University of Peradeniya
Faculty of Engineering
Department of Computer Engineering

CO326: Computer Systems Engineering, Industrial Networks

Project Smart Building: Group-G (Control)

October 2022

Abstract

Smart building is the project component for the 6th-semester course CO326 in the computer engineering specialization program. It encompasses the implementation of all aspects of Industry 4.0 needed to self-regulate a building's environment and operations. We as a group are responsible for the overview control of the smart building. While other groups developed the main systems (HVAC, Lighting, Safety, Energy usage, etc...), the control group worked on how they connect and collaborate to perform as a single unit. We built control logic flows for different scenarios where two or more systems connect and follow automated operations, implemented node-red flows for each of them, and set up MQTT and database facilities in a new server dedicated to this project.

Group Members:

- E/17/154
- E/17/072
- E/17/342
- E/17/380
- E/17/338

Table of Contents

Chapter 01: Introduction

Control Group

The control group was tasked with maintaining and organizing sub-system interoperations within the given system. However, the group had to undertake a bigger role in the completion of the whole project acting as the main point of contact for all design decisions. The project was divided into subsections where each group focuses on data generation, process control, and actuator controls of relevant systems. The Control group established itself in the midst of all these exchanges as a mediator.

Overall System Design

The system was divided into sub-systems and given to separate groups to work on the design and implementation aspects. Defined groups are as follows,

Group	System	Description
A	HVAC	Controls heating, ventilation and airflow based on temperature, humidity, and pressure
B	Lighting	Controls Lighting systems based on ambient light and occupancy
C	Safety	Controls safety components based on cautionary sensors
D	Energy	Controls power supplied to elements on the system for efficient power usage
E	Occupancy	Controls access for people and monitors population behavior

F	Photovoltaic	Monitors solar powered battery system and controls energy usage by the whole system
G	Control	Provides control over the underlying subsystems

Setting up Infrastructure

The initial setup of the infrastructure was based on docker containers running on the department VPN server at vpn.ce.pdn.ac.lk. However, the upkeep of the docker containers provided to be troublesome as it requires admin privileges for the server. As a result, we had to move our containers to a newer machine.

The machine is hosted on the local ip 10.40.18.10 which can be accessed by logging into the vpn. The actual infrastructure setup built is as follows,

<table>
<tr><td>Nodered : 50001</td><td>Nodered : 50001</td><td>Nodered : 50001</td><td>Nodered : 50001</td><td>Nodered : 50001</td><td>Nodered : 50001</td><td>Nodered : 50001</td><td>MQTT : 1883</td><td>MongoDB : 27017</td></tr>
<tr><td colspan="9">Docker Daemon</td></tr>
<tr><td colspan="9">Host OS (Ubuntu Server 22.04)</td></tr>
<tr><td colspan="9">Proxmox VM Manager</td></tr>
<tr><td colspan="9">Hardware</td></tr>
</table>

With the use of docker containers the process of running multiply instances of nodeRED , MQTT and MongoDB was handled successfully.

References

- https://docs.docker.com/
- https://pve.proxmox.com/pve-docs/
- https://mosquitto.org/documentation/
- https://nodered.org/docs/
- https://www.mongodb.com/docs/

Chapter 02: Conceptual Design

The conceptual design of the whole system was based on the Computer Engineering Department building , University of Peradeniya. For the implementation only one floor was considered while the design was done for three floors. During the design process we took into account the various systems and their components and the placements of these components thought the building.

Building

Legend

Symbol	**Component**
FE	Fire Extinguisher stations
FA	First Aid box
Ss	Safety Shower
PS	pull station
S	Sprinkler
SD	smoke-detector
FD	fire detection
FAL	fire alarm
IND	indicators
OC	Occupancy sensors
RFID	RFID card reader
AHU	Air Handling Unit
DPS	Differential pressure sensor
THS	Temperature Humidity sensor
L	Lights

I	Intensity
B	Boiler
C	Chiller

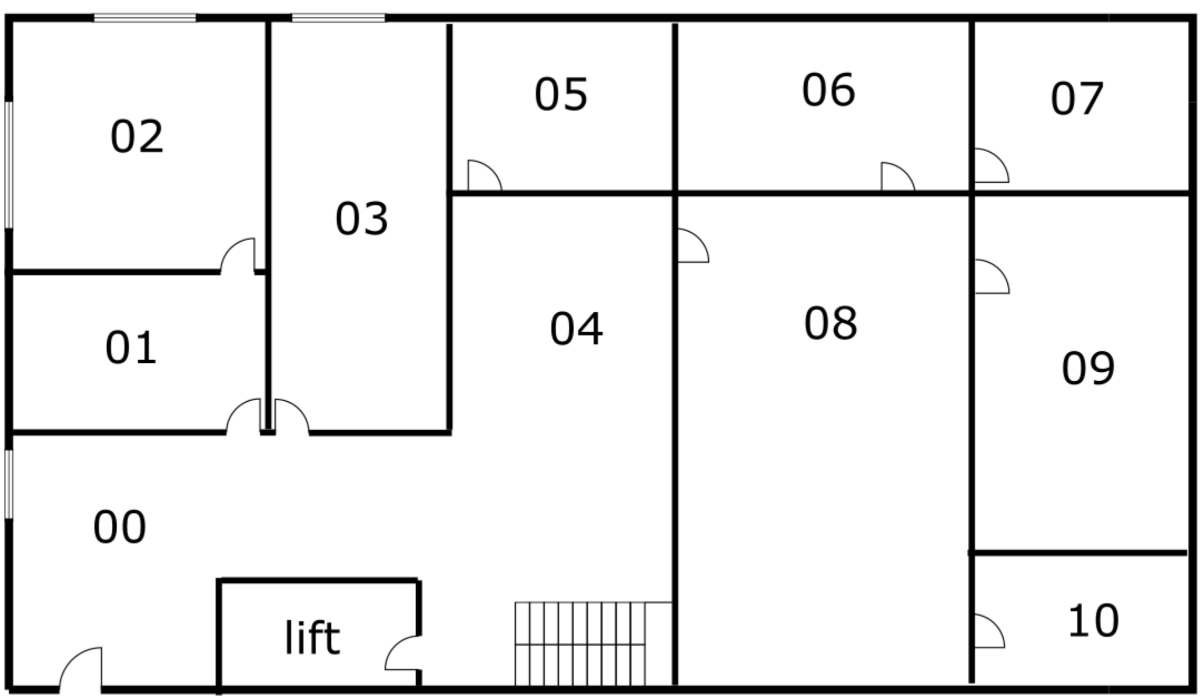
Floor0(GND)

Components

Room_no	Description	Emergency	Sensors	Actuators
00	Lobby, seating area		THS, I,OC,FD	IND,L,FAL
01	Department Office		FD, I , OC	IND, FAL,L
02	HoD's Office		AHU, DPS,THS	FAL,L
03	Conference Room		FD, SD, AHU, DPS,THS, I	S, FAL,L
04	Central Lobby	FE, PS	FD, SD, I	S,IND,L
05	Storage / Filing room	FE, PS	FD, SD, AHU, DPS,THS	L
06	Staff Room	FA	FD, SD, AHU, DPS,THS	S,L
07	Boiler/Chiller room	FE	FD,SD, THS	B,C
08 09	Open Lab / Space	FE, PS	FD,SD, AHU, DPS,THS, I	S, FAL,L
10	Washrooms			IND,L

Considerations

- In room no 05, a sprinter is not suitable because there might be many files then it would be even worse when these files are got wet. So, we have placed a fire extinguisher station and a pull station is placed in it.
- AHU placed at enclosed spaces such as rooms 04,08 along with DPS to measure
- Fire alarm placed in lobby along with fire indicators for rooms that might not hear the alarm. (washrooms)
- Pull stations located in open spaces like the lobby and closed places like storage room.
- Occupancy sensors to keep track of people present in rooms .Helps to maintain airflow and temperature as well as at emergencies.
- RFID card reader as security mechanism for specified rooms with restricted access.
- Boiler and chiller in room 07 will be used by the whole building.



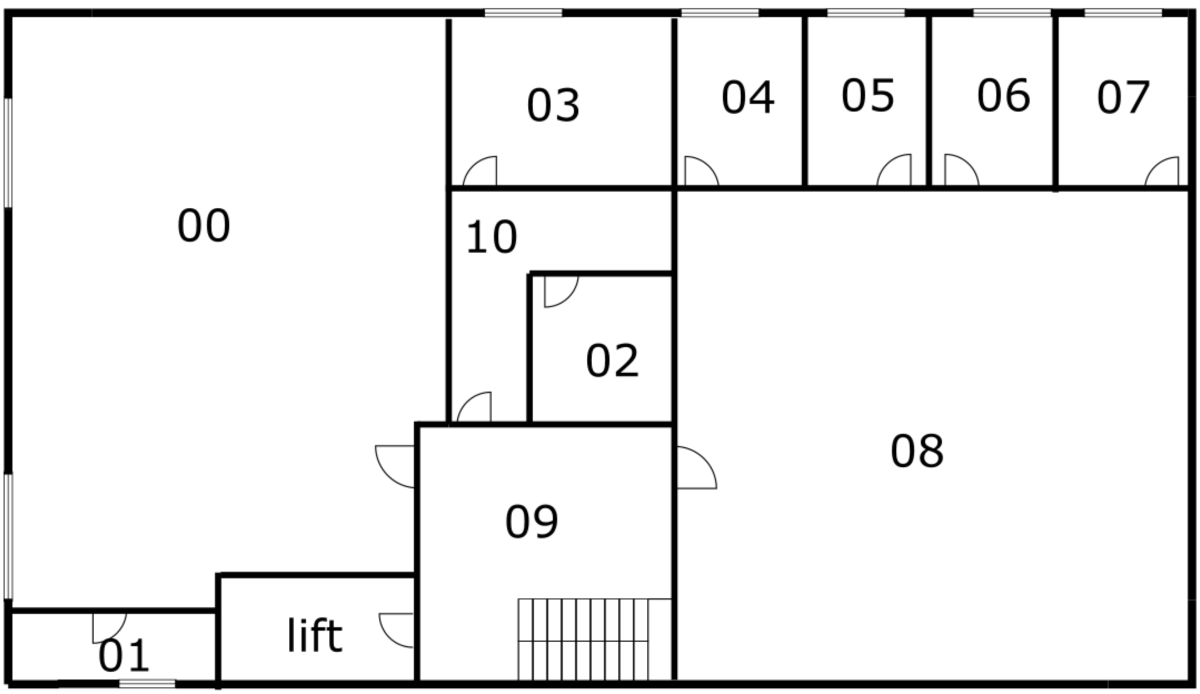
FLOOR_0 (Ground floor)

Floor1

Components

Room_no	Description	Emergency	Sensors	Actuators
00	Computer Lab with 64 student seating + 2 built-in projectors	FE, PS, FA	AHU, DPS,THS, SD, FD,OC	IND, FAL,L
01	Technical officer room		LS, THS,THS	IND, S, L
02	Instructors room (for 4)	FE, PS	LS, OC, AHU, DPS,THS	IND, S, L
03	Lecturers' Rooms		LS, THS,RFID	IND, S, FAL, L
04			LS, THS,RFID	IND, S, L
05			LS, THS,RFID	IND, S, L

06			LS, THS,RFID	IND, S, L
07			LS, THS,RFID	IND, S, L
08	Network Engineering Lab 32 students	FE, PS,FA	OC, AHU, DPS,THS, SD, FD	IND, FAL, L
09	Lobby	PS	LS, SD, FD	IND, FAL, S, L
10	Lift	FE	SD, FD,OC	B, C, S,L



FLOOR_1

Considerations

- Sprinklers are not placed in labs , fire extinguishers are placed instead.

