Journal of Engineering Science and Technology
Special Issue on SIET2022, May (2022) 242 - 265

SMART ENERGY BUILDING MANAGEMENT SYSTEM

YAP P. JET[1], LEONG Y.H. GIDEON[1], CHEW Z. YUE[1],
SYDNEY C.S. XUAN[1], CHUA WENYI[1], SURESH GOBEE[1,*],
VICKNESWARI DURAIRAJAH[2]

[1]School of Engineering, Asia Pacific University of Technology and Innovation,
Bukit Jalil, 57000 Kuala Lumpur, Malaysia
[2]Faculty of Computing, Engineering & Technology, Asia Pacific University of Technology
and Innovation, Bukit Jalil, 57000 Kuala Lumpur, Malaysia
*Corresponding Author: suresh.gobee@staffemail.apu.edu.my

Abstract

Buildings contribute to a significant portion in energy consumption. As energy, consumption and carbon emissions rate rises due to the increasing population, a more sustainable solution was proposed. Solutions proposed as with a smart energy building management system, by monitoring and controlling the various systems in a building, which consumes power, by automating the systems to reduce power consumption while utilizing PV panels to generate clean energy at the same time to create a more sustainable environment and allows monitoring of data collected through IoT.

Keywords: Energy management, Green building, IoT dashboard.

1. Introduction

Energy is a lifeline and a critical component of many countries' social, economic, and sustainable growth. Due to Malaysia's rapid infrastructure development and economic expansion, this has continued to have an effect on the country's rising energy consumption. Population increase, per capita income, and demographic changes all have an effect on this energy demand (such as increasing urbanization and economic growth). Energy usage increased by 7.5% in 2012 and is anticipated to increase by 6 to 8% in the following years [1]. However, the issues of sustainability, energy security, rapid depletion of indigenous fossil resources, and climate change have compelled the government to incorporate energy efficiency in buildings into its policy road plan. Thus, effective energy use is critical for maintaining existing energy supplies and so alleviating rising energy demand.

Additionally, Malaysia was placed 52 on the 2009 climate change performance index, a metric that examines and compares 90 percent of the world's 57 countries' CO_2 emissions [2]. The building sector appears to be one sector that is addressing the aforementioned difficulties. Given that buildings and the built environment account for a sizable portion of greenhouse gas (GHG) emissions.

The aforementioned difficulties can be addressed by energy-efficient building practices. In Malaysia, buildings absorb 14% of total energy and 53% of all electrical energy consumed in the residential and commercial sectors which are mostly comprised of buildings which can be seen as shown in Fig. 1. Through which the statistics was provided by energy commission Malaysia (2019), the provided figure may include some non-building sources that contribute relatively little, such as water and sewage treatment, street lighting, and losses. Additionally, industrial building-related facilities have been excluded, which will very certainly raise the share of energy consumed by buildings. As a result of the resources used to create electricity, it is a significant contributor to the country's greenhouse gas emissions. As a result, energy efficiency in buildings is critical for reducing energy consumption and enhancing local environmental sustainability.

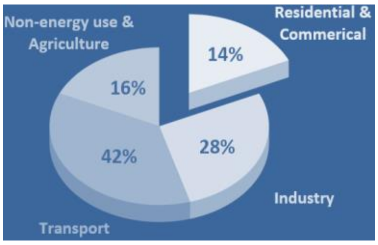

Fig. 1. Energy consumption in Malaysia.

2. Methodology

2.1. Automated MVAC monitoring control system

The overview of the entire block diagram of the automated MVAC system, as shown in Fig. 2. The indoor carbon dioxide, temperature, and humidity are the input variables of the system that will be monitor in this system. Whereby the input variables will be converted into input signals by respective sensors. Then, transmitted into a microcontroller.

Hence, the control speed of the actuators, such as the air conditioner and exhaust fan motor, will depend on the current condition of the given input signal in order to optimize indoor air quality. Thus, the microcontroller will give the low-current control speed signal to the motor driver. The motor driver will amplify it up into a high-current signal to control and drive the motor with its respective speed.

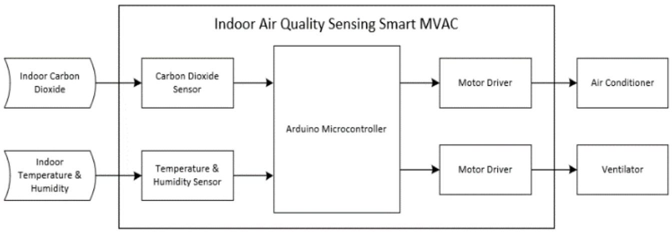

Fig. 2. Block diagram of indoor air quality sensing smart MVAC.

Figure 3 shows the flow chart of the entire MVAC monitoring control system. Inside this MVAC monitoring control system is using fuzzy logic. Therefore, once the data collection from sensors, it will go through fuzzification of data. Fuzzification is a process of converting a crisp input value to a fuzzy value that is performed by using the information in the knowledge base. Then it will go through fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. However, the rule base is a pre-built inference rule to infer an output based on input variables. Next will be going through the defuzzification process, which obtains a single number from the output of the aggregated fuzzy set that is used to transfer fuzzy inference results into a crisp output. Lastly, the values of the control speed of air conditioner and exhaust fan are obtained.

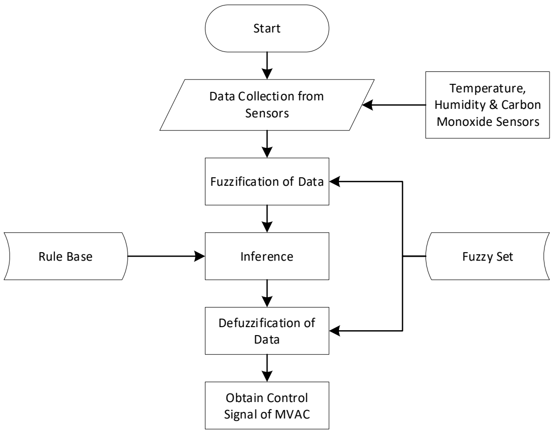

Fig. 3. Flow chart of MVAC monitoring control system.

2.2. Automated lighting monitoring control system

The block diagram (Fig. 4) presents the overall view of the smart lighting system as inspired from [3]. As shown in the input, this system is control via light intensity or motion. The said variables will be read and fed into the microcontroller using LDR and PIR sensors to determine the processes to be executed by the actuators where in this case are the motor driver, motor and lights.

To clarify, the lights mentioned are of different purposes and will be functioning in different compounds of the building. For the light brightness control, the lights mentioned are the ones in rooms where natural light can be made use of while the motion sensing components will be incorporated in compounds that are rarely used like the emergency stairs.

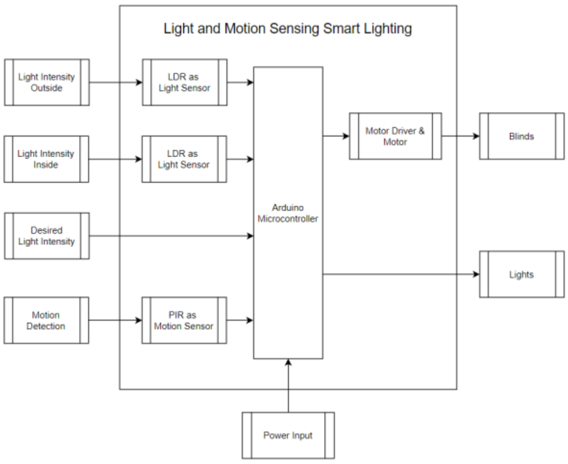

Fig. 4. Block diagram of the automated lighting system.

2.2.1. Flowchart of light sensing smart lighting system

Figure 5 shows the trigger to start the system is through the first diamond or decision block indicating that the illuminance outside also known as the natural light must be able to fulfil the desired brightness. Only then, the other part of the circuit will be activated to start monitoring and controlling the system by providing instructions for the movement of the blinds as referred to the desired brightness. When the main trigger mentioned fails, the second circuit will then be turned off and proceed to close the blinds entirely while turning the room lights on.

