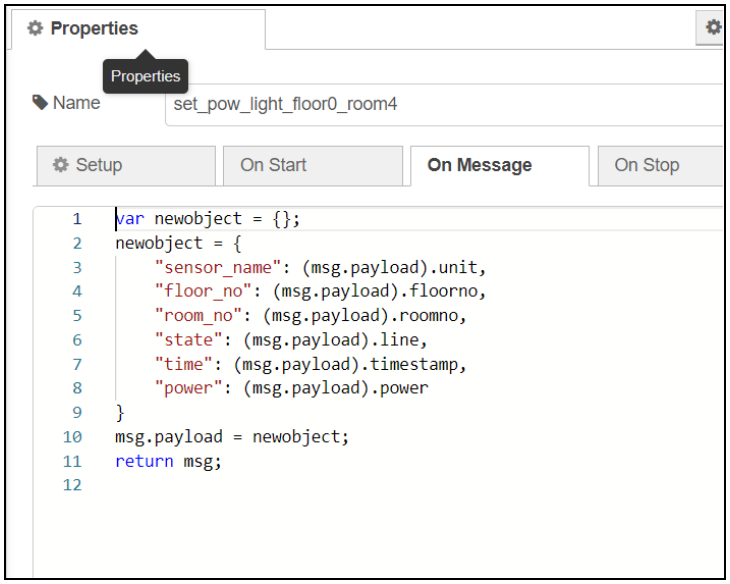