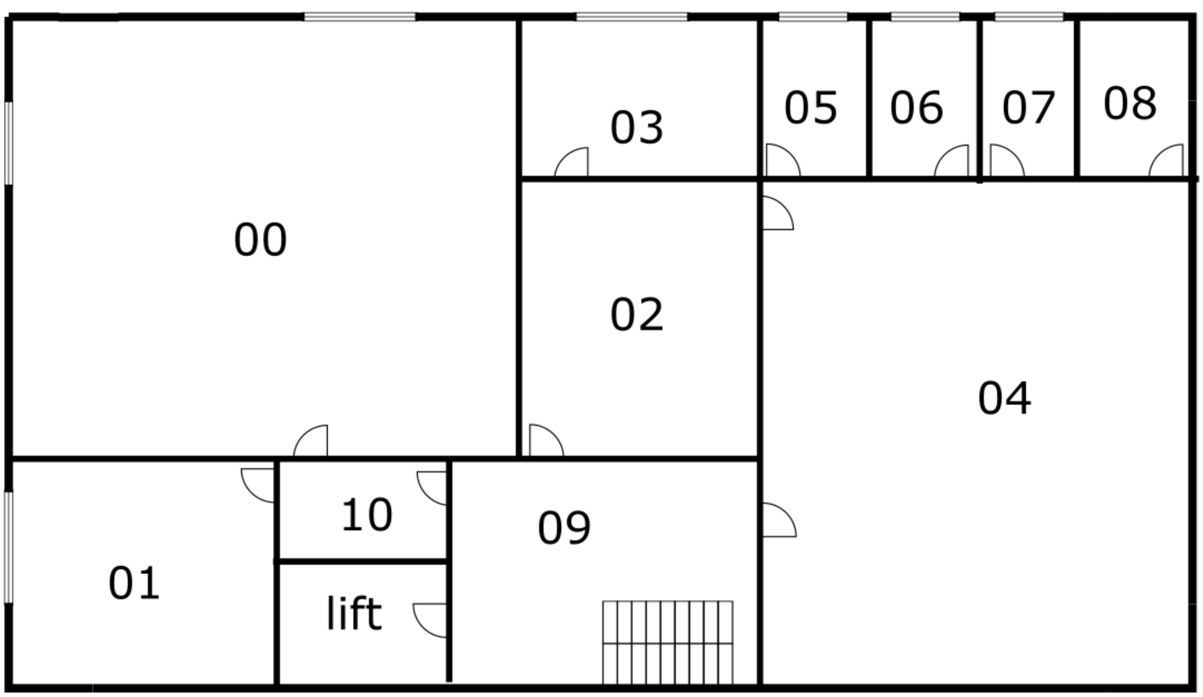
- Fire alarms are placed in all the labs and the lobby and in the lecturers' room 3 because it is separated from the others.
- Pull stations are placed in all the labs and the lobby.
- AHU placed at enclosed spaces such as rooms 00, 08 along with DPS to measure
- Lecturers' rooms and labs have RFID due to security concerns.
- Boiler and chiller placed in every room except the lobby area.
- Fire extinguishers and sprinklers are placed in the lift since it is a vulnerable area.
- Rooms, where many people can be present, are tracked using occupancy sensors.

Floor2

Components

Room_no	Description	Emergency	Sensors	Actuators
00	Electronics Laboratory • 30 computers • Oscilloscope • Power Supplies etc. Seating for 60 students	PS, FE	AHU, DPS, THS, OC, SD, FD,I, RFID	IND,L
01	Technical officer's room		RFID	IND,L
02	Instructors' Room (4 people)		AHU, DPS, THS	L
03	Industrial Automation lab		FD, SD, AHU, DPS, THS,RFID	S,L
04	Discussion room	FE, PS	FD, SD, AHU, DPS, THS,I,OC	S, IND,L

05	Lecturer's rooms	FA	OC, SD	L
06				
07				
08				
09	Lobby	FE,FA	I	FAL,S,L
10	Passage			L,IND



FLOOR_2

Considerations

- Boiler and chiller common to whole building
- Each temperature control there is modules, fans, sensors, joint temperature (AHU), blower and the air handling part
- Rooms with security concerns have RFID card readers to authorize entrants.(Technical officer’s room / Industrial Automation room/ Electronics Lab)
- Sprinklers are not used in lab rooms and Fire extinguishers are used instead.
- Rooms, where many people can be present, are tracked using occupancy sensors.
- Light intensity sensors are located in open areas to get a better reading of ambient light.

Chapter 03: MQTT

One of the tasks undertaken by the control group is to standardize and maintain the MQTT topics used by the systems. Which meant coming up with a suitable format for the topics that can be used by the groups. This was done by taking into account the conceptualized building that the system is going to be placed in which would cause the topic to include information such as system name, floor no, and room no.

All the topics pertaining to specific system starts with a similar set of sections as,

Common portions

HVAC	**326project/smartbuilding/hvac/**
Lighting	**326project/smartbuilding/lighting/**
Safety	**326project/smartbuilding/safety/**
Energy	**326project/smartbuilding/energy/**
Occupancy	**326project/smartbuilding/occupancy/**
PhotoVoltaic	**326project/smartbuilding/pv/**

After the beginning portion of the topics the next parts will be <floor_no> and <room_no> for sensors and actuators that can specified for one room. If the considered elements are building wide or floor wide exceptions have been made.
For ex:- Boiler/ Chiller control in HVAC system

Specific portions

HVAC

Boiler/chiller and main control/acquisitions	**control**	**326project/smartbuilding/hvac/control/boiler**
		326project/smartbuilding/hvac/control/chiller
	Sensing	**326project/smartbuilding/hvac/coldairduct/temperature**
		326project/smartbuilding/hvac/hotairduct/temperature
	set thresholds	**326project/smartbuilding/hvac/control/temp-thresh**
		326project/smartbuilding/hvac/control/temp-thresh-coldairduct
		326project/smartbuilding/hvac/control/temp-thresh-hotairduct
Ventilation control in rooms	**Sensing**	**326project/smartbuilding/hvac/<floorno>/<roomno>/temperature**
		326project/smartbuilding/hvac/<floorno>/<roomno>/humidity
		326project/smartbuilding/hvac/<floorno>/<roomno>/pressure

		326project/smartbuilding/occupancy/<floorno>/<roomno>/count
	Control	**326project/smartbuilding/hvac/<floorno>/<roomno>/control/set-temperature**
		326project/smartbuilding/hvac/<floorno>/<roomno>/control/ahu/blower
		326project/smartbuilding/hvac/<floorno>/<roomno>/control/ahu/airflowrate
	set thresholds	**326project/smartbuilding/hvac/control/temp-thresh**
		326project/smartbuilding/hvac/control/humid-thresh
		326project/smartbuilding/hvac/control/flowrate-thresh

Lighting

326project/smartbuilding/lighting/<floornoroomno>/lightsensor/inside/<sensorno>
326project/smartbuilding/lighting/<floorno>/<roomno>/lightsensor/outside/<sensorno>
326project/smartbuilding/lighting/<floorno>/<r>/<oomno>/switch/inside/<switchno>
326project/smartbuilding/lighting/<floorno>/<roomno>/switch/outside/<switchno>

Safety

326project/smartbuilding/safety/<floorno>/<roomno>/smoke
326project/smartbuilding/safety/<floorno>/<roomno>/fire

326project/smartbuilding/safety/<floorno>/<roomno>/sprinkler
326project/smartbuilding/safety/<floorno>/<roomno>/firealarm
326project/smartbuilding/safety/<floorno>/<roomno>/pullstation

Energy

326project/smartbuilding/energy/

Occupancy

326project/smartbuilding/occupancy/<floorno>/<roomno>/occupants
326project/smartbuilding/occupancy/<floorno>/<roomno>/rfid

PhotoVoltaic

326project/smartbuilding/pv/<panelno>/current
326project/smartbuilding/pv/<panelno>/voltage
326project/smartbuilding/pv/<panelno>/watt
326project/smartbuilding/pv/<panelno>/kwhmeter
326project/smartbuilding/pv/controls/sw1
326project/smartbuilding/pv/controls/sw2

Chapter 04: Database

Since a NoSQL database is being used in the system MongoDB collections were used to store all sensor data. The CRUD operations are provided by using the MongoDB node installed on the nodeRED instances. The availability of creating collections has been given to the groups under a specified naming convention as ,

System	**collection name**
HVAC	co326_hvac_<xxx>
Lighting	co326_lighting_<xxx>
Safety	co326_safety_<xxx>
Occupancy/Access Control	co326_occupancy_<xxx>
Energy	co326_energy_<xxx>
Photovoltaic	co326_pv_<xxx>

Where xxx defines sensor/actuator names.

The collections store information as,

Id_ = timestamp of data received

sensor/actuator state = values to define component behaviour

Topic = mqtt topic data is received by

During control operations location of components are found by the topics as they contain information regarding floor and room no.