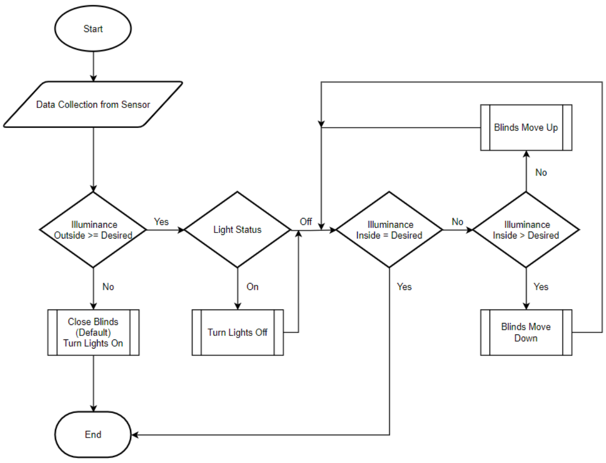

Fig. 5. Flowchart of light sensing smart lighting system.

2.2.2. Flowchart of motion sensing smart lighting system

For the motion sensing system, logic flow (inspired from [4]) as shown in Fig. 6 is very direct. If there's motion detected which is assumed to be people in the area, the lights will turn on until some delay after no motion is detected.

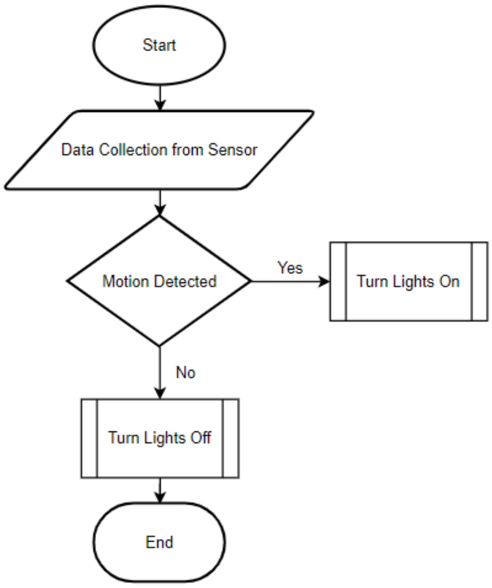

Fig. 6. Flowchart of motion sensing smart lighting system.

2.3. Optimization of photovoltaic energy system

The proposed methodology can be seen as the following block diagram in Fig. 7:

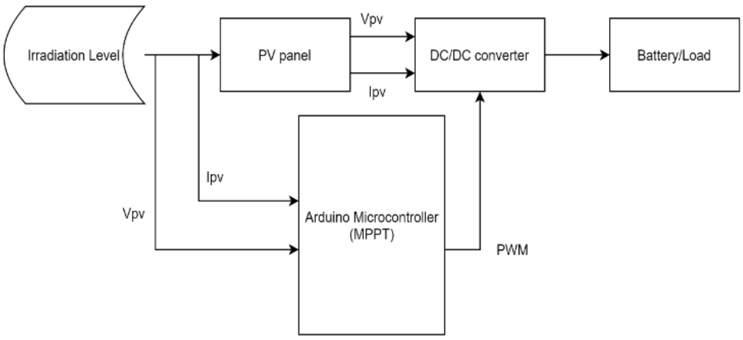

Fig. 7. Block diagram of PV system.

The overall block diagram shows that the PV panel affected by the change in the surrounding irradiation level. Whereby the photovoltaic voltage and current will be measured by sensors and the values are then passed on to the DC/DC converter as well as the Arduino UNO microcontroller due for the MPPT algorithm to calculate the suitable PWM signal to be sent to the DC-DC converter to step-up the output voltage for the charge battery.

2.3.1. Modelling of the PV panel in Proteus

To model the photovoltaic panel in Proteus, its electrical model is constructed whereby a voltage controlled current source is connected in parallel with a diode (its SPICE code is modified to match the PV panel's specification), and then two resistors are connected in series and parallel to model the shunt and series resistors. Hence, to mathematically represent a photovoltaic panel, we extract the basic equation from the panel's equivalent circuit following Figs. 8 and 9 [5]:

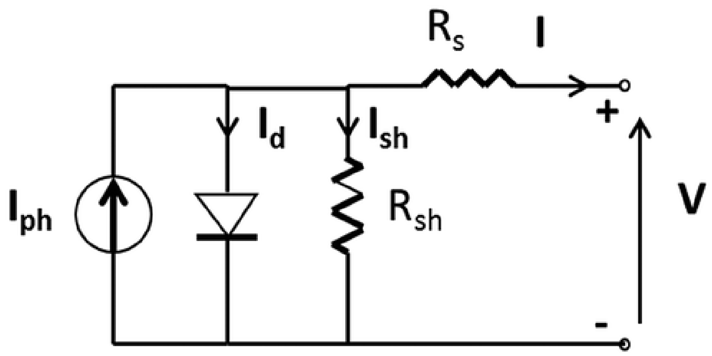

Fig. 8. PV equivalent circuit.

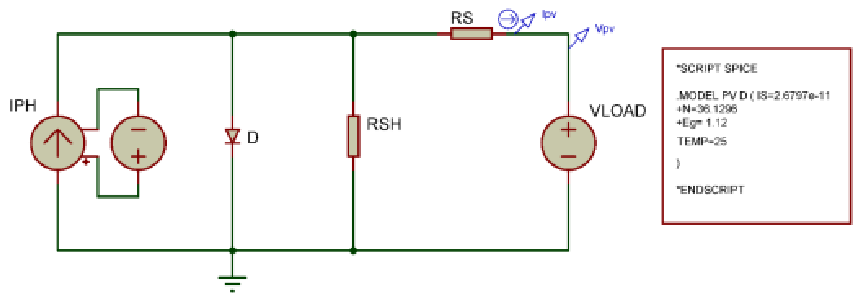

Fig. 9. PV equivalent circuit in Proteus.

As illustrated in Fig. 9, this model utilizes a diode with a modified spice code, as it is necessary to modify the values of the saturation current (Is), the ideality factor, the number of cells, and the bandgap energy in the Spice code to adhere to the TDC-M20-36 panel specification provided. The spice script was created using Proteus's "Text Script mode."

2.4. Modified INC algorithm

The improved method was proposed by [5] as illustrated in Fig. 10, where all division computations are eliminated using logic and arithmetic operations. Therefore, the complexity of the method operation is decreased, allowing for the use of low-cost microcontrollers. Besides that, a rounding error of 0.06 is utilized to determine the MPP and which allows stabilizing the system at this point when the MPP is reached to reduce fluctuations at the MPP.

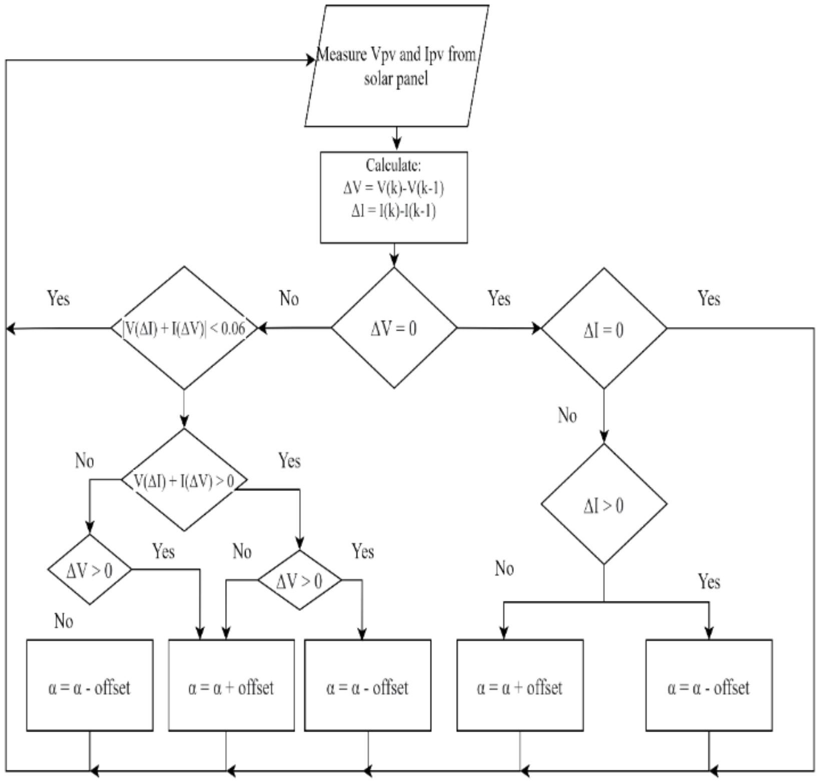

Fig. 10. Modified INC algorithm flowchart [5].

