CHAPTER 3

PROJECT DEVELOPMENT

3.1 Introduction

3.1.1 Background of study

Smart Home Internet of Things is a project that aims to transform traditional homes into an automated, intelligent, and elegant home. The purpose of this project is to develop the sensors are connected to the internet to provide a better quality-of-life in terms of control and monitor the sensor to end-user in the house. This, will allow end-users to control or monitor their house with the push of a button on their smartphone. Also, Sensors that have been installed on the current project are RFID sensor, DHT11 sensor, photoresistor sensor, and flame sensor. Sensors that I have added are WI-FI and LEDs. The software used for the project is Arduino IDE and BLYNK as shown in Figure 1 below.

Figure 3.1: Software used: BLYNK and Arduino

The ESP8266 ESP-01s is a WI-FI module that has been installed on the Arduino Mega. There are many notable WI-FI modules that can be used on the Arduino Mega but ESP-01s is one of most popular choices for applying connectivity on most Arduino microcontroller. Having WI-FI connectivity can allow the sensors to be controlled and monitored by end-users. The ESP-01s is used to connect to the WI-FI where it will provide internet to the Arduino Mega. With the implementation of the WI-FI module, the Smart Home IoT project will provide an improvement in terms of convenience to end-users by controlling the sensors of the smart home using the Internet.

Furthermore, BLYNK has been implemented to the Smart Home IoT project to control the sensors. The reason why BLYNK was chosen is because BLYNK is an IoT platform / framework that allows end-users to control and monitor the smart home via smartphone. BLYNK offers many features like being able to make an user interface and widgets where it can interact with the sensors of the smart home. The LEDs that connected to the project now can be controlled by BLYNK where end-users can control the LEDs anywhere with the use of BLYNK's application or their website.

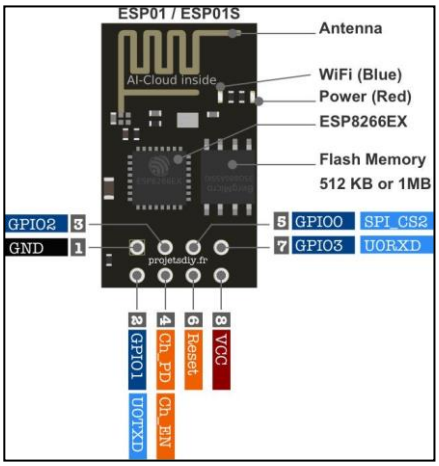


Figure 3.2: ESP-01s WI-FI module

Based on figure 2 above, the ESP-01S ESP8266 Wi-Fi Module have 8 pins with their specific functions when they are connected on each pin which are:

1. Voltage Common Collector (VCC): Supplies power to the module requiring 3.3V power source
2. Ground (GND): Provide grounding for the module
3. General-purpose input/output 1 (GPIO0): Determine the boot mode of the module which are programming mode, normal mode.
4. General-purpose input/output 2 (GPIO2): Serves as regular digital I/O pin.
5. Transmit data (TXD): Transmit pin for serial communication
6. Receive data (RXD): Receive pin for serial communication
7. Chip Power Down (CH_PD): Enables or disables the entire module
8. Reset (RST): Resets the module

An Integration with BLYNK API has made the smart home send any alert to smartphone and even emails. By that, if a certain condition has met on the sensors or the Arduino itself, it can send notification alert to smartphone and emails. For example, it will notify the smartphone when a fire sensor has detected a fire nearby or DHT11 sensor has detected an abnormal temperature will cause a notification to be sent to the smartphone. With BLYNK integration, it has made the smart home more user-friendly and increased in terms of safety.

3.1.2 Project Scope

The scope of this project is to expand the current Smart Home IoT features by adding various sensors and modules to the project which are a WI-FI module called ESP8266 ESP-01s. The primary objective is to integrate the smart home with an API called BLYNK, enabling end-users to effortlessly control and monitor their sensors using their smartphone. This integration aims to provide easy-access control over the smart home. This project will be focused on implementing an user interface to allow end-users to control and monitor the sensors on smart home project via a smartphone. Other than that, this project will also aim to make the smart home improve in terms of performance and efficiency.

3.1.3 Project Objectives

The main objectives of this Smart Home IoT project are to:

1. Implement an ESP8266 ESP-01s Wi-Fi Module into the existing smart home project to enable wireless communication and control between sensors and devices of the smart home.