Current Collections and stored data formats

System	Fields	TYPE	Mongo DB Collection Name	Remarks
HVAC				
themostat	temp_level	FLOAT(3,2)	326_hvac_sensor_temp	
humidity	humidity_level	FLOAT(3,2) // percentage	326_hvac_sensor_humid	
DPS	TBD	TBD		
SPS	TBD	TBD		
Boiler	active	BOOL	326_hvac_control_boiler	
Chiller	active	BOOL	326_hvac_control_chiller	
AHU	fan_speed	FLOAT(3,2) // percentage speeed	326_hvac_control_ahu	
Voltage	voltage	FLOAT(3,2)		
Current sensor	current	FLOAT(3,2)		
Lighting				
Intensity (LDR)	light_level	INT	co326_lighting_sensor_data	
Light on/off	active	BOOL	co326_lighting_status	
Voltage	voltage	FLOAT(3,2)		
Current sensor	current	FLOAT(3,2)		has additional collection, co326_lighting_control to store control

				decision on the database.
Safety				
Fire_detect	smoke_level	BOOL	326_fire_detect	
Smoke_detect	active	FLOAT	326_smoke_detect	
Sprinkler	active	BOOL	326_sprinkler	
Fire_alarm	active	BOOL	326_fire_Alarm	
pull	active	BOOL	326_pull_station	
Voltage	voltage	FLOAT(3,2)	326_voltage	
Current sensor	current	FLOAT(3,2)	326_current	
Energy				
	Sensor name(This can be AHU,Boiler or Chiller)	VARCHAR	co326_Energy_final	
	Floor Number	VARCHAR	co326_Energy_final_lighting	Addition to this attributes this collection has state attribute
	Room Number	VARCHAR		
	Time	TIME		
	Power	FLOAT(3,2)		
Occupancy / Access control				
rooms_status	count	count = INT ,floor_number = int, room_number = int,time = TIMESTAMP	326_occupancy_room	this collections stores the data of the relevant room. Room has a unique number and a uniques password
rfid_data	room_number,	room_number =int, rfid =	326_occupancy_rfid	

	rfid_value,floor_number	string,floor_number = int		
room_keypad_data	room_number,keypad_value,floor_number	room_number =int , keypad_value = string	326_occupancy_key_pad	
recent_history	object		326_occupancy_recernt_history	
Photovoltaic System				
Current sensor	voltage	FLOAT	co326_pv_current Sensor	
Voltage sensor	current	FLOAT	co326_pv_voltageSensor	
Electric kWh meter	electric_meter_val	INT	co326_pv_kWhmete	
Watt meter	watt_meter_val	FLOAT	co326_pv_Wattmeter	
Battery Controller	battery_state	INT	co326_pv_btReady	

Chapter 05: Control Flows

HVAC

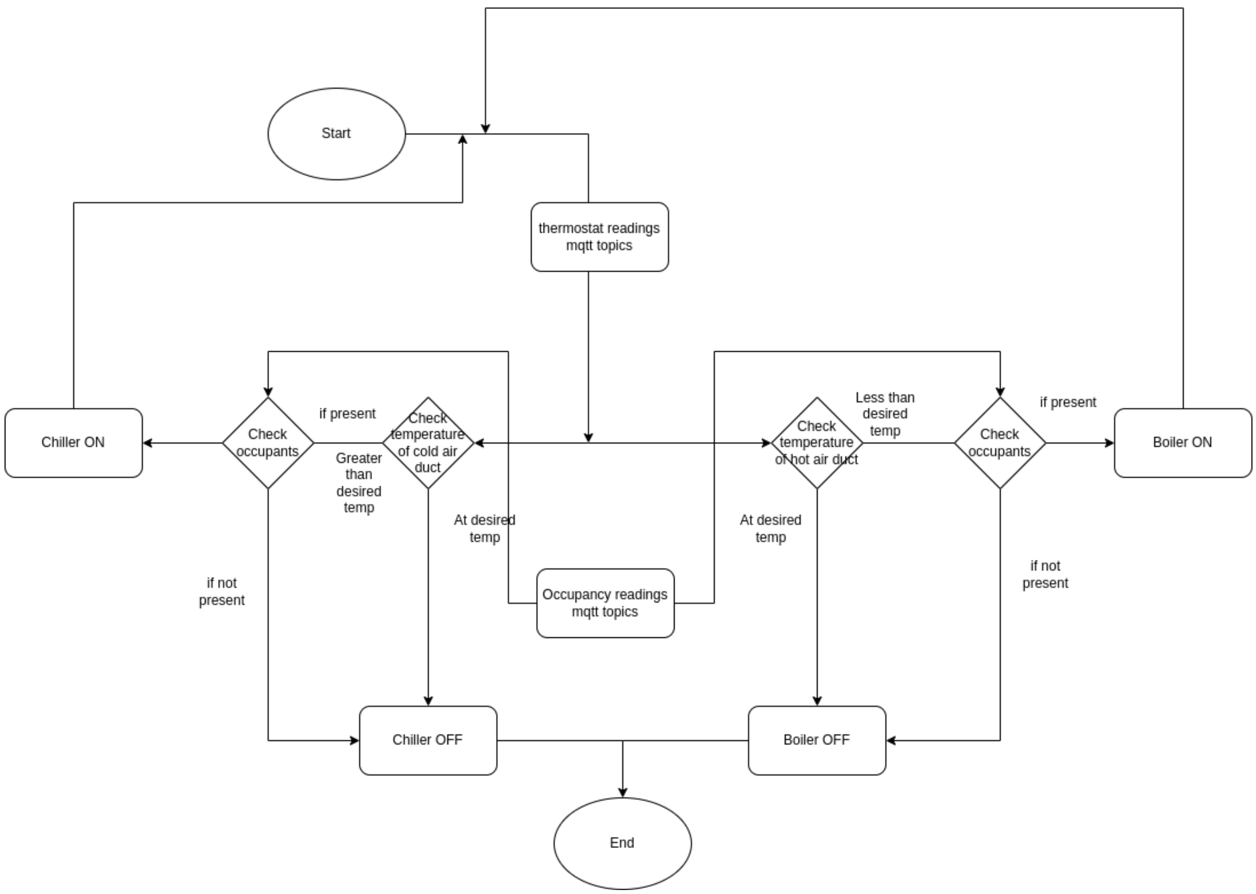
Comfortable temprature

Since the ambient temprature varies in each places, the comfortable temperature varies for each places. Therefore, we took the average temperature as the comfortable temperature for that building.

Occupancy based Boiler/Chiller Control

HVAC has some control units; those units will work according to some independent parameters such as temperature, humidity, and air. Those parameters are handled well in HVAC controls. Occupancy controls detect and publish the occupants' status. But to save power, these handling units can be handled incorporated with occupants in a room.

Thus, the diagram below explains how the HVAC boiler and chiller units are handled depending on the occupancy status.



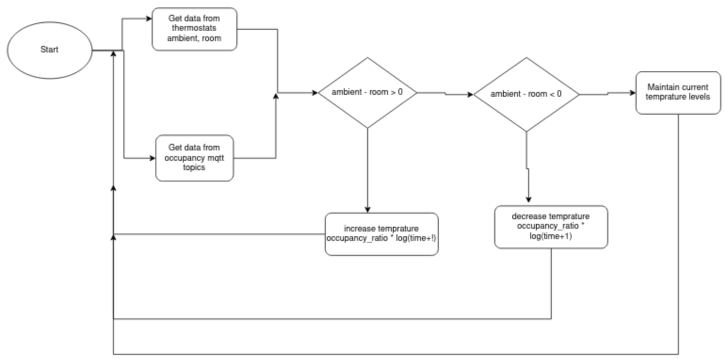
The points below explain the considerations when the above control flow was created.

Considerations

- Extension of HVAC Boiler/Chiller control based on occupancy sensors.
- Based on the temperature sensor MQTT topic.
- Boiler and chiller control work as ON or OFF; hence temperature value is thresholded for control.
- The boiler and Chiller are both turned off in nonoccupant spaces.

Occupancy-based AHU

On the other hand, the Air handling unit has been controlled to maintain a temperature with a smaller variance. But power could be wasted when depends For example, when occupancy is law, there is no need for higher rate control in the ahu unit. Therefore, the Air handling unit must also be handled by the occupancy status, to save power.

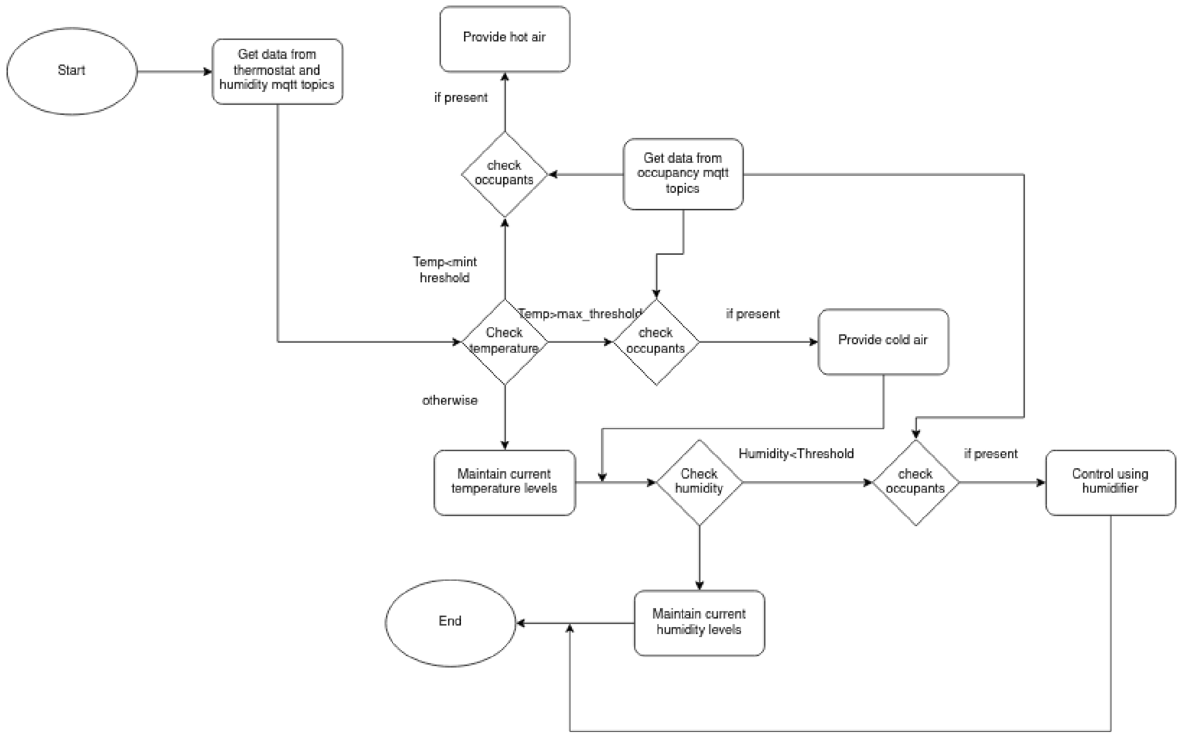


Considerations

- Control of air handling unit based on the differential pressure sensor and occupancy data through MQTT.
- On checking the current air flow rate controller decides to taking in air, remove air or maintain airflow.
- Considering occupancy sensors AHU can be moved to an IDLE state where it turns off but periodically checks sensor values.

Occupancy-based Humdity

When the room's temperature is really high there is a chance humidity could go low. This dryness can cause dry sinuses, bloody noses and cracked lips. humidifiers can help soothe these familiar problems caused by dry indoor air. This is ideally denfiend by HAVC group. But to save power. This has to be handled incorporate with the occupancy status.



The points below explain the considerations when the above control flow was created.