2.5. Modelling of boost converter

The DC/DC converter to be utilized would be a boost converter, to enhance the voltage generated by the PV panel. A boost converter is required to convert a direct current (DC) voltage (V) to another direct current (DC) voltage (Vo) where the circuit to be built connects the boost converter to the PV panel at the input and the load at the output such as Figs. 11 and 12:

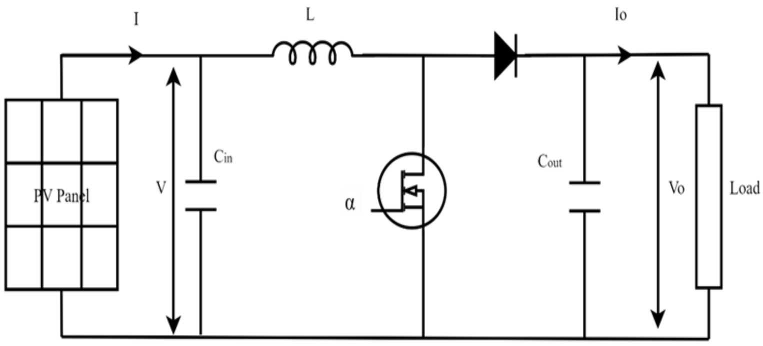

Fig. 11. Boost converter circuit design.

2.6. PV system overall flowchart

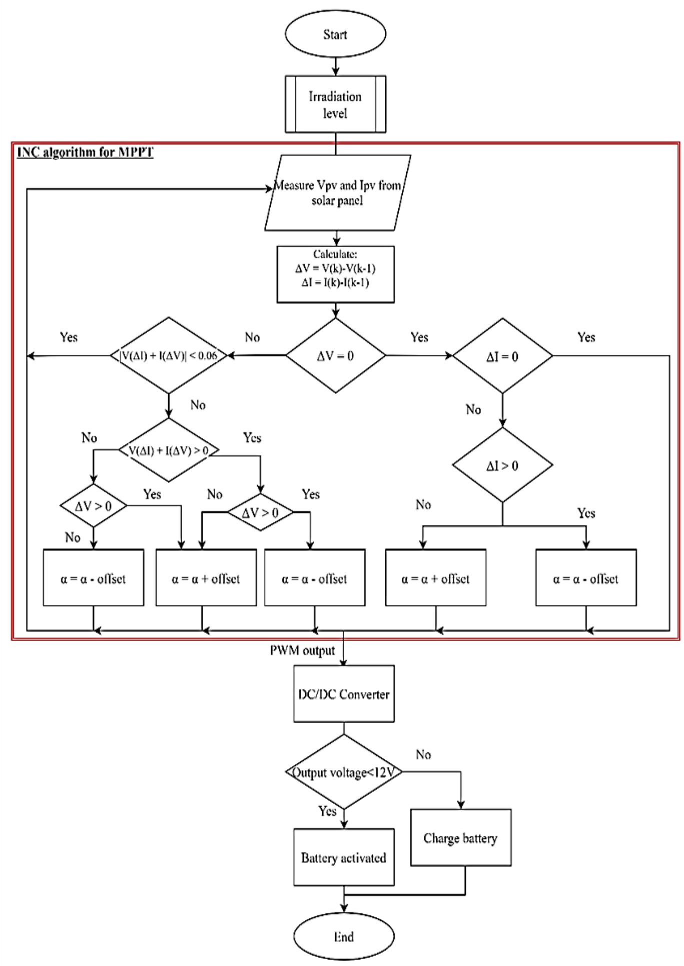

Fig. 12. PV system overall flowchart.

2.7. Setting up broker and data processing

The block diagram (Fig. 13) shows the general processes done in the processing of data obtained from the 3 systems mentioned above. Firstly, all the data that was obtained from the previous 3 data collection system will be published onto HiveMQ. All the data will be centralized into a Node-RED flow for data processing. The data processing includes splitting all the data obtained and assigning suitable topics for them before publishing them to the broker for the dashboard. Besides, the data processing stage also includes calculating the total energy consumption for each of the components, as well as the total energy consumption for the overall system.

Lastly, the data processing also includes calculating the total cost that is required to be paid due to the energy consumption, as well as the energy generated by the PV system and the cost saved from the PV system.

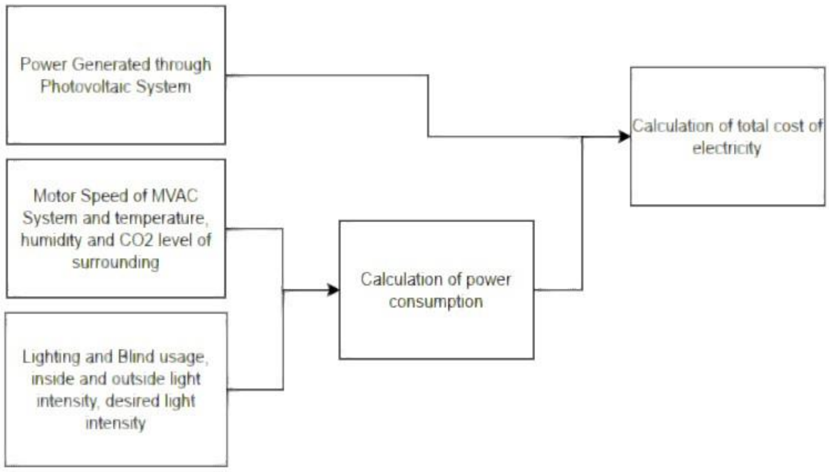

Fig. 13. Block diagram of data processing.

2.8. Flowchart of the data processing

Figure 14 shows the flowchart for the data processing. Firstly, the MQTT client is subscribed to the same topic that is used to publish the data obtained from the data collection systems. After that, the first step is to distinguish and split the incoming messages into their respective systems, which is the MVAC system, the Smart Lighting System and the PV System. After splitting them into their own system, the content of the message will be split according to their type. For example, for the Smart Lighting system, the status of the light, as well as the outdoor lux value and the indoor lux value will be split. Moving on, the energy consumption will be calculated based on the status of the systems. Lastly, the total cost will be calculated after calculating the overall energy consumption of the system. When the data processing stage is done, the message will be published into MQTT broker for the dashboard.

3. IOT Dashboard Development

In this section, the block diagram of data visualisation stage was shown in Fig. 15. The leftmost side of the blocks are the outputs of data processing, which includes the lighting system for stairs, lighting system for Level 1 and Level 2, PV system, MVAC system for Level 1 and Level 2 as well as the overall system. All these data are then being transferred to a MQTT broker, HiveMQ that has done by Gideon in the previous section and the nodes in Node-Red will be receiving these data by selecting the right server for the broker. After that, a few measurements which have been divided into 5 groups (stairs, Level 1, Level 2, PV system and overall) are created to serve as a container for the data. Moving on, InfluxDB data source is added into the Grafana dashboard by using InfluxQL as the query language. With the appropriate query language and the measurements that have been created, Grafana dashboard will display and graphs the available data from the node red. Other than that, the dashboard is separated into three sections where the first one is the main page that includes main parameters of the system, whereas the second and third are level 1 and 2 respectively.

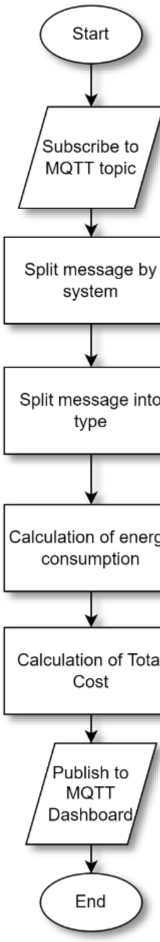

Fig. 14. Flowchart for data processing.

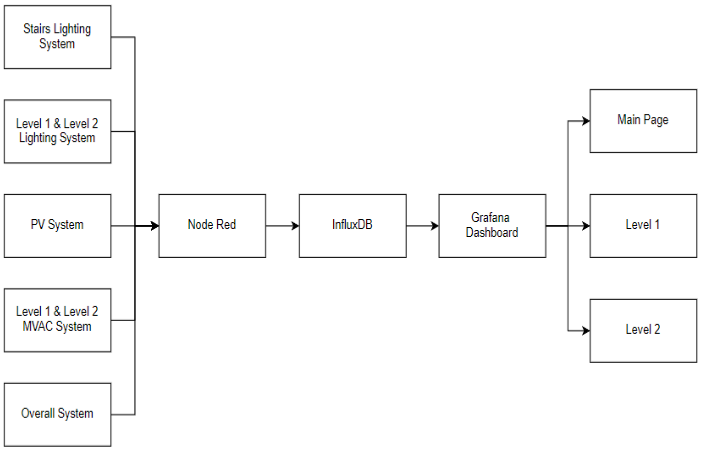

Fig. 15. Block diagram of data visualisation stage.