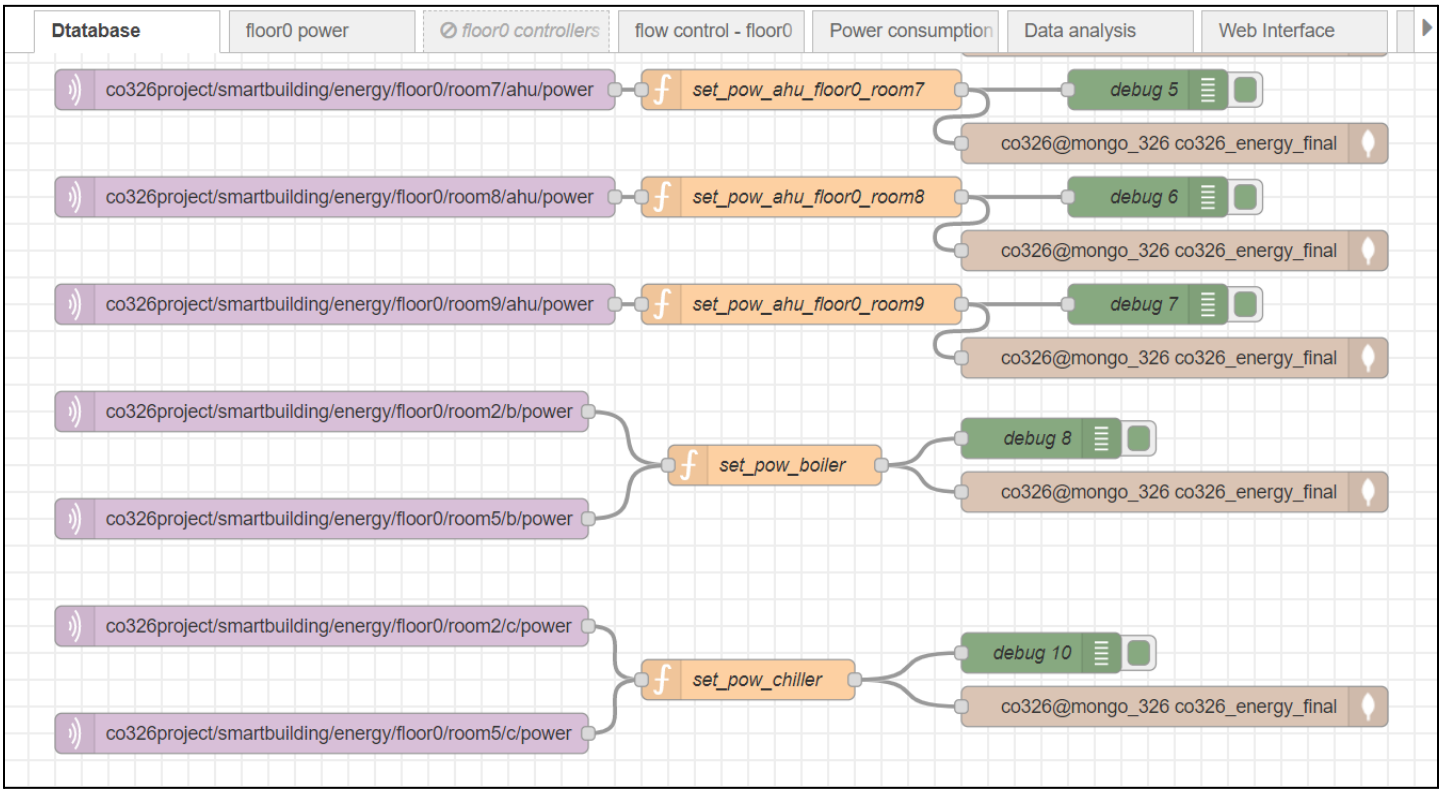
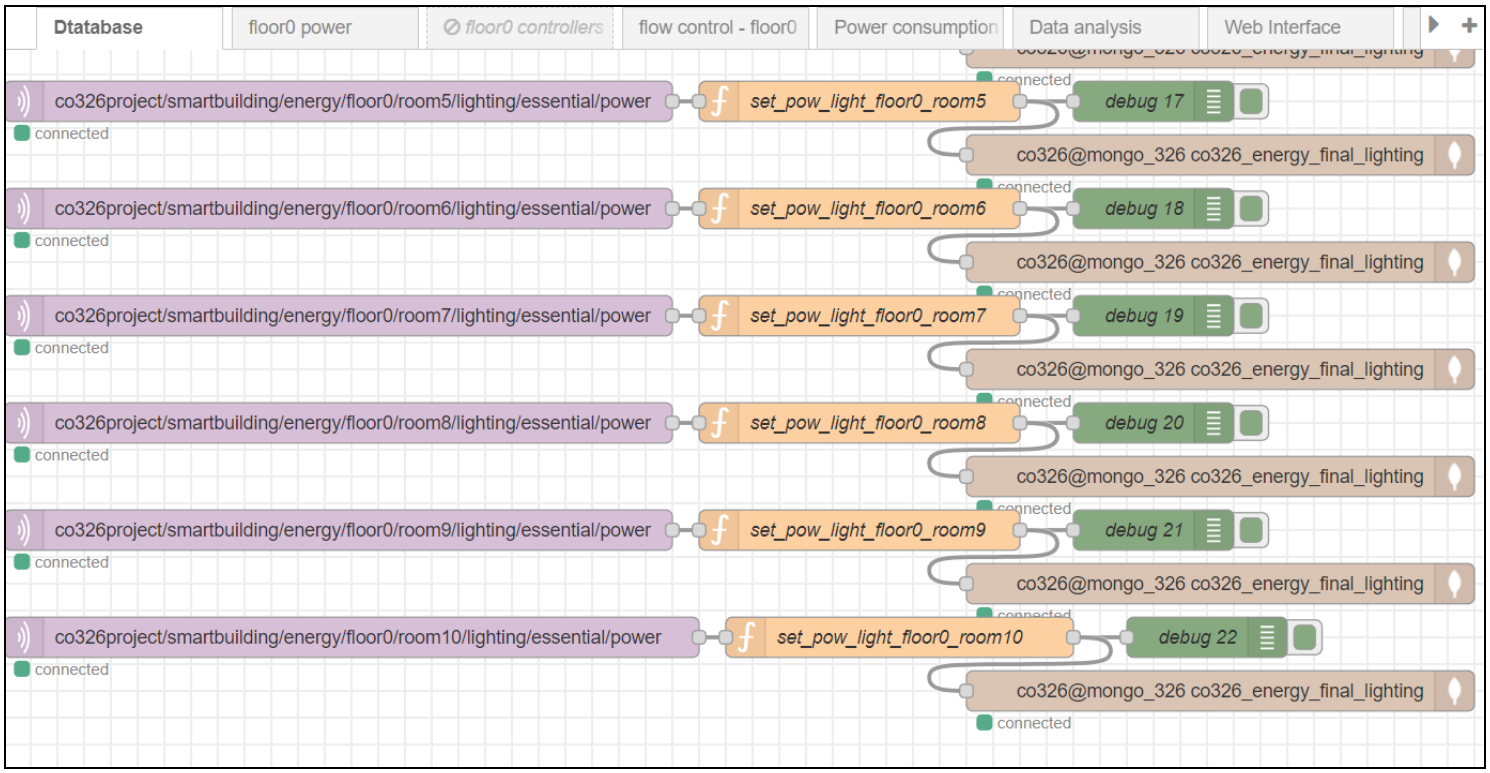