Considerations

- Extension of HVAC temperature/humidity (evaporative humidifier) control logic based on occupancy data.
- Thermostat and humidity taken from MQTT topics which are published by the sensors.
- Minimum and maximum threshold values are used for operation. This minimum and maximum temperature value is derived from the average ambient temperature value.
- Change temperature/humidity according to deviations or else maintain levels.
- Temperature controlled using the Boiler and Chiller signals are sent to the respective control signal MQTT topics.

- Humidity is maintained using the humidifier.
- Threshold values for the sensors are decided upon the user input from the *SCADA* interface.
- Acts as above with the added consideration of occupancy where an empty room temperature and humidity will not be maintained as rigorously as when the space is occupied.
- On default operation humidifier can remain idle for an hour and sample air for a few minutes and act accordingly.

Ambient Temperature modeling

$$y = low_{temp} + \left(\left(\frac{(High_{temp} - Low_{temp})}{High_{temp}} \cdot x \right) \cdot \exp\left(- \frac{x}{100} \right) \right)$$

Figure : equation

- Usually in srilanka the temperature varies between values around 28 - 29°
- To model the ambient temperature the above exponential function (depicted in (Figure : equation))is used so that the temperature can be vary between two temperatures showed in the figure.

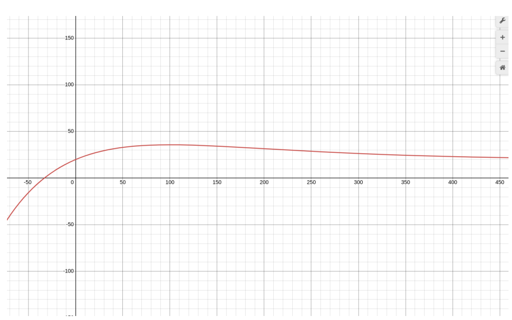


Figure : temperature variation when High_temp = 35 and Low_temp 25

- This function is used to model and simulate the ambient temperature.

```
import paho.mqtt.client as mqtt
import json
from time import asctime, time, sleep
from math import exp
from sys import argv
def temprature_model(temp_low, temp_high, time):
   return temp_low + ((temp_high - temp_low) / (temp_high)) * time * exp(
       -1 * round(time / 100, 2)
   )
# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
   print(msg.topic + " " + str(msg.payload))
# Initialize start time
start_time = time()
floorno = argv[1]
roomno = argv[2]
# MQTT info
broker_addr = "10.40.18.10"
broker_port = 1883
ambient_topic = "326project/smartbuilding/hvac/ambient_temperature"
# Create MQTT client instance and connect to broker
client = mqtt.Client(f"Floor{floorno}Room{roomno}Temp")
client.connect(broker_addr, broker_port)
print("Connected to broker")
client.on_message = on_message
# Subscribe to relevant topics
client.subscribe(ambient_topic)
print(f"Subscribed to {ambient_topic}")
count = 0
temp = 0
# Publish sensor readings
while True:
   elapsed_time = time() - start_time
   # Update temp value
   temp += temprature_model(25, 28, elapsed_time)
   temp /= count
   count += 1
   # Publish to MQTT topic
   data = json.dumps({"time": asctime(), "amb_temprature": round(temp, 2)})
   print(client.publish(ambient_topic, data))
   print(round(temp, 2))
   sleep(1)

   if elapsed_time > 180:
       count = 0
       temp = 0
       start_time = time()


# Never runs but added for safety
client.loop_stop()
```

Implementation

```
import paho.mqtt.client as mqtt
import json
from time import asctime, time, sleep
from math import exp
from sys import argv

# ambient temprature Model
def temprature_model(temp_low, temp_high, time):
   return temp_low + ((temp_high - temp_low) / (temp_high)) * time * exp(
       -1 * round(time / 100, 2)
   )

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
   print(msg.topic + " " + str(msg.payload))

# Initialize start time
start_time = time()

floorno = argv[1]
roomno = argv[2]

# MQTT info
broker_addr = "10.40.18.10"
broker_port = 1883
ambient_topic = "326project/smartbuilding/hvac/ambient_temperature"

# Create MQTT client instance and connect to broker
client = mqtt.Client(f"Floor{floorno}Room{roomno}Temp")
client.connect(broker_addr, broker_port)
print("Connected to broker")

client.on_message = on_message

# Subscribe to relevant topics
client.subscribe(ambient_topic)
```

```
print(f"Subscribed to {ambient_topic}")

count = 1
temp = 0

# Publish sensor readings
while True:
    elapsed_time = time() - start_time
    # Update temp value
    temp = temprature_model(25, 28, elapsed_time)
    # temp = temp / elapsed_time

    # Publish to MQTT topic
    data = json.dumps({"time": asctime(), "amb_temprature": round(temp, 2)})
    client.publish(ambient_topic, data)
    print(round(temp, 2))
    sleep(1)

    if elapsed_time > 180:
        count = 1
        temp = 0
        start_time = time()

# Never runs but added for safety
client.loop_stop()
```

Figure python script for controlling ahu temperature based on occupancy

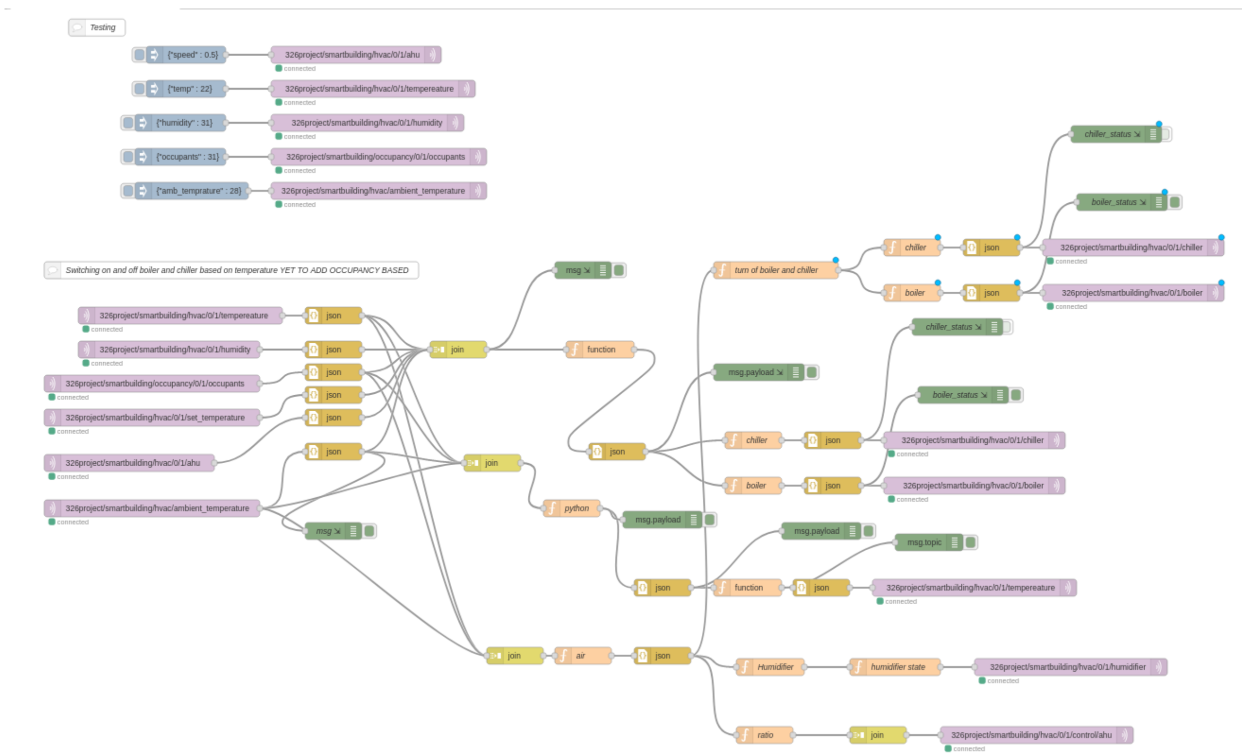
- 

Figure : node-red implementation

The system is defined using node-red. The upper portion of the diagram is used for testing purposes. These diagram is consists of all three systems mentioned above. From the most left part the MQTT topics which are published from HVAC and Occupancy are read. Then those topics are converted into json objects and passed to control functions. Those control functions are deciding some parameters based on occupancy status. Then it is published to the topics to control the machines.

References

- https://energyeducation.ca/encyclopedia/Home_temperature_control
- https://www.researchgate.net/publication/328944509_A_University_Building_Test_Case_for_Occupancy-Based_Building_Automation

Fire Control

Flow Diagram

In case of a fire it is very important that all the systems in the smart building work in sink and make sure that the necessary precautions are taken to quickly evacuate the building and make sure the fire is put out with minimum property damage. So in the below diagram shows how the control system acts in a fire, possible fire or a false alarm.