2. Integrate the BLYNK API to allow end-user to control and monitor the smart home project via a smartphone application.

3. Design a user-friendly interface for end-users to control via the BLYNK application on the smartphone.

4. Optimize the code of the project to improve performance and efficiency of the smart home.

3.1.4 Literature Review

3.1.4.1 Internet of Things (IoT)

The Internet of Things (IoT) is rapidly evolving to this day where it enables the connection of physical devices to the internet and creating a smart ecosystem. IoT offers increased automation in various fields such as transportation, manufacturing, healthcare, and much more (Gubbi et al, 2013). However, there are weaknesses on IoT which are having risks on data privacy and security.

Also, most smart cities, smart homes, and smart industries are using IoT and it have change human interact with environment with the implementation of IoT. Basic components of IoT are real-time data monitoring, autonomous online, and predictive analysis have led to improvement in industries such as cost savings. (Atzori et al, 2010). IoT offers opportunities in automation, innovation, and efficiency to the industries (Caragliu et al, 2011).

In conclusion, Internet of Things (IoT) is very popular nowadays because it offers various benefits to the industry such as transportation, manufacturing and health care. However, there are weaknesses of IoT that people need to aware of which are data privacy and security.

3.1.4.2 Smart Homes

The concept of smart home has gained popularity in recent years, with the evolving of Internet of Things (IoT). A smart home is a home integrated with devices that are capable of automation, controlling the house via online, intelligent decision making (Khan et al, 2019).

Devices that are implemented to the smart home such as sensors and smart appliances are all connected to the network where it can communicate with the end-user and this will enhance comfort, security, safety, and much more to the homeowner. (Han et al, 2018).

In smart home, there are many different choices of devices that can be implemented on a smart home but are not limited to smart thermostats, smart bulb, security cameras, and voice assistants. A device that is connected to the network is an IoT device. (Chen et al, 2019)

3.1.4.3 Arduino in Smart Homes

Arduino microcontroller has gained popularity with smart home in the recent years. The reason why Arduino microcontroller is popular due to its affordability, flexibility, and ease-of-use. Smart home that uses Arduino offers less energy consumption, enhance security and overall convince to the homeowner. Arduino is an open-source project where it allows for customization and expansion possibilities such as enabling integration with Internet of Things (IoT) (Smith and Johnson, 2018).

A study has shown that Arduino is a popular choice due to its active community support and detailed documentation of the Arduino itself. There are wide range of sensors and module that Arduino are compatible with. Furthermore, programming language that is used on Arduino is C++ and it makes it accessible even for new learners (Johnson et al, 2018).

In conclusion, Arduino is one of the most popular choices for the development of a smart home due to its affordability, extensive community, easy programming language, energy efficiency, and much more.

No	Title	Key point	References
1	The Internet of Things: A survey	Purpose of Internet of Things (IoT)	Atzori et al, 2010
2	Internet of Things (IoT): A vision, architectural elements, and future directions	Basic components of Internet of Things (IoT)	Gubbi et al, 2013
3	Smart cities in Europe	Opportunities in Internet of Things (IoT)	Caragliu et al, 2011
4	Future of healthcare with smart homes integrated with internet of things (IoT) advances	Concept of smart home	Khan et al, 2019
5	A smart home energy management system based on internet of things	Connection and purposes of smart home	Han et al, 2018
6	A survey on smart home management technologies and applications	Various type of devices on smart home	Chen et al, 2019
7	Arduino-based home automation: An energy-efficient approach.	Popularity the use of Arduino microcontroller in smart home	Smith and Johnson, 2018
8	Comparative analysis of microcontrollers for home automation	Advantages using Arduino microcontroller on smart home	Johnson et al, 2018

Table 3.1: Summary of literature review

3.2 Design and Implementation

3.2.1 Hardware and Software Designs

3.2.1.1 Hardware components

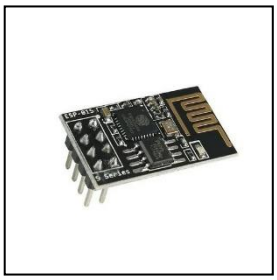
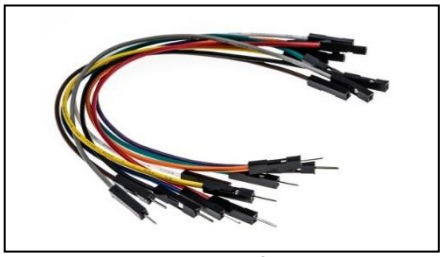
Hardware	Function
 ESP8266 ESP-01s	Provide Wi-Fi connectivity to the smart home
 3mm LEDs	LEDs for every room of the smart home project
 220Ω resistors	Resistors are used on LEDs to regulate voltage.
 Jumper wires	Provide connection from Arduino Mega to breadboard