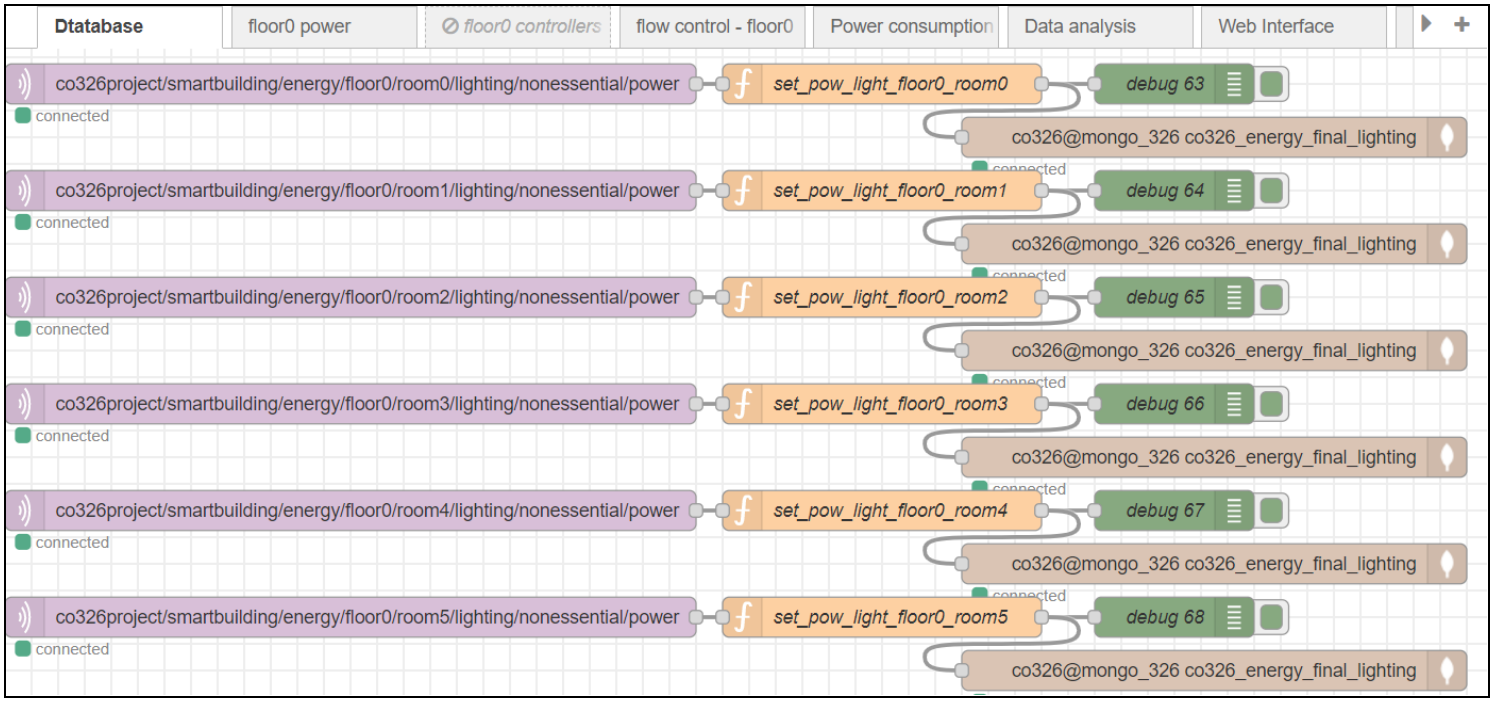