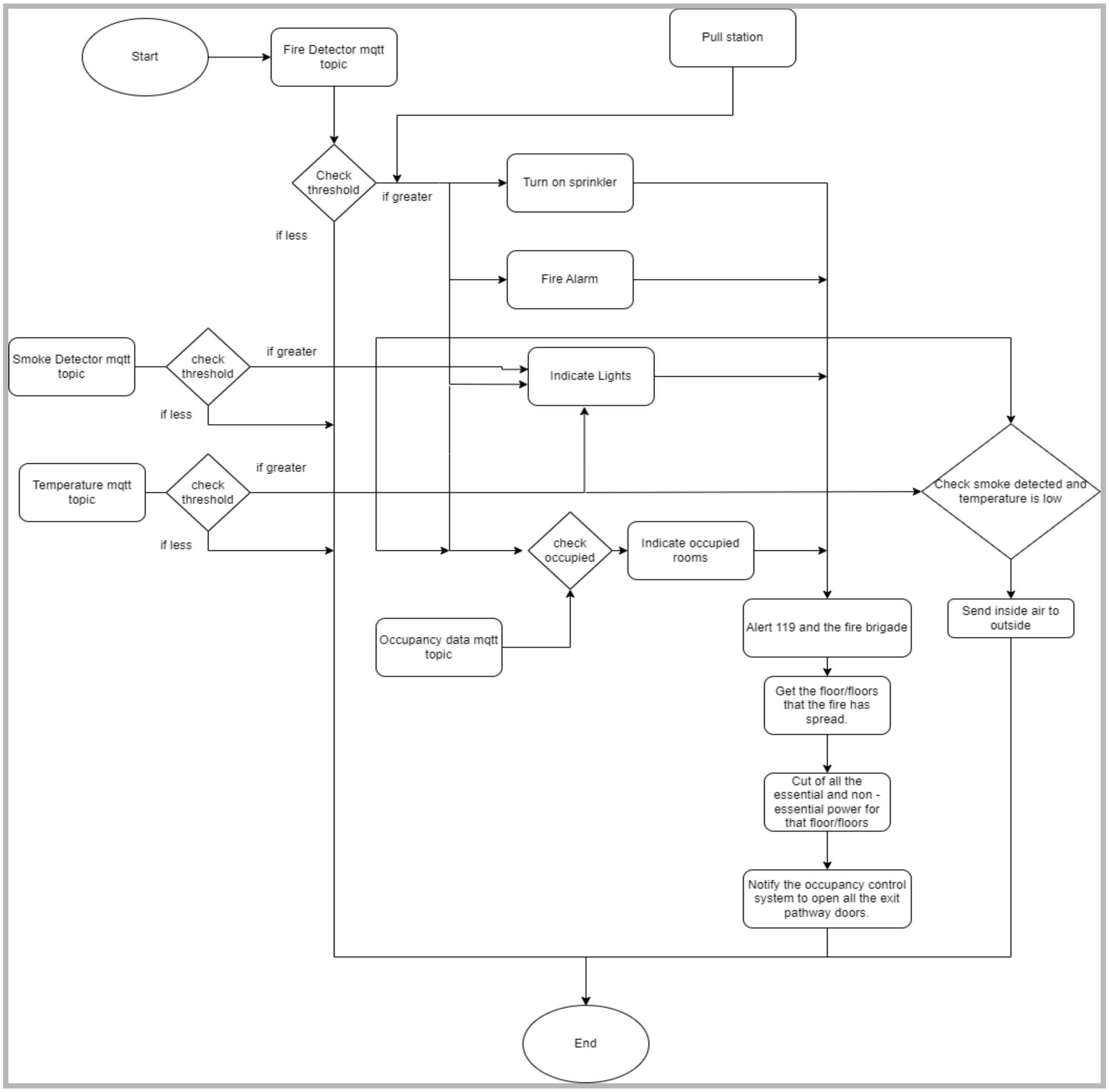


Figure: Control flow for the fire control

Considerations

Here are some of the considerations that were considered when creating the control flow.

- If a temperature of a room exceeds a particular threshold value there is a possibility that having a fire in that room. So the indicator lights are being turned on in that instance.
- If only smoke is detected and no high temperature or pull station triggers, it might not be a fire. So the HVAC control system is notified to put the smoke out of that room.

- On a fire detection indication from a fire sensor or a pull station trigger the sprinklers, fire alarms, and indicators will go off.
- Occupant rooms are checked, and they are being notified in case of a fire.
- If there is a fire required
- authorities will be alerted.
- Get the floor/ floors that the fire has already spread and cut off all essential and non-essential power for that floor. The essential and nonessential power will be cut from the floors where the fire has already spread.
- Open all the exit pathway doors.

Then the above control logic was implemented using node-red as you can see in the below diagram. Required sensor data and other payload data are acquired by subscribing to the relevant topics. Then some data like occupant count is get by querying the MongoDB database.

Implementation

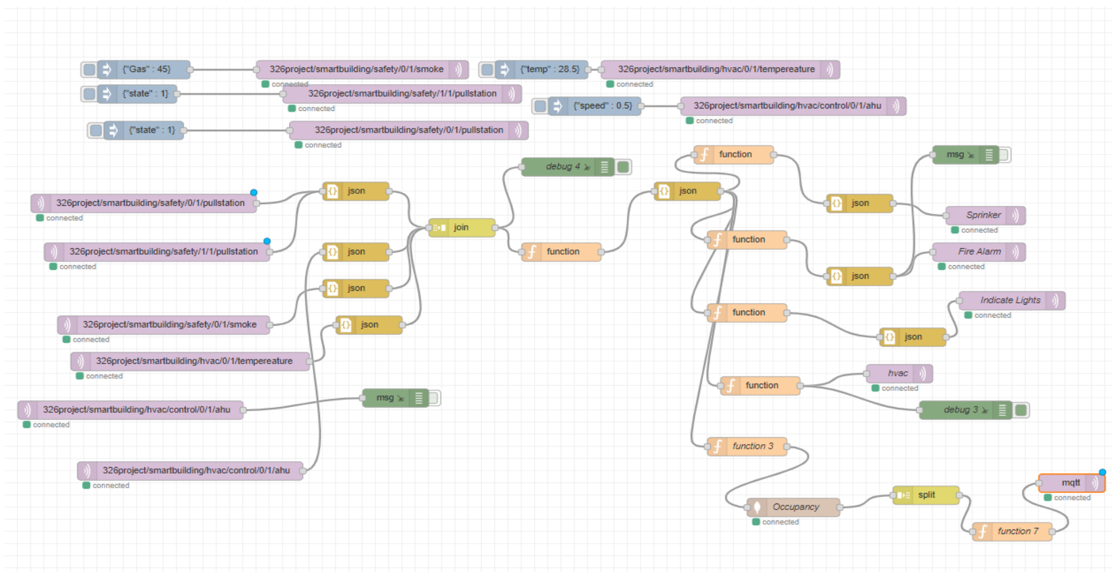


Figure: Node-red implementation for fire control logic.

The upper portion of the above diagram is used to inject some object data into MQTT topics in order to do the necessary testing. Then in the first part of the diagram is used to collect required sensor data from the MQTT topics and using e “join” node those singular objects are merged into bigger objects. Then using the big object some control decisions are made. Then they are sent to separate functions to take control decisions specific to each sensor, and those control decisions are published to the relevant MQTT topics.

Since we are giving overriding controls, only the control signals that need to be overridden are published to the topics.

References

- https://encyclopedia.pub/entry/19082

Energy Control

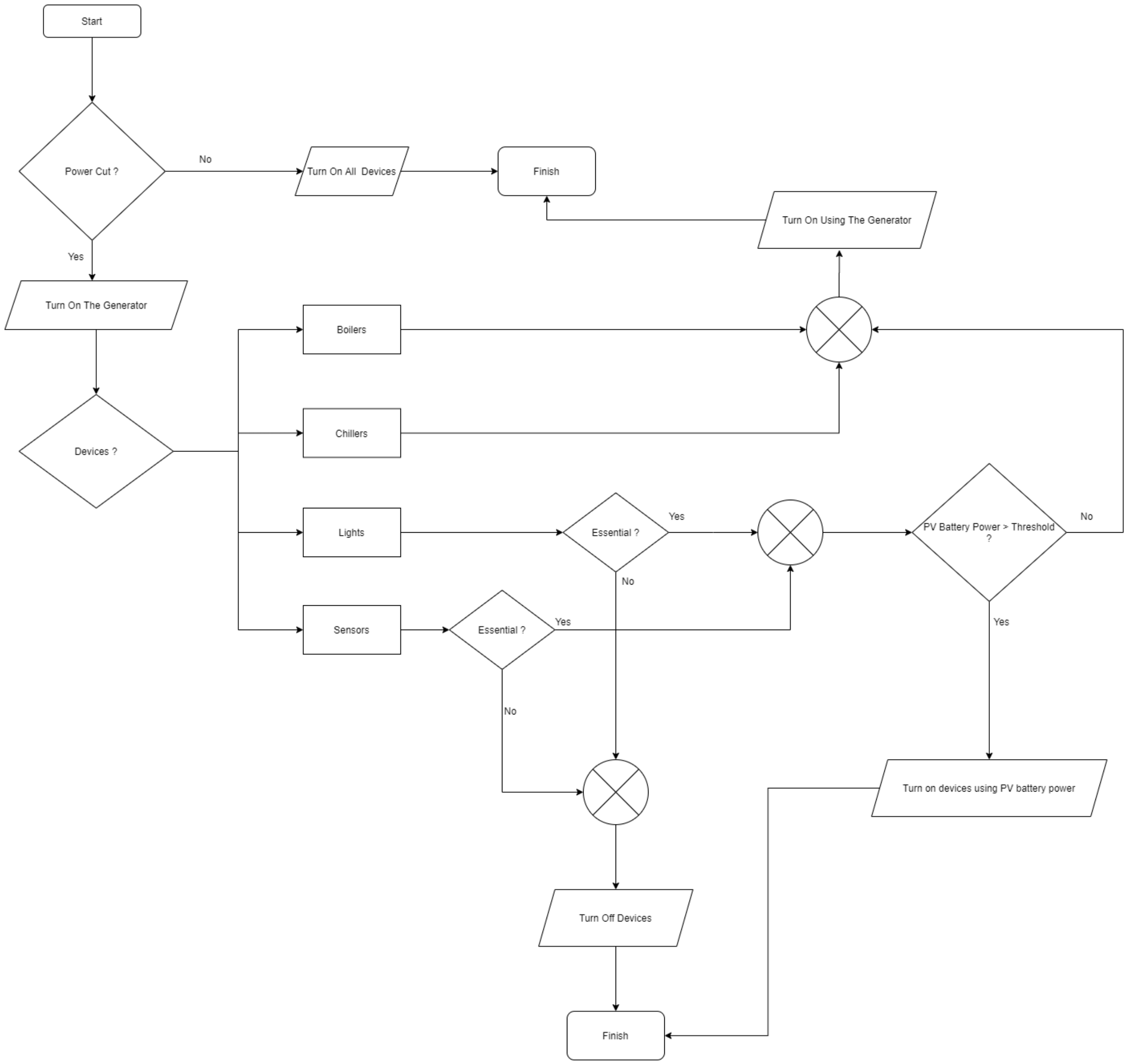
Keeping track of how components are powered is important when it comes to ensuring efficient usage of energy within the smart building implementation. This section is a compilation of two subsystems defined above as Energy and Photovoltaic groups.

The Energy group is focused on the distribution of power within the building. They maintain two circuits pertaining to essential and nonessential power. The switch of non-essential power depends on the power surges that happen in the circuit. And also controls power supply based on occupancy data.