3.1. Flowchart of IOT dashboard

The flowchart of the development of IoT dashboard illustrated in Fig. 16. After the outputs from other devices has been processed, they will be sent to the Node-Red using built in MQTT nodes with MQTT broker, HiveMQ to receive data for the particular topics. The data will then go through data organization using ‘function’ nodes for classification. As said, the data are then stored in the database, InfluxDB with specified measurements then to the Grafana dashboard for data visualization.

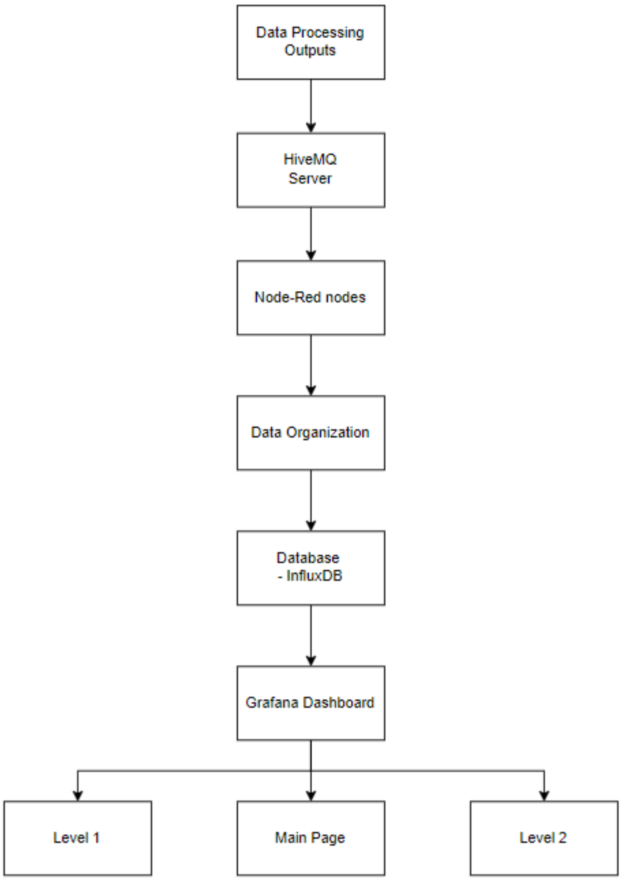

Fig. 16. Flowchart of IoT dashboard.

3.2. System implementation

3.2.1. Automated MVAC monitoring control system

Figure 17 shows the circuit diagram of the entire automated MVAC system by using Proteus. Inside this circuit diagram, there are two sensors: the CO_2 gas sensor as MQ-135 and the temperature and humidity sensor using DHT-22 connected to the microcontroller as input analog signal. In addition, the connected motor driver is used to receive the control PWM signal from the microcontroller to control the speed of two 12V DC motors represented as air conditioner motor and exhaust fan motor.

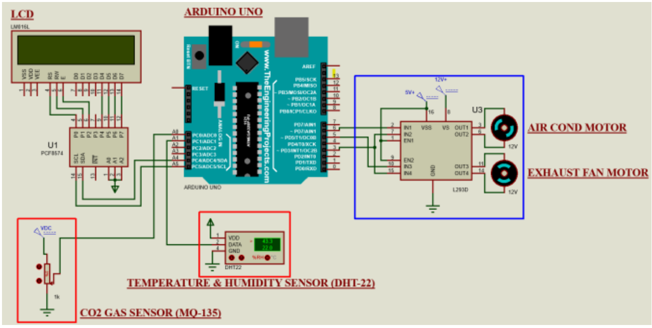

Fig. 17. Overall circuit design of automated MVAC system.

As mentioned initially, this automated MVAC system is used to optimize indoor air quality by implementing the AI fuzzy logic approach. AI has been applied in smart building management and is reviewed by [6, 7]. It is shown that AI can increase the energy performance of the building.

Inside the implementation of fuzzy logic of this system consists of three input variables, and two output variables were created using Fuzzy Logic Designer in MATLAB software. As shown in Fig. 18 is the Fuzzy Inference System (FIS) properties will be implemented in the automated MVAC system, whereby it has temperature, carbon dioxide concentration, and humidity as input, while the air conditioner speed and exhaust fan speed as output. The FIS type that will be used is Mamdani fuzzy inference system for processing the fuzzy logic data.

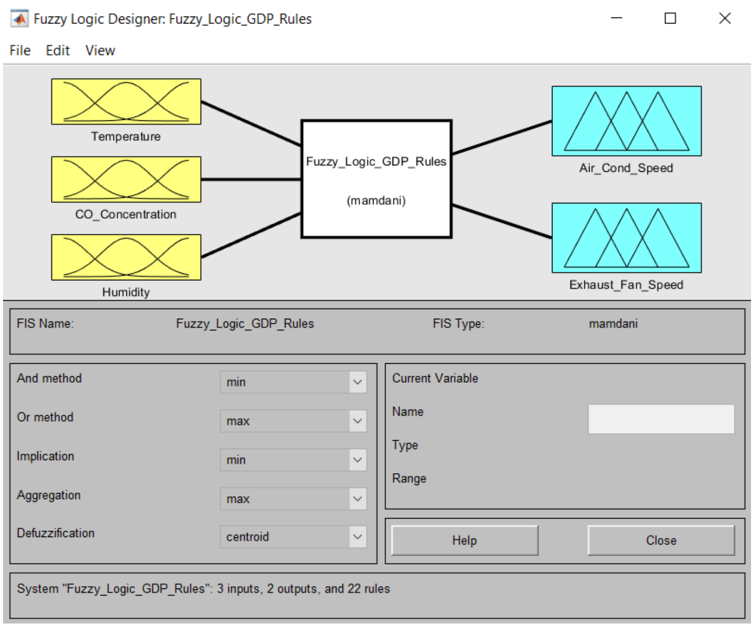

Fig. 18. FIS properties in automated MVAC system.

Figure 19 shows the input membership function plot of temperature. It has three member functions which are low (Trapezoid), medium (Triangle), and high

(Trapezoid). The temperature assumption made here was within the working temperature that an air conditioner can reach, and the room temperature was consider from 16 ºC to 32 ºC of indoor temperature only. The position and range of each membership function are referencing the research journal by [8].

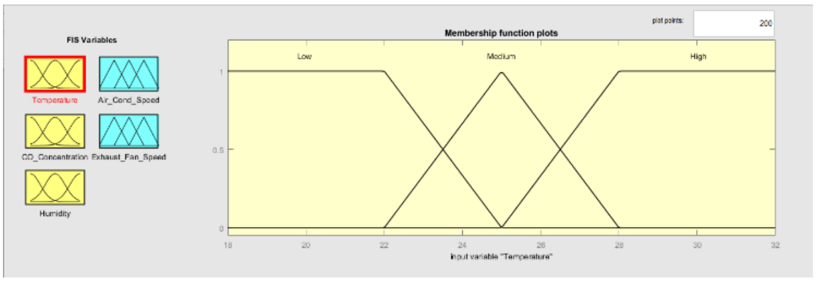

Fig. 19. Input membership function plots of temperature.

Figure 20 shows the input membership function plot of carbon dioxide. It has three member functions which are low, medium, and high all in trapezoid as changes are gradual for the gas concentration. The carbon dioxide concentration assumption made here was within the working range of the MQ-135 carbon dioxide sensor, which is 0 ppm to 200 ppm only was considered.

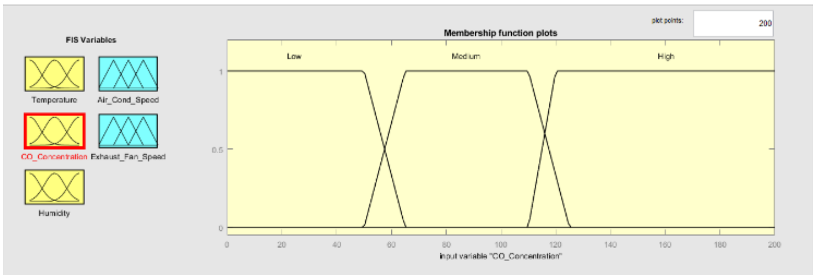

Fig. 20. Input membership function plots of carbon dioxide.