Figure 2.1







Finally this stored data, can be used for data analysis.for example an energy controlling unit needs a threshold value for a room,then this recorded data can be used to calculate that value.

Data Analysis

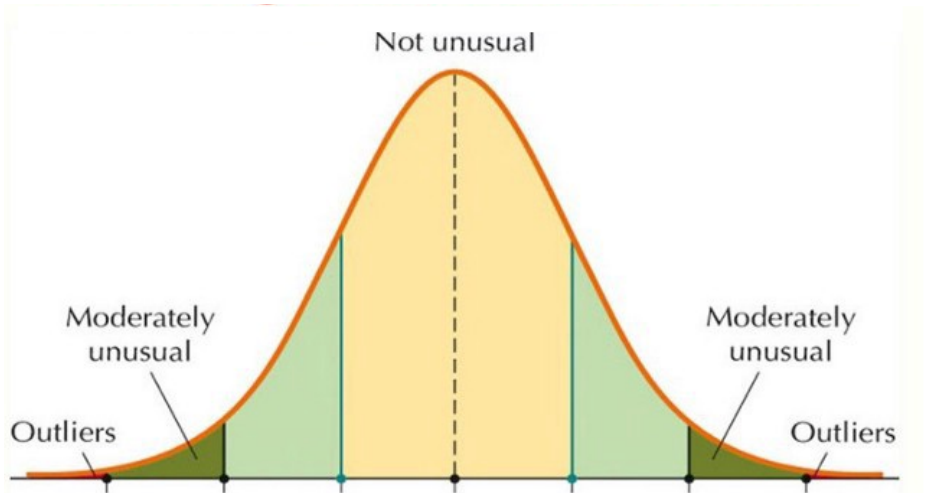
Energy optimization

- Get data from database
  For the analysis of each collection, sufficient records should be there in the collection in order to analyze correctly. Therefore after the collection related to a relevant topic is filled with at least 10 records,all records are analyzed.

```
for(let i=0; i<msg.payload.length;++i){
    if (typeof msg.payload[i].payload !== 'undefined') {
        if (msg.payload[i].payload.floor_no === "floor0") {
            arr.push(msg.payload[i].payload.power);
        }
    }

}
if (arr.length < 10) {
    msg.payload = Infinity;
    return msg;
}
```

- Calculate threshold
  Basic idea of calculating the threshold value is to provide a parameter which can be used to control actuators.
  In this case, values which is larger than mean + 1.0* standard deviation are considered as outliers. Therefore upon receiving value bigger than that, actuators should be controlled to lower the energy consumption.



threshold = mean + 1.5 * std

```
if (arr.length < 10) {
    msg.payload = Infinity;
    return msg;
}
//handle insufficient data

let mean = arr.reduce((acc, curr) => {
    return acc + curr
}, 0) / arr.length;

// Assigning (value - mean) ^ 2 to every array item
arr = arr.map((k) => {
    return (k - mean) ** 2
})

// Calculating the sum of updated array
let sum = arr.reduce((acc, curr) => acc + curr, 0);

// Calculating the variance
let variance = sum / arr.length

// Returning the Standard deviation
let std = Math.sqrt(sum / arr.length)
let thresh = mean + 1.0 * std;
```

- Store threshold value in a global variable

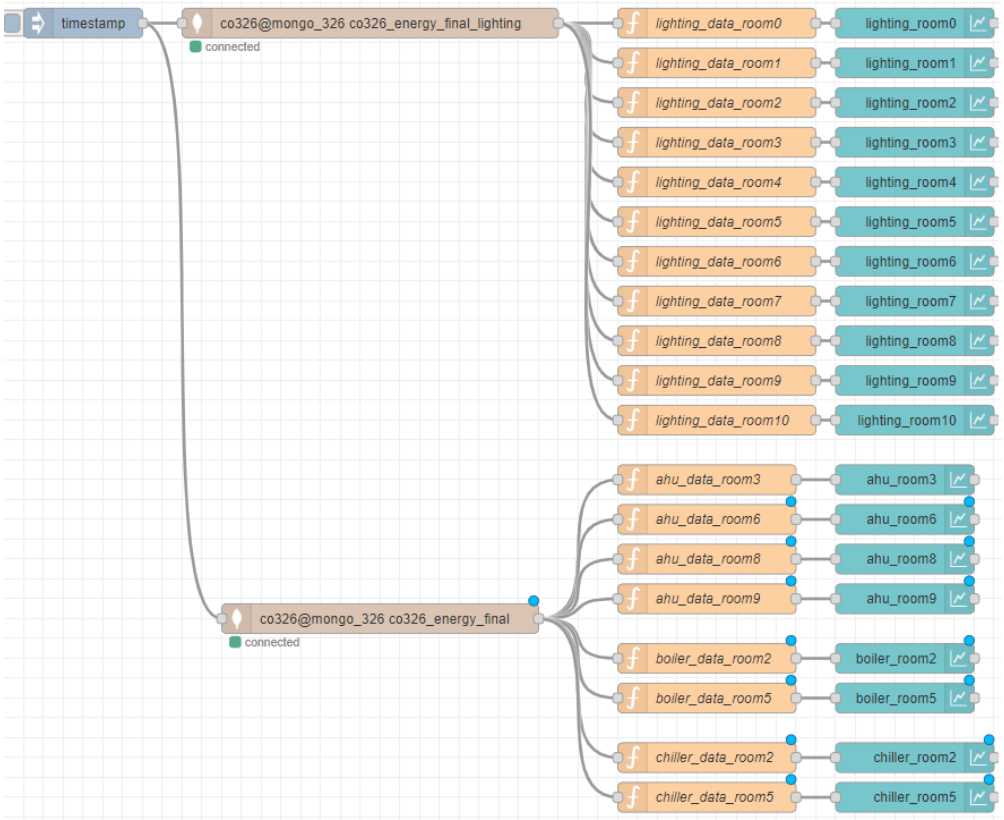
```
global.set('floor0_thresh',msg.payload)
```