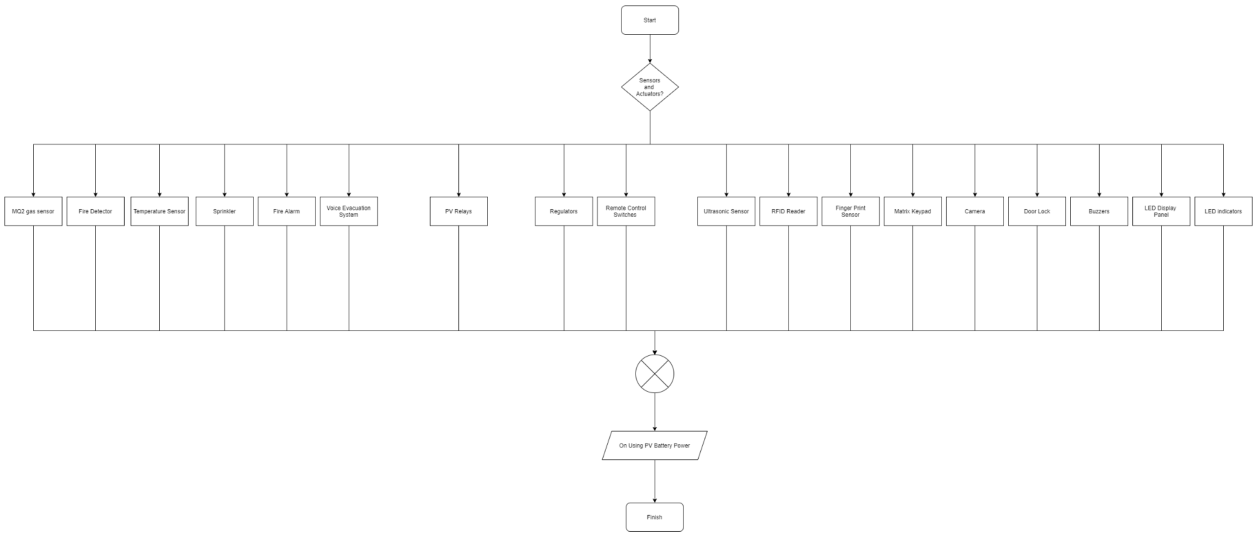
PV group focuses on using solar power to generate electricity and store it in a battery. The group uses this power for essential elements within the grid.

However, the essential elements themselves cannot all be powered by the solar power battery alone. Hence we define a control flow for the power supply of the components within the building.

Flow 1 - Control logic for using power



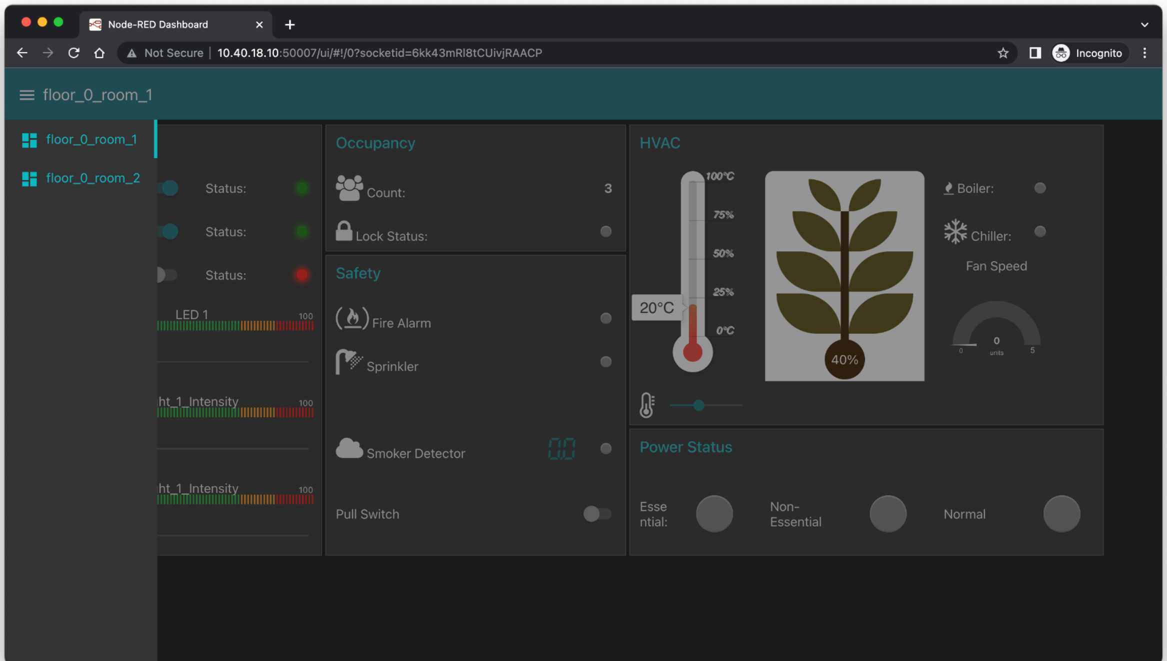
Flow 2 - Components and Power



References

- Photovoltaic European Technology & Innovation Platform "Assessing the need for better forecasting and observability of PV", 2017.
- S. Vergura, "Big data and efficiency of PV plants", 20th IMEKO TC4 International Symposium, Benevento, Italy, September 15-17, 2014.
- E.Koubli, D.Palmer, T.T. Betts R. Gottschalg. "Assessment of PV system performance with incomplete monitoring data", 31st EU-PVSEC, Hamburg, pp.1594-1597, 2015.
- https://www.leonics.com/support/article2_14j/articles2_14j_en.php#:~:text=MPPT%20checks%20output%20of%20PV,connected%20directly%20to%20the%20battery.
- https://www.ti.com/lit/ug/tiduej8a/tiduej8a.pdf?ts=1665980392608&ref_url=https%253A%252F%252Fwww.google.com%252F
- Variation of Voc against SoC of a lead acid battery

Dashboard



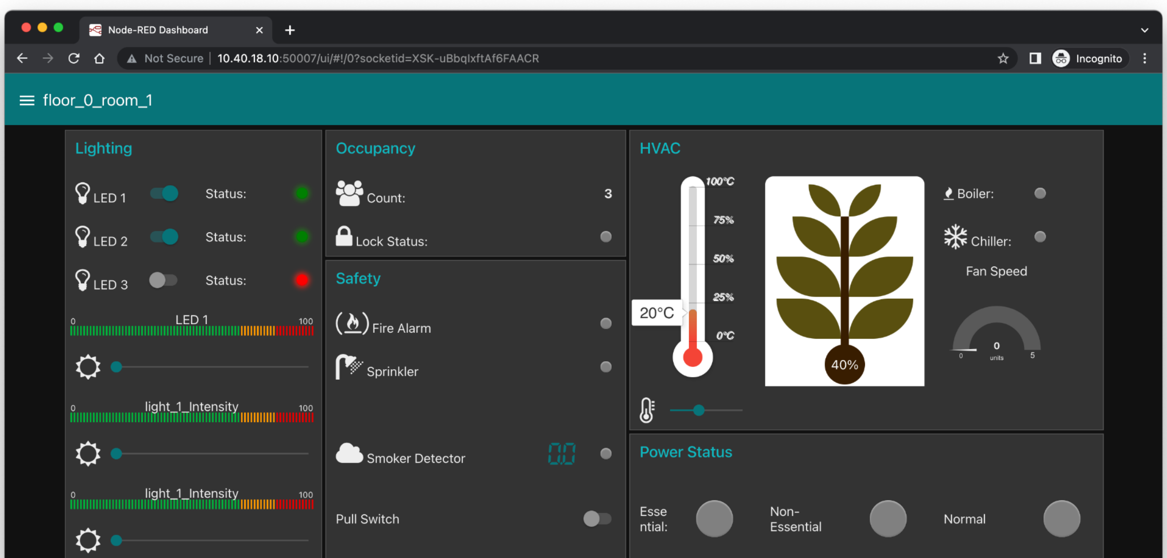


Figure: Occupancy Component

The dashboard provides a system to monitor the overall control systems and provides data insights about the condition of the entire operation of the whole system. The above figure demonstrates an overview of a single room on floor 0. It is a web-based dashboard we developed using node-red. It allows us to remotely view and control critical incidents that could occur inside a single room. Following is a brief description of each group currently available in a single tab for a room.

Occupancy

The occupancy portion of the dashboard is the smallest control view available. It basically gets the number of occupants of a single room from the database dedicated for the smart building project through a mongodb query. The occupancy group has developed a system where the sensors publish to a specific mqtt topic and saves that value simultaneously to the database that we set up. That value is extracted to be shown as the number of people who are present in the room at a given moment.

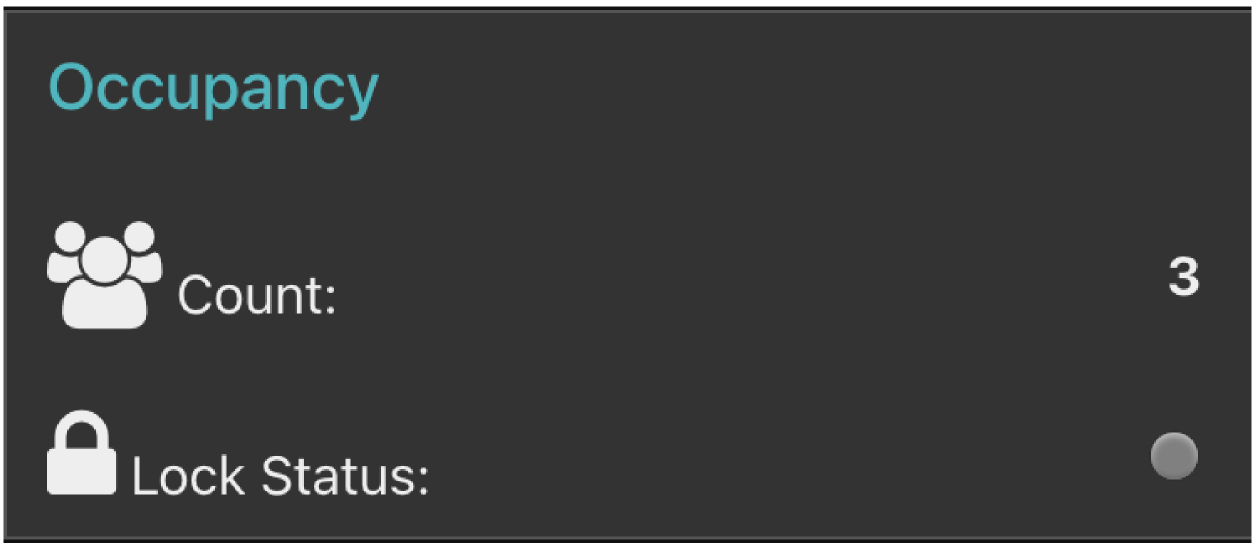
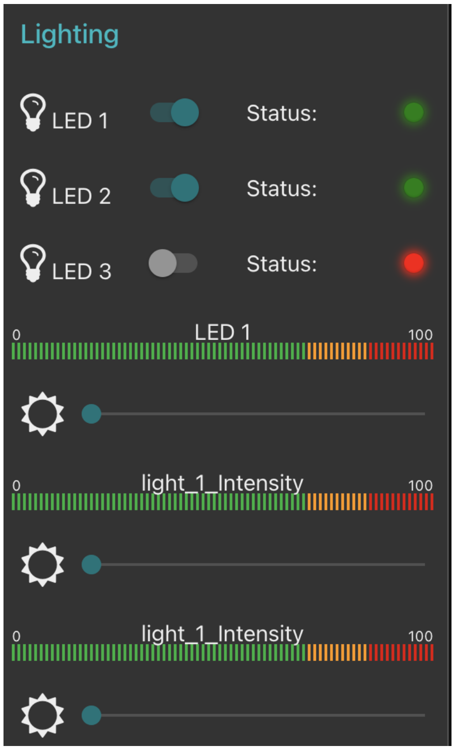


Figure: Occupancy Component

Lighting

This part is merely used to control the lights in a single room and monitor them. The control switches can be used turn on and off the lights through the dashboard. Switching the controls in dashboard will publish to the mqtt topic *326project/smartbuilding/lighting/<floor_number>/<room_number>/<light_number>* which is subscribed by the controller configured by the lighting group. The light intensity is also controlled in a similar manner.