Figure 21 shows the input membership function plot of humidity. It has four member functions which are low, medium, high, and very high. The humidity level assumption made here was the range of humidity from a percentage of 0 to 100 was considered. The position and range of each membership function are referenced on the level of healthy relative humidity range according to the Environmental Protection Agency.

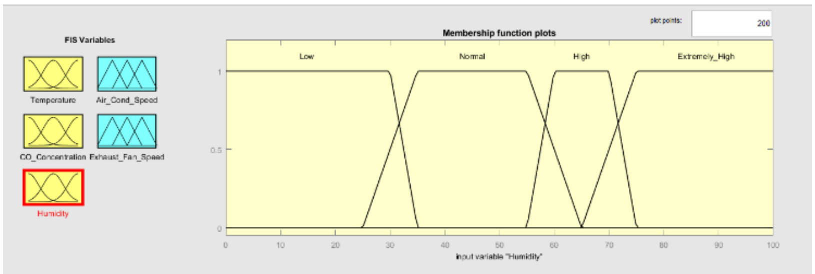

Fig. 21. Input membership function plots of humidity.

Figure 22 shows the output membership function plot of air conditioner speed. It has three member functions which are slow, moderate, and fast are triangular shapes as change is fan speed will be fast. The air conditioner speed assumption made here was the range of output PWM signal that the microcontroller could provide that ranges from 0V to 5V, which is 0 and 255 values were considered.

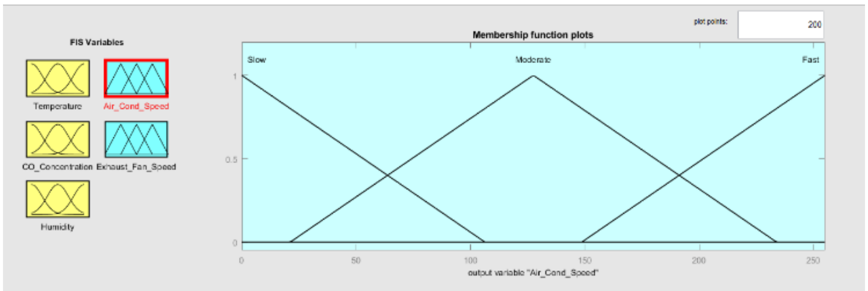

Fig. 22. Output membership function plots of air conditioner speed.

Figure 23 shows the output membership function plot of exhaust fan speed are trapezoidal shape as changes in fan speed is gradual. It has four members functions, which are very slow, slow, moderate, and fast. The exhaust fan speed assumption made here was also the range of output PWM signal that the microcontroller could provide that ranges from 0 V to 5 V, which is 0 and 255 values were considered.

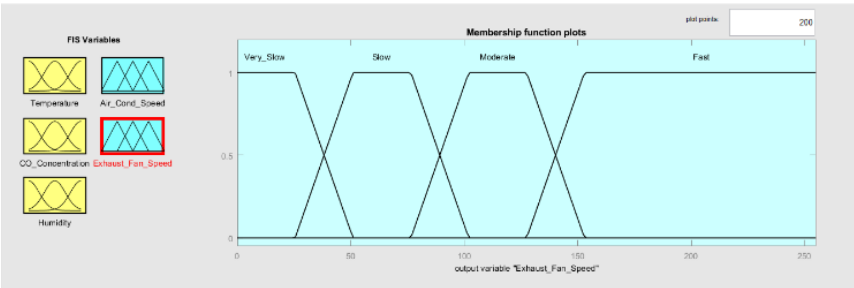

Fig. 23. Output membership function plots of exhaust fan speed.

3.2.2. Testing and results

Figure 24 shows the 22 rules that implemented to determine the outcome of fuzzy logic. Inside here were the combinations of three "AND" relationships of input rules that determine the type of output speed of air conditioner and exhaust fan speed. These rules were created based on various conditions, the supposed speed of the air conditioner and exhaust fan.

Figure 25 shows the result of fuzzy rules outputs speed when given specific inputs conditions inside the MATLAB Fuzzy Logic Designer. The output will only be considered once the red line of input fulfils three conditions since the "AND" rules are implemented.

Figure 26 shows the result of fuzzy rules outputs speed when at specific inputs conditions. Based on the sample of result, inside the LCD display, it displayed "T" is temperature, "H" is humidity, "C" is the carbon dioxide concentration, whereas the “AS” is the air conditioner speed and “ES” is the exhaust fan speed. Both of the speeds displayed in the LCD display meet the result shown in the bottom right-side of the fuzzy rules output result when given the same input condition. Therefore, the implementation of fuzzy logic into the microcontroller to perform the smart MVAC system is successful.

Table 1 shows the tabulation of data when 12 different conditions of temperature, carbon dioxide, and humidity, the output of air conditioner speed, and exhaust fan speed given by the microcontroller, as shown in the table.

1. If (Temperature is Low) and (CO_Concentration is Low) and (Humidity is Normal) then (Air_Cond_Speed is Slow)(Exhaust_Fan_Speed is Very_Slow) (1)
2. If (Temperature is Medium) and (CO_Concentration is Low) and (Humidity is Normal) then (Air_Cond_Speed is Moderate)(Exhaust_Fan_Speed is Very_Slow) (1)
3. If (Temperature is High) and (CO_Concentration is Low) and (Humidity is Normal) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Very_Slow) (1)
4. If (Temperature is Low) and (CO_Concentration is Medium) and (Humidity is Normal) then (Air_Cond_Speed is Slow)(Exhaust_Fan_Speed is Moderate) (1)
5. If (Temperature is Low) and (CO_Concentration is High) and (Humidity is Normal) then (Air_Cond_Speed is Slow)(Exhaust_Fan_Speed is Fast) (1)
6. If (Temperature is Low) and (CO_Concentration is Low) and (Humidity is Low) then (Air_Cond_Speed is Slow)(Exhaust_Fan_Speed is Very_Slow) (1)
7. If (Temperature is Low) and (CO_Concentration is Low) and (Humidity is High) then (Air_Cond_Speed is Moderate)(Exhaust_Fan_Speed is Moderate) (1)
8. If (Temperature is Low) and (CO_Concentration is Low) and (Humidity is Extremely_High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
9. If (Temperature is Low) and (CO_Concentration is Medium) and (Humidity is High) then (Air_Cond_Speed is Slow)(Exhaust_Fan_Speed is Fast) (1)
10. If (Temperature is Medium) and (CO_Concentration is Medium) and (Humidity is High) then (Air_Cond_Speed is Moderate)(Exhaust_Fan_Speed is Fast) (1)
11. If (Temperature is High) and (CO_Concentration is Medium) and (Humidity is High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
12. If (Temperature is Medium) and (CO_Concentration is Low) and (Humidity is High) then (Air_Cond_Speed is Moderate)(Exhaust_Fan_Speed is Moderate) (1)
13. If (Temperature is Medium) and (CO_Concentration is High) and (Humidity is High) then (Air_Cond_Speed is Moderate)(Exhaust_Fan_Speed is Fast) (1)
14. If (Temperature is Medium) and (CO_Concentration is Medium) and (Humidity is Extremely_High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
15. If (Temperature is High) and (CO_Concentration is High) and (Humidity is Low) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
16. If (Temperature is High) and (CO_Concentration is High) and (Humidity is Extremely_High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
17. If (Temperature is Low) and (CO_Concentration is High) and (Humidity is Extremely_High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
18. If (Temperature is Medium) and (CO_Concentration is High) and (Humidity is Extremely_High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
19. If (Temperature is High) and (CO_Concentration is Low) and (Humidity is Extremely_High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
20. If (Temperature is High) and (CO_Concentration is Medium) and (Humidity is Extremely_High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
21. If (Temperature is High) and (CO_Concentration is High) and (Humidity is Normal) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)
22. If (Temperature is High) and (CO_Concentration is High) and (Humidity is High) then (Air_Cond_Speed is Fast)(Exhaust_Fan_Speed is Fast) (1)

Fig. 24. Fuzzy rules in automated MVAC system.