Table 3.2: List of hardware components to be added on smart home

3.2.1.2 List of Sensors

Arduino Mega is a microcontroller board that offers large amount of input and output pins where connecting multiple sensors at once is not a problem for the Arduino Mega. For the current project, there are 4 sensors that already have been installed on the smart home which are RFID sensor, Flame sensor, DHT11 sensor, and photoresistor sensor. All the sensors are connected to the Arduino Mega via jumper wire. Sensors that have been newly added are ESP-01s Wi-Fi module and LEDs on every room.

1. ESP-01s Wi-Fi Module

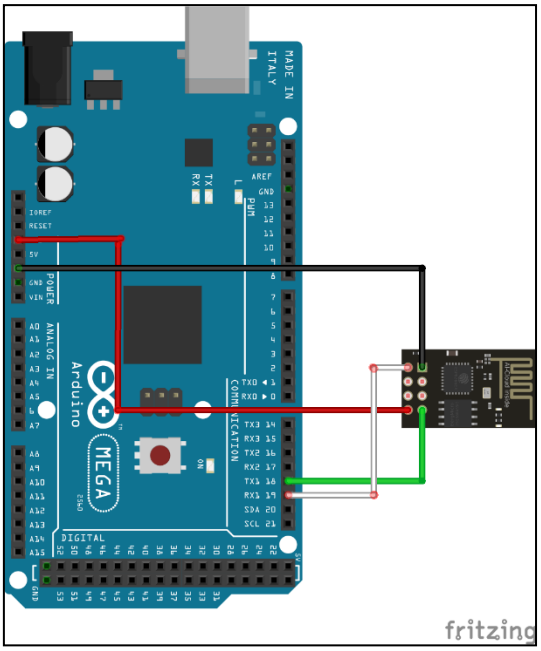


Figure 3.3: ESP-01s connection to Arduino Mega

Based on Figure 7, it shows a diagram of the connection between ESP-01s and Arduino Mega. The main purpose of the Wi-Fi module is to enable the smart home to connect to the internet and control from various devices. This module is required to let the smart home be integrated with BLYNK APIs where it will allow automated tasks, control sensors via smartphone, and much more.

2. LEDs

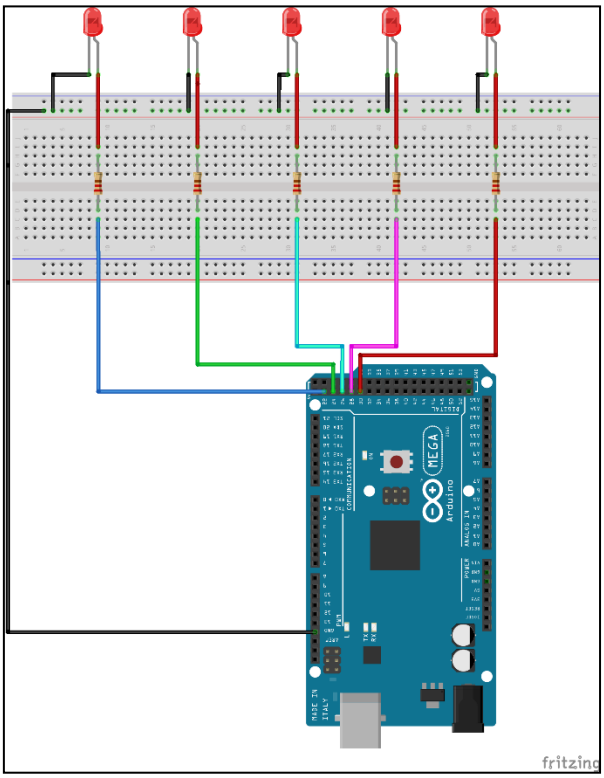


Figure 3.4: 5 LEDs connection to Arduino Mega

Based on Figure 5, it shows a diagram of the connection between 5 LEDs connection and Arduino Mega. All the LEDs are controlled via BLYNK API where we can control it via a smartphone and even on the BLYNK's website. There will be a user interface for end-users to control the LEDs on the BLYNK's application.