The given figure shows the light control of room 1 in ground floor. The behaviour can be controlled using the given controls in the dashboard, or they will show what's happening in the room according to the flows that are configured by the lighting or any other group who has developed flows regarding to lighting.

Figure: Lighting Component

HVAC

Heating, ventilation, and air conditioning is another main system implemented in the smart building per each room. The temperature and humidity of a single room can be monitored using a dedicated component for HVAC. The behvaiour of the thermostat according to the temperature and humidity pairs are to be done by the HVAC team. Since we are the overall control the given dashboard component is mostly responsible for monitoring the values. It has the ability to change the temperature of the room through the relevant mqtt topic which will control the air conditioning or the heater. The humidity change can be observed with the temperature and other factors that affect it. According to the relevant scenarios the boiler or the chiller is turned on. Their statuses can be observed here as well. The fan speed of the blower found in the AHU controls the air flow rate supplied to a room. The fan speed indicates the percentage value.

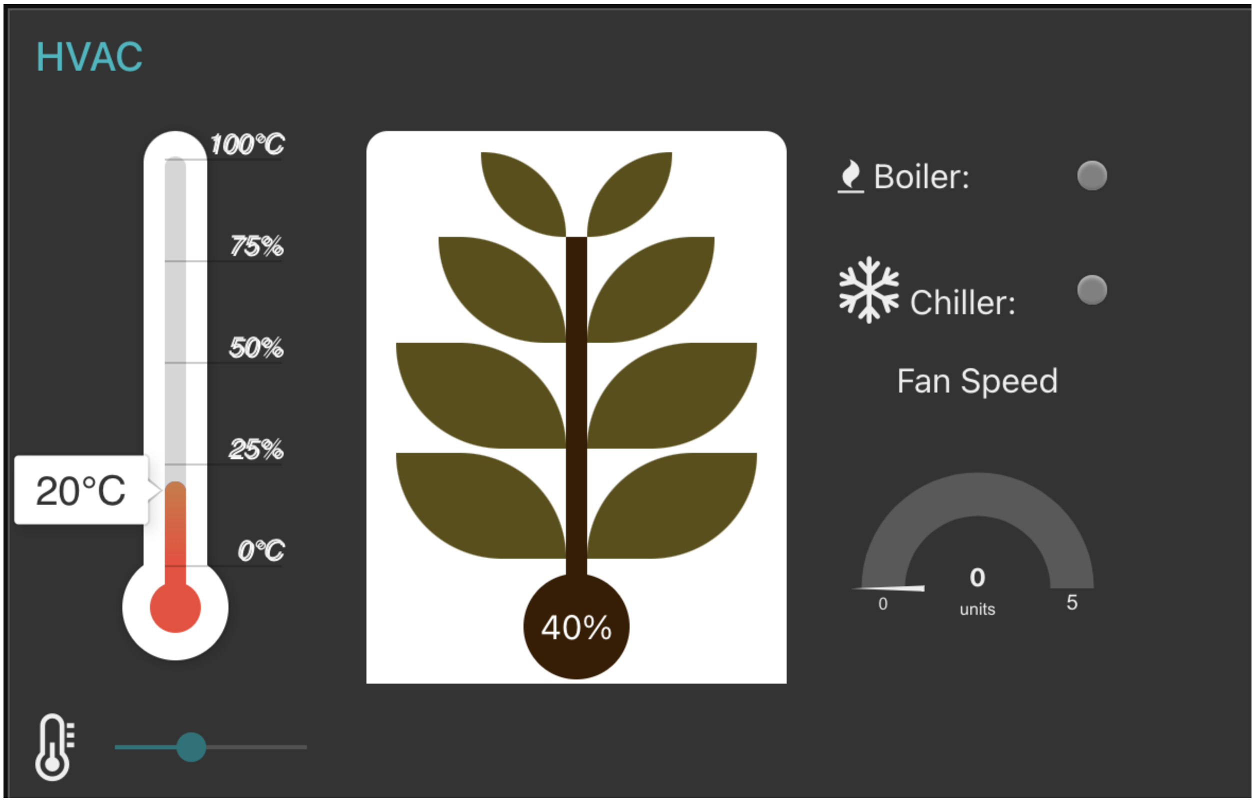


Figure: HVAC Component

Safety

The safety component of the dashboard is mainly focused on monitoring the status of the room respective to disasters and malfunctions that would cause damage to occupants and other valuables. When the fire alarm turns on due to a fire inside the room, the indicator dedicated for it lights up and an alarm goes on. Once the sprinklers are turned on that respective indicator will light up as well. If the sprinklers does not turn on after the fire alarm starts there's a malfunction with the hardware, and quick and necessary actions have to be taken instantly. The smoke detector shows the smoke density of the room at that moment. If that value goes over a threshold value defined by the safety team, the indicator will light up and necessary actions will be taken to avoid any harm caused to humans and other valuables. A manual control for the pull switch is provided in the SCADA interface so that the control system can be activated.

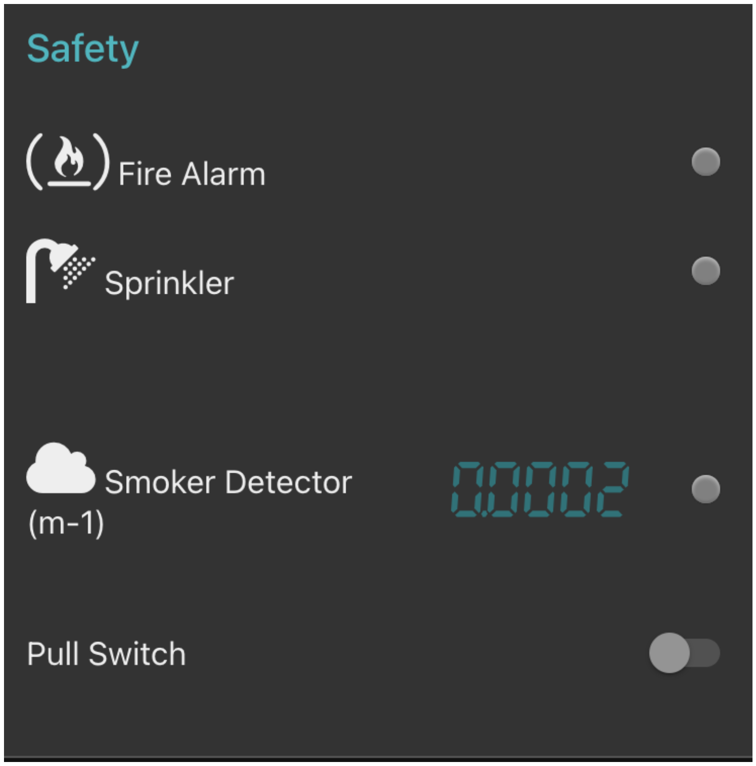


Figure: Safety Component

Power Status

This component of the dashboard is still wokr in progress. The energy team has to figure out mqtt topics and mongo db collections on how to monito essential and non essential component behaviours and what components are given power within normal conditions. After proper configuration, if power is given to only essential components the indicator will light up, if it's just also for non essentials as well that indicator will light up as well. If the behaviour of the components are normal the dedicated indicator will turn green, if not it will turn red.