Fig. 25. Result of fuzzy rules output.

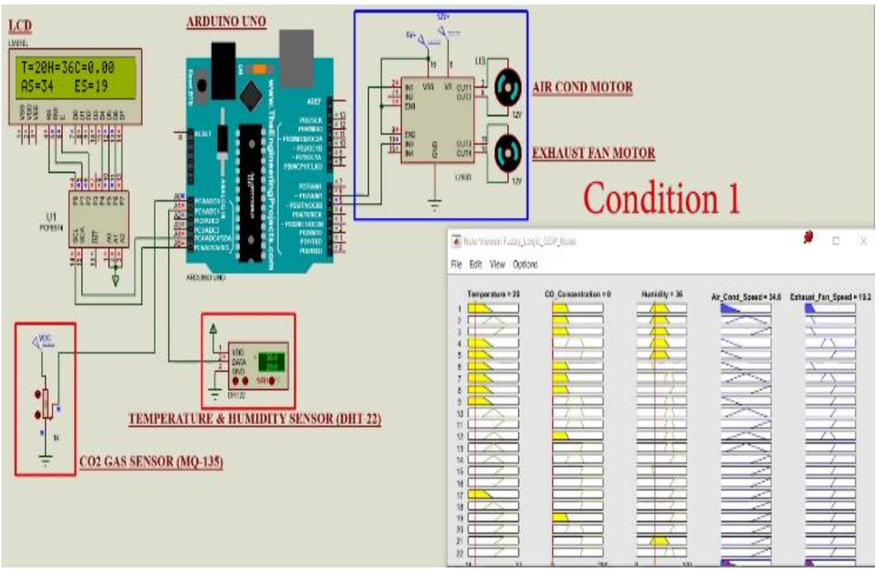

Fig. 26. Sample of result.

Table 1. Tabulation of data with 12 various conditions.

Conditions	Temperature (°C)	Carbon dioxide (ppm)	Humidity (%)	Air cond Speed	Exhaust fan speed
1	20	0.0	36	34	19
2	25	0.0	36	127	19
3	29	0.0	36	220	19
4	20	57.4	45	40	75
5	20	90.9	45	34	114
6	20	159.0	45	34	198
7	20	22.0	45	34	19
8	20	22.0	60	108	85
9	20	22.0	86	220	198
10	25	86.6	86	220	198
11	30	200.0	90	220	198
12	25	100.0	65	127	198

4. Automated Lighting Monitoring Control System

Constructional details and working principle

Both circuit diagrams including the Light Sensing and Motion Sensing systems displayed in Fig. 27. From Fig. 27, three sensors can be seen including a PIR and two LDR sensors. However, the only relation that they have are the fact that they are connected in the same Arduino Microcontroller and will not interfere in each other’s processes.

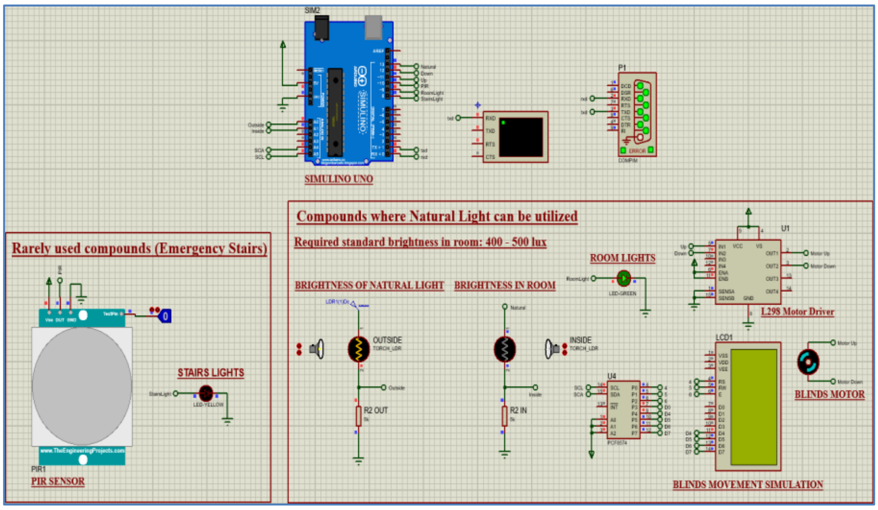

Fig. 27. Overall circuit design of the system in Proteus.

5. Testing and Results

5.1. Light sensing smart lighting

This system as mentioned previously aims to reduce electricity usage at places where natural light can be utilized. In the simulation, the desired brightness in the room is set to be at the range of 400-500 lux. This setting must be in a range instead of a constant value as brightness in a particular area is never constant and can easily be affected by its surroundings. From the circuit seen in Fig. 27, two LDR were connected in series with two static resistors with the voltage output taken in between them. However, only one of them is connected to a constant voltage source while the other voltage supply is controlled by the microcontroller. The justification of this design is to have the LDR circuit in the room to be turned off when the lights in the room are on as the desired brightness will be fulfilled. The actuators as seen include the stairs lights, room lights and the blinds motor along with its driver. An LCD display is included to be used to simulate the movement of the blinds for this simulation purpose. The logic of this system is as tabulated in Table 2.

Table I. Logic of light sensing system.

Case	Brightness outside (lux)	Brightness inside (lux)	Room light	Blinds
A	< 400	-	On	Fully Closed
B	> 400	< 400	Off	Move Up
C	> 400	> 400 & < 500	Off	Stop
D	> 400	> 500	Off	Move Down

5.2. Optimization of photovoltaic energy system

Constructional details and working principle.

The overall PV system design for the Proteus simulation can be seen as shown in Fig. 28, through which the PV panel has a subcircuit as shown in Fig. 8. at the methodology section for modelling of PV panel. Then, the output of the PV panel is connected to the boost converter to step up the voltage of the PV panel to supply to the load and matching the voltage of the battery. However, it is required for the implementation of the INA169 analogue DC current sensor and the B25 voltage sensor module to measure the actual voltage and current of the PV panel and send the data to the Arduino UNO, as the B25 voltage sensor is a compact and extremely handy module that reduces any input voltage by a factor of 5 using a potential divider. This enables us to use a microcontroller's Analog input pin to monitor voltages greater than the microcontroller is capable of perceiving.

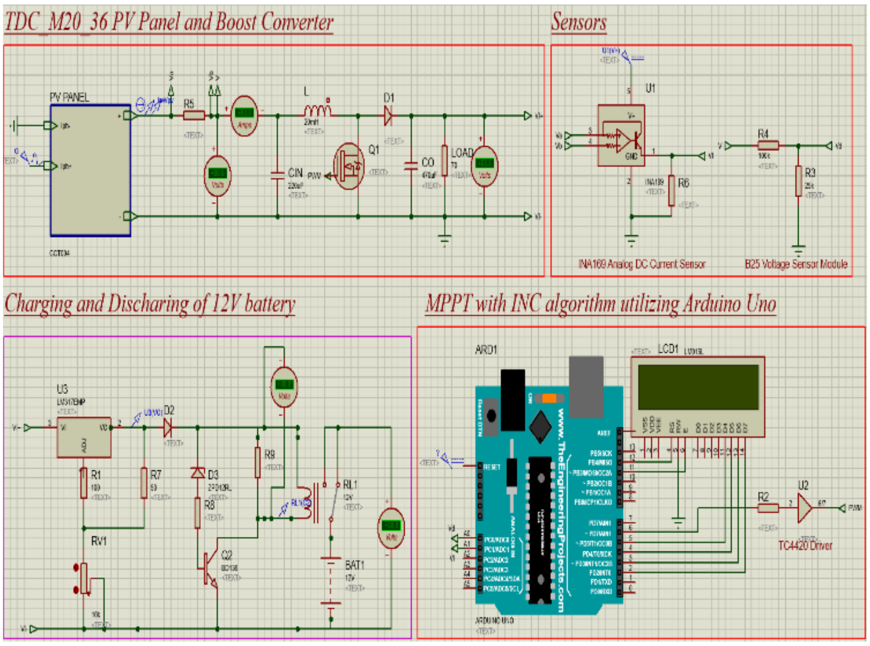

Fig. 28. Overall PV system design with battery.

For instance, with a 0 V – 5 V analogue input range on the Arduino UNO, it is necessary to lower the voltage using the voltage sensor and then compute the real value through a series of calculations for tracking the MPP. For testing reasons, the change in irradiation levels is rapid and drastic, to make this possible, a power input generator is fed to the PV panel to simulate the irradiation level to the following parameters shown in Table 3.