  These values change when database entries are updated/new entries are inserted.

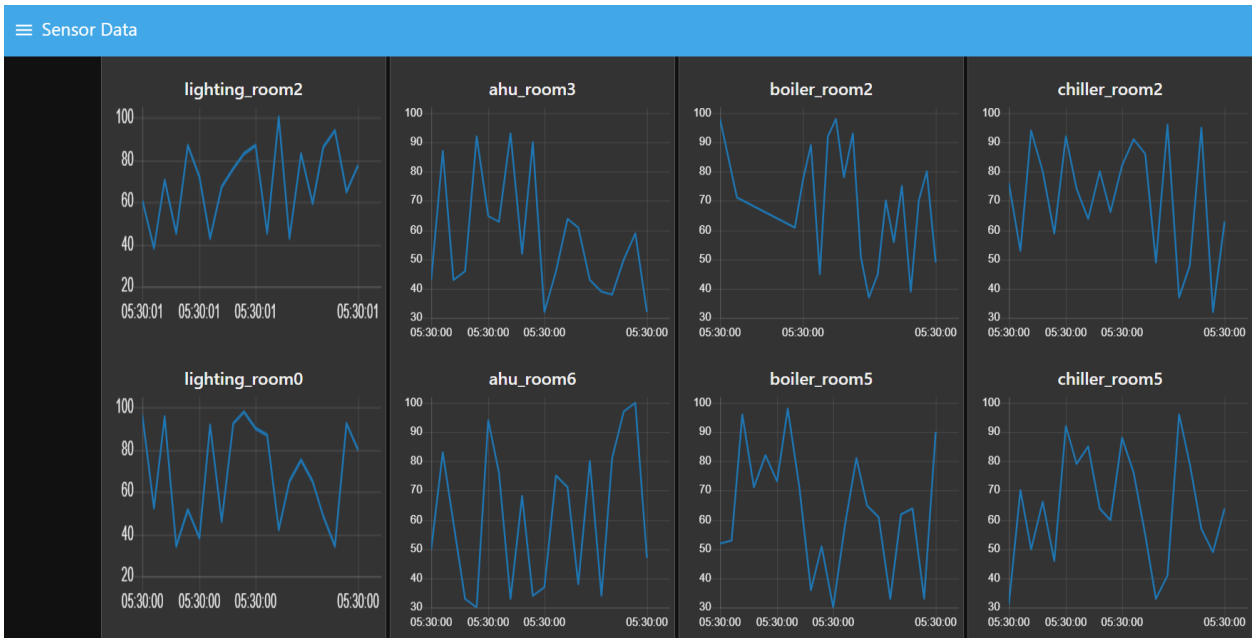
Display database data in graphs

Data from 2 database collections, co326_energy_final and co326_energy_final_lighting are displayed in graphs on the node-RED dashboard for data visualization in order to do necessary improvements. Separate graphs are created for each unit for which the sensors are connected, for each room. Graphs show the power consumption against time.

Node-RED flow;



Node-RED dashboard;


Sensor Data
lighting_room2
ahu_room3
boiler_room2
chiller_room2
lighting_room0
ahu_room6
boiler_room5
chiller_room5

Reference

1. https://www.ibm.com/docs/en/tnpm/1.4.1?topic=thresholds-configuring-baseline
2. https://www.aceee.org/sites/default/files/publications/researchreports/a1701.pdf

3. Blinking an LED With ESP32 : 3 Steps - Instructables
4. Smart Electricity Meter With Energy Monitoring And Feedback Syste... (electronicwings.com)
5. ESP32 MQTT client: Publish and Subscribe. HiveMQ and BME280 example (survivingwithandroid.com)