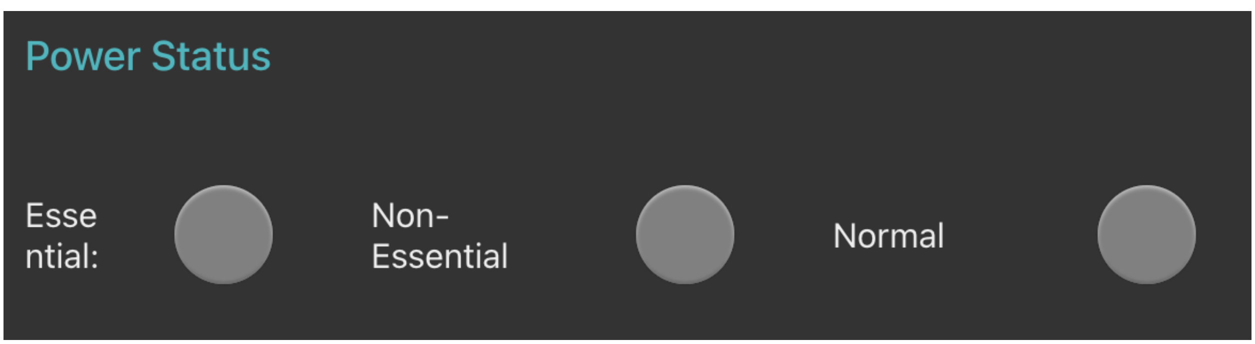


Figure: Power Status Component

Chapter 06: Data Analysis

Data analysis of the temperature and the relative humidity of rooms

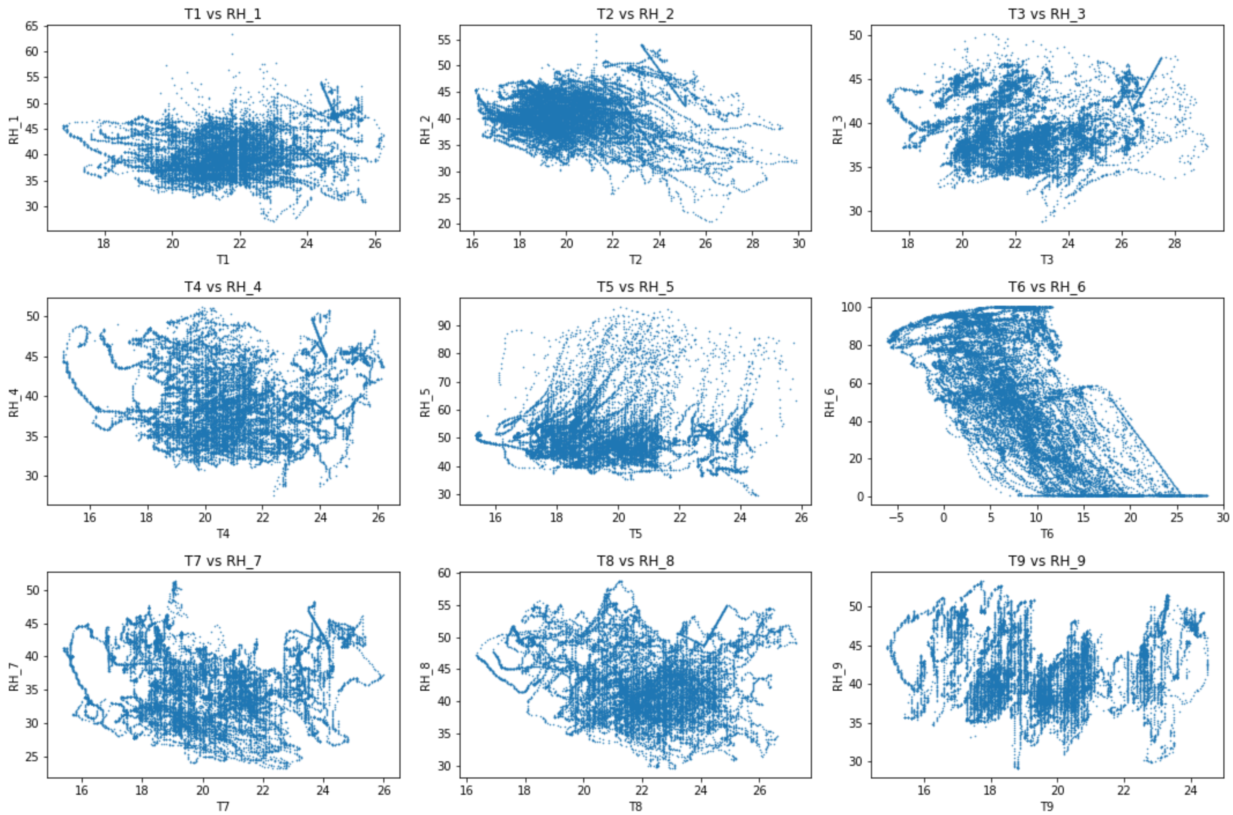
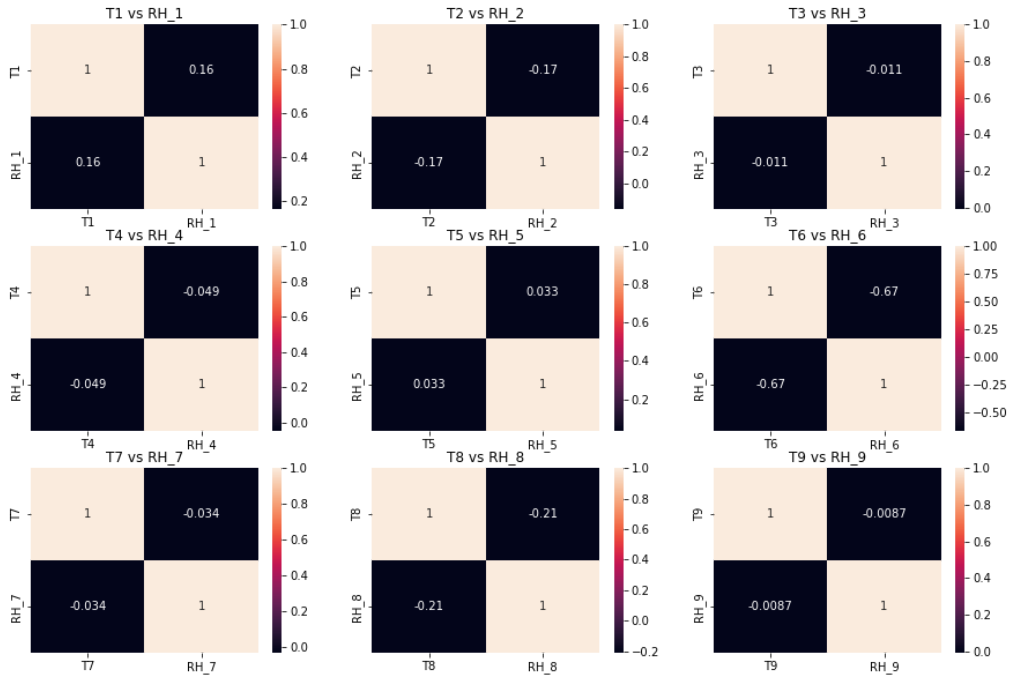


Figure: Temperature vs relative humidity



<u>FIgure: Correlation between temperature and humidity</u>

As you can see in the above diagrams there is very little correlation between temperature and the relative humidity. So we checked the maximum temperature and the corresponding humidity and found out that the relative humidities upper limit gets higher when the temperature increases.

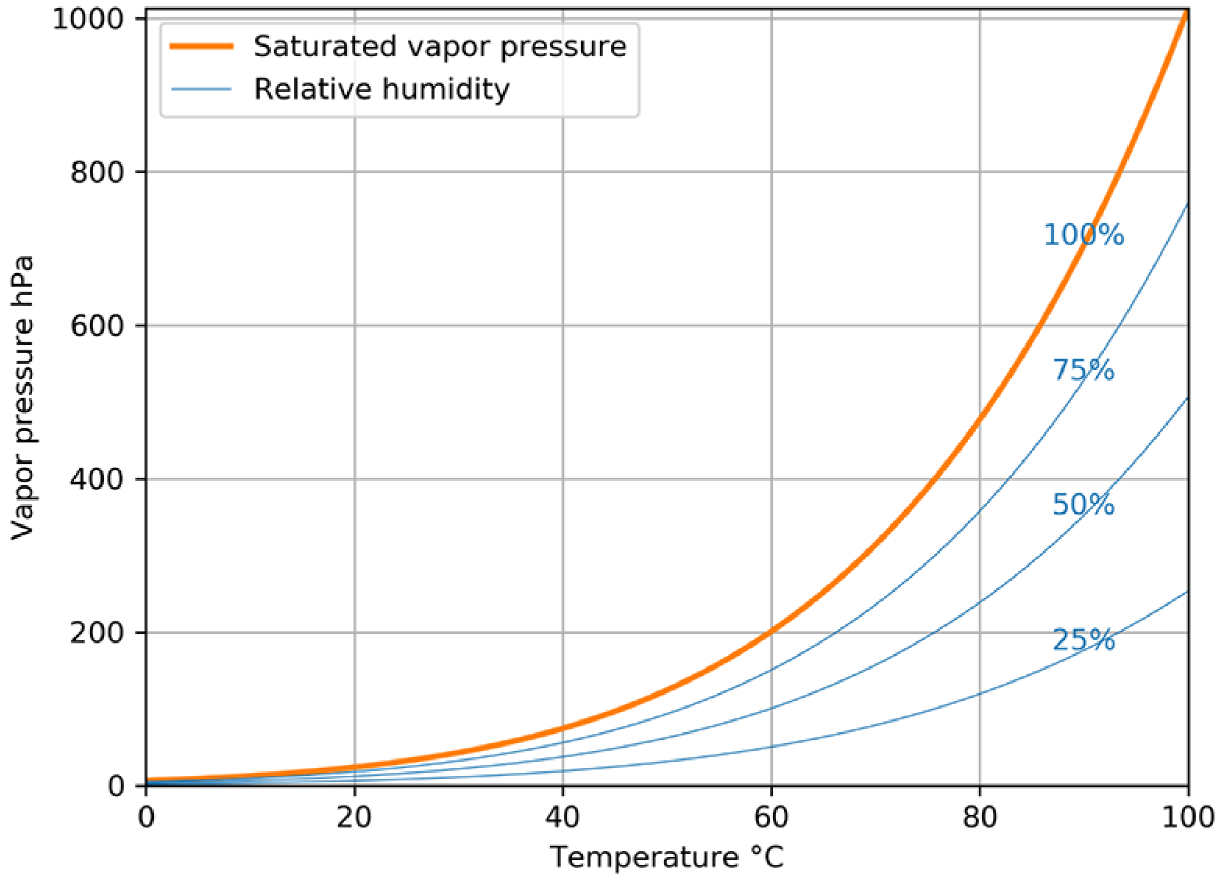


Figure: Temperature vs relative humidity

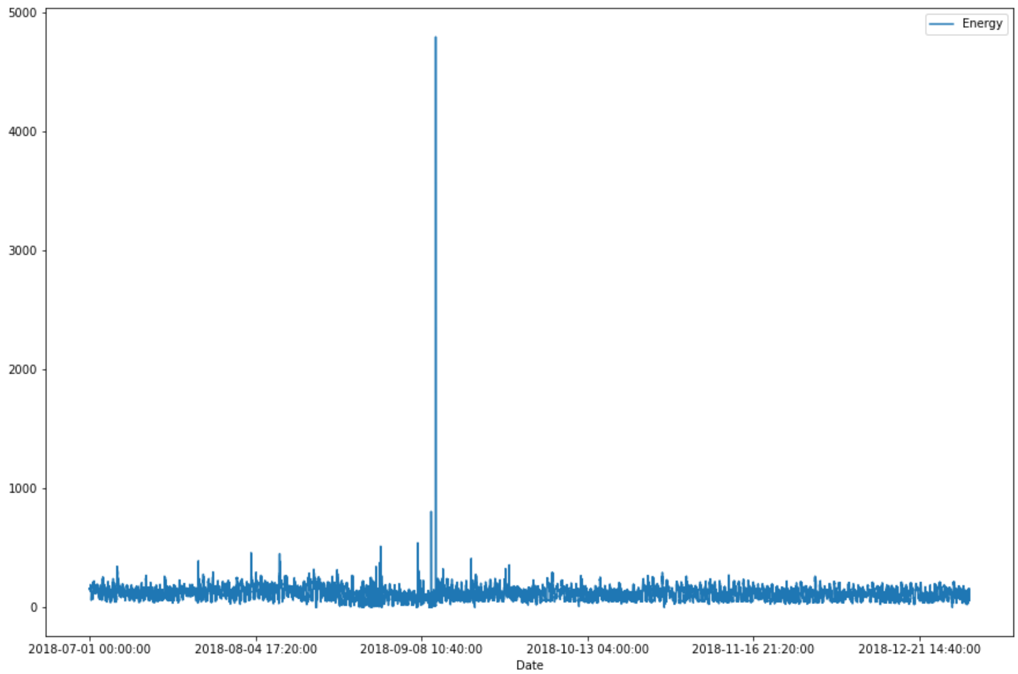


Figure: Energy consumption over time

Chapter 07: Conclusion

In conclusion the whole project in itself has been a continuous learning experience, with many pros and cons. On the positive side, we learned a lot. Not only in industrial systems but also in the field of building server infrastructure, and most of all team collaboration. However, it also comes with its downsides since a large team has been working on a single project it has proven to be harder to keep everyone on the same page. Even though there were defined formats for data, MQTT topics , and other naming conventions it has proven to be harder to enforce everyone to use them. In actuality, the cons have proven to be pros as well since it has given us a better understanding of working with large teams and the fallacies in doing so.