Table 3. Irradiation level fed To PV panel.

Time (s)	Irradiation (w/m^2)
0.0	1200
2.0	800
4.5	400
7.0	1000
9.5	0

Moving on, the construction of the charge battery circuit design is mainly a voltage divider rule to control the voltage going through the LM317EMP, while there is a Zener diode for voltage regulation across the base of the transistor, as a protection for the transistor switch. The transistor switch basically is responsible for triggering the relay which is connected to the battery. Through the implementation of voltage divider rule, as the voltage is greater than 12 V provided from the boost converter and PV panel, the battery charges. However, if the voltage falls below 12 V, the relay will be triggered, and the battery will be discharging instead and isolated by the relay. This is because if the battery remains connected to the boost converter, there would be a back flow of current as the higher potential from the battery would flow back to the lower potential circuit which is the boost converter and PV panel.

5.3. Power Generated from PV panel with MPPT controller

As mentioned previously, the PV panel would depend mostly on the surrounding irradiation level. The irradiation levels to be simulated to determine the functionality of the PV system are as shown in Table 4. From these irradiation levels, it is expected for the power generated to be directly proportional to the change in irradiation levels. Whereby the following graph shows the change in power produced with respect to the change in irradiation level following the irradiation levels in Table 4.

Table 4. Power generated from PV panel under fast changing irradiation levels.

Irradiation (W/m^2)	Average power generated (W)
1200	20.5
800	12.0
400	4.3
1000	17.0
0	0

The results show that the MPPT can track the maximum power point of the PV panel throughout the rapid change of irradiation level which makes the PV system more efficient with a low cost MPPT controller to be implemented in buildings. The result from the graph achieves a steady state response of 0.4 s or lesser when the irradiation level changes which is acceptable.

5.3.1. Functionality of battery charging

The circuit designed mostly depends on voltage divider rule to bias the voltage supplying the battery to ensure that sufficient voltage is charging the battery, with the implementation of diodes, transistor, and relay. As shown from Fig. 29, the power is low during 4.5 s of the simulation time, while the power is 0 W at 9.5 s. Hence, the voltage is lesser than 12 V which is the battery rating. Hence, during this period the relay in the charge battery circuit would isolate the battery and the battery starts to discharge from 4.5 s to 7 s and reverts to charging at 7.1 s when the voltage is high again as shown in Figs. 30 and 31:

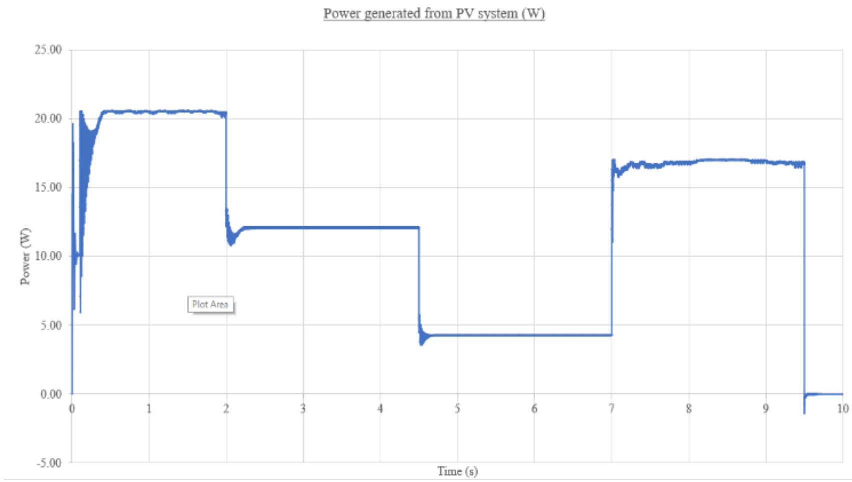

Fig. 29. Power generated from PV system.

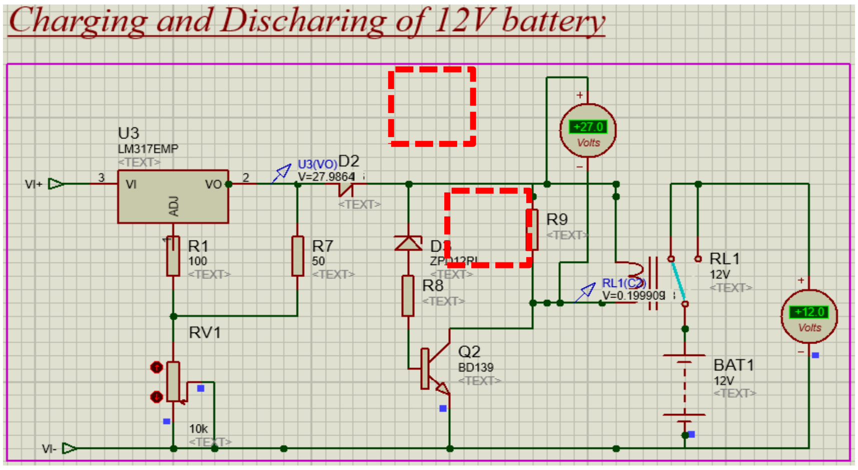

Fig. 30. Battery is charging.

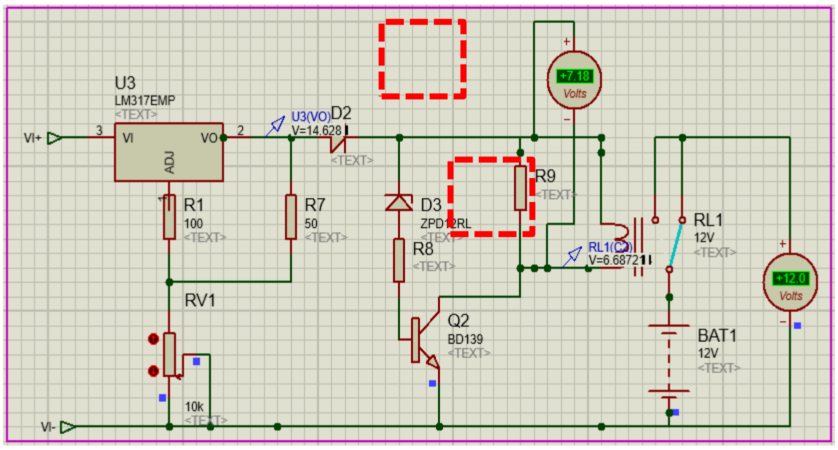

Fig. 31. Battery is isolated and discharging.

5.3.2. Integrated system and enhancement

In this chapter, PV system with the addition of MPPT controller, smart lighting and MVAC systems are integrated to the IoT dashboard through the MQTT broker to Node-Red for data processing. Thus, allowing the IoT dashboard for monitoring and data collection to allow the users to be aware of the current energy generated and harvested through a period of time, whereby the computation of costs, power consumption and generated are done with Node-Red.

5.3.3. Overall block diagram and working principle

The overall block diagram in Fig. 32 illustrated the three stages of the entire system. This includes the initial stage which they are made up of the Indoor Quality Sensing Smart MVAC, Smart Lighting and PV System. The similarity of these three systems mentioned includes having an arrow pointing at the HiveMQ MQTT Broker indicating that there will be data being transferred into it. Therefore, it can be said that they are the data collection component of the system.

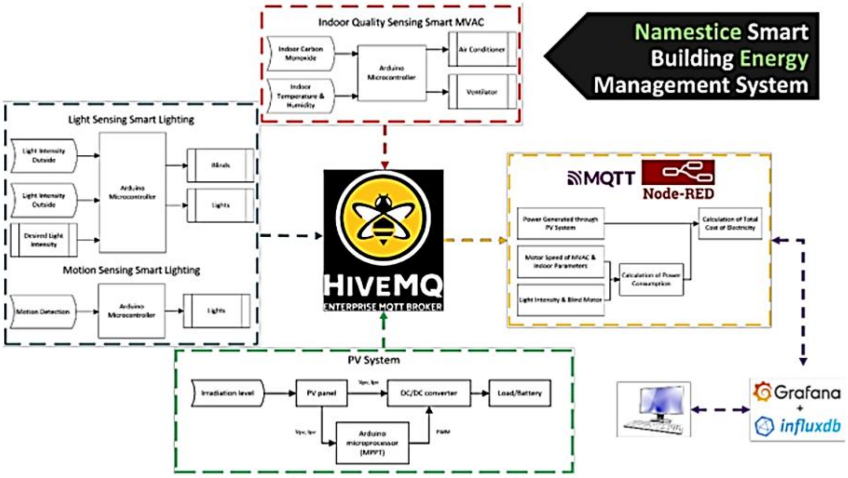

Fig. 32. Overall block diagram of the project.

After that, is a system which has an arrow pointed at from the MQTT broker. This system is where all the data collected will be centralised and gathered in a Node-Red environment for the data pre-processing as well as the processing stage. The pre-processing stage in this case involves the procedure of breaking down and classifying all the data received while the processing stage will involve mathematical functions to convert these data into more useful information like the power consumed, costs and so on. Then, the output of the system will be sent to the final data visualization stage.

In the final stage, all data analysed will be stored in a time series database called InfluxDB. In the database, the data collected are organized in different 'Measurements' based on the pages to be used in the dashboard. Then, using a data visualization tool, Grafana all relevant data were plotted in time series line, area, bar charts and so on.

5.3.4. Integrated system

The final integrated system involves the converging of the data collection components including the MVAC, lighting and PV systems into the data processing stage where the outputs will be projected on the finalized dashboard. Therefore, only the dashboard will be discussed in this section as it is the visibly most obvious outcome of the integration as shown in Fig. 33.

During the development of the dashboard, some modifications were done to the initial design during the integration phase. These changes include the addition of two more Stat Panels for the display of both natural light and room brightness. Other than that, an additional table and time-series graph were included to provide a clearer picture on the power consumption of both lighting and MVAC system separately as shown in Fig. 34.

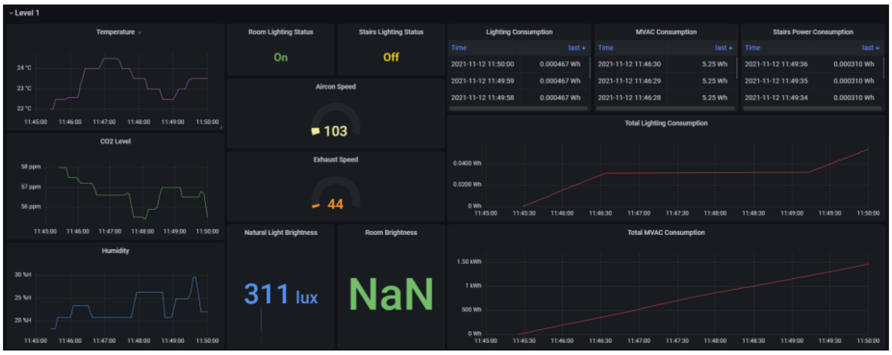

Fig. 33. Integrated IoT dashboard design for level 1.

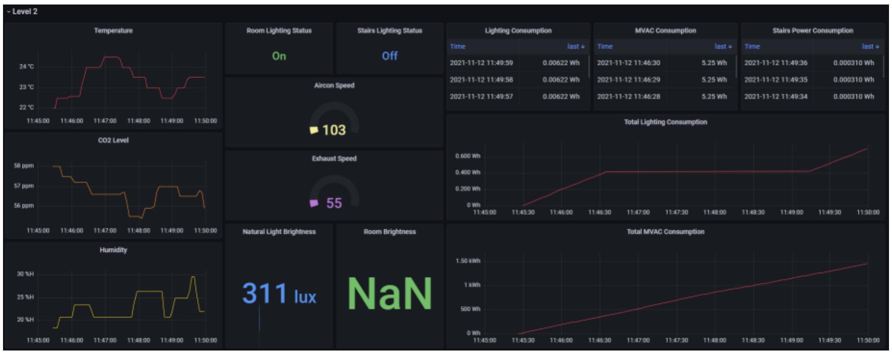

Fig. 34. Integrated IoT dashboard design for level 2.

5.3.5. Enhancement

Enhancement of this group design project including Artificial Intelligence (AI) and self-tracking algorithm that was implemented with the solar MPPT. The AI that is implemented is fuzzy logic algorithm which used to optimize the indoor air quality based on the created fuzzy rule base with 22 rules and also five membership functions, while the self-tracking Modified Incremental Conductance (INC) algorithm able to track the maximum power generated from solar panel under fast irradiation changes which is able to reduce power losses and maximize power generated due to the fluctuations that was reduced with the algorithm implemented.

6. Conclusion

In conclusion, the objectives of this project were achieved, such as to design an efficient PV system in harvesting power through fast irradiance changes. A smart lighting system which implements daylight strategy as well as motion sensing, and an MVAC system that is automated based on surrounding temperature, humidity and carbon dioxide. Moreover, the evaluation of the various systems was proven to be accurate and consistent as seen from the IoT dashboard. With the aim of creating a smart building energy monitoring and control system that is able to help client or users to save costs in the long run, through the reduction of energy consumption and offer convenience by monitoring the data through an IoT dashboard with its costs saved. For further improvement, the system may be implemented through a premium MQTT broker that is able to process data without disconnection as well as implementing a prototype to evaluate the practical data.

Acknowledgment

The authors would like to thank Suresh Gobee for providing his helpful and crucial opinions as well as feedback on making this project successful.

References

1. Maxwell, J.C. (1873). *A treatise on electricity and magnetism* (3rd ed). Oxford: Clarendon Press.
2. Dixon J. (2009). Is Climate Change On Malaysia's Agenda? Filed under: Climate Change.
3. Taufik, T.; and Hasanah, R. (2018). Light sensing smart blinds. *Proceedings of the 2018 Electrical Power, Electronics, Communications, Controls And Informatics Seminar (EECCIS)*. Batu, Indonesia.
4. Twumasi, C.; Dotche, K.A.; Banuenumah, W.; and Sekyere, F. (2017). Energy saving system using a PIR sensor for classroom monitoring. *Proceedings of the 2017 IEEE PES Powerafrica*. Accra, Ghana.
5. Motahhir, S.; Chalh, A.; Ghzizal, A.E.; and Derouich, A. (2018). Development of a low-cost PV system using an improved INC algorithm and a PV panel Proteus model. *Journal of Cleaner production*, 204, 355-365.
6. Chae, M.; Kang, K.; Koo, D.; Oh, S.; and Chun, J.Y. (2020). Fuzzy controller algorithm for automated HVAC control. *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC 2020),* 566-570.
7. Xie, X.; Lu, Q.; Herrera, M.; Yu, Q.; Parlikad, A.K.; and Schooling, J.M. (2021). Does historical data still count? Exploring the applicability of smart building applications in the post-pandemic period. *Sustainable Cities and Society*, 69.
8. Alanne, K.; and Sierla, S. (2022). An overview of machine learning applications for smart buildings. *Sustainable Cities and Society*, 76.
9. Stolojescu-Crisan, C.; Crisan, C.; and Butunoi, B.P. (2021). An IoT-based smart home automation system. *Sensors*, 21, 1-23.
10. Miah, S.; Islam, G.M.J.; Das, S.K.; Islam, S.; Islam, M.; and Islam, K.K. (2019). Internet of Things (IoT) based automatic electrical energy meter billing system. *Journal of Electronics and Communication Engineering*, 14(4), 39-50.

11. Stolojescu-Crisan, C.; Butunoi, B.P.; and Crisan, C. (2020). IoT based intelligent building applications in the context of COVID-19 pandemic. *Proceedings of the 2020 International Symposium on Electronics and Telecommunications (ISETC)*. Timisoara, Romania, 1-4.
12. Bhatt, J.; Jani, O.K.; and Bhatt, C.B. (2020). *Advances in 21st century human settlements*. Chapter: Automation based smart environment resource management in smart building of smart city. Springer, 93-107.
13. Chasta, R.; Singh, R.; Gehlot, A.; Mishra, R.G. and Choudhury, S. (2016). A smart building automation system. *International Journal of Smart Home*, 10(8), 91-98.
14. Nguyen, T.A.; and Aiello, M. (2013). Energy intelligent buildings based on user activity: A survey. *Energy and Buildings*, 56, 244-257.
15. Vijayan, D.S.; Rose, A.L.; Arvindan, S.; Revathy, J.; and Amuthadevi, C. (2020). Automation systems in smart buildings: a review. *Journal of Ambient Intelligence and Humanized Computing*, 1-13.