3.2.2 Software

3.2.2.1 Arduino IDE

Arduino IDE is used in this project because it allows us to write code and upload it to the Arduino Mega. Other than that, it offers a friendly user-interface for programming. All the codes have been coded via Arduino IDE and all the libraries have been gathered from the Arduino IDE integrated online library.

3.2.2.2 ESP8266 Flasher

ESP8266 flasher was used to flash new firmware to the ESP8266 ESP-01s due to firmware being corrupted upon arrival. There are multiple ways that you can flash ESP01-s which are connecting directly to the ESP01-s using USB to TTL converter or via Arduino Mega itself. For this project, the ESP01-s has been flashed through Arduino Mega microcontroller where it connects RX and TX pin directly to Arduino Mega allowing connection between ESP-01s and the ESP8266 Flasher.

3.2.2.3 Fritzing

Fritzing was used to design a circuit diagram to show connection between ESP-01s, and LEDs with Arduino Mega itself. The main purpose of Fritzing is to show how the connection works visually.

3.2.2.4 BLYNK application

BLYNK application was used on an android smartphone to control all the sensors of the smart home anywhere using Wi-Fi. BLYNK application offers a user-friendly interface where end-users can interact and monitor all the sensors of the smart home without any hassle.

3.2.3 Integration with BLYNK IoT

In this smart home project, I have successfully integrated the smart home with BLYNK IoT. There are many other choices of IoT that can be use on the smart home like ThingSpeak, Microsoft Azure, codebender, Frame, and much more but judging from the sensors that have been installed on the smart home, it is better to go with BLYNK since the smart home does not have a lot of data to be analyzed and monitored by other data focused IoT.

Implementation of BLYNK IoT

1. Setting up Blynk account

    - Create an account by registering it on the BLYNK website or smartphone application using a valid email.
    - Obtaining an auth token from BLYNK platform that will be used later for implementation on the smart home project.

2. Configuring the BLYNK application

    - Installing the application on smartphone device to make use of its controller for the smart home IoT.
    - Make a user interface on BLYNK like adding buttons, gauge, and much more on the smartphone app.
    - Add Datastream on BLYNK according to the smart home like Datastream for LEDs, Fan, Temperature, fire sensor, and much more.

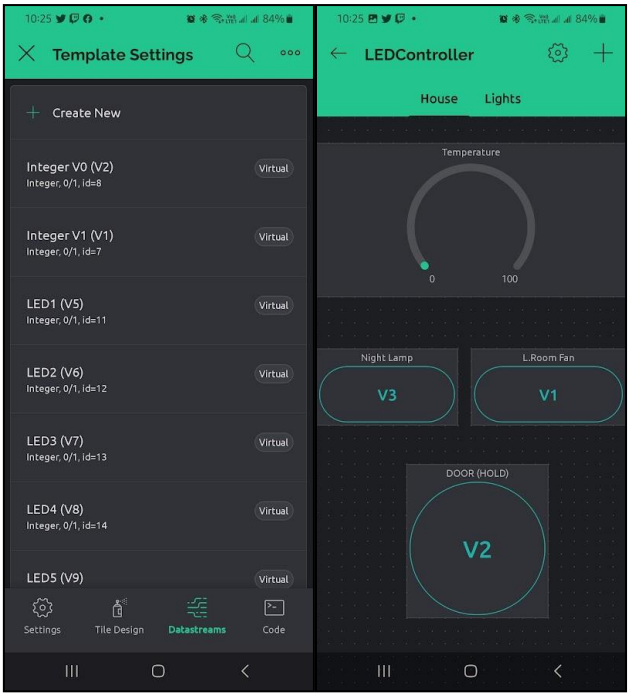


Figure 3.5: Datastream and user interface of BLYNK

3. Code development

- Coded the BLYNK connection for the Arduino Mega project which included with BLYNK library using Arduino IDE to allow BLYNK integrate with the Arduino Mega (Smart Home IoT)
- Implementing code to send data from sensors to BLYNK because this will enable controlling LEDs and other sensors based on the command received from BLYNK.

```
#include <BlynkSimpleShieldEsp8266.h>

char auth[] = "AUTHCODEHERE"; // Line 49 - 51: Login WI-FI
char ssid[] = "gamesense";
char pass[] = "12345678";


ESP8266 wifi(&Serial1); // Declare pin 18,19 arduino mega


void setup() {
    Blynk.begin(auth, wifi, ssid, pass);
    if (Blynk.connected())
    {
        tone(7,1000);
        delay(500);
        noTone(7);
    }

}

void loop()
{
    Blynk.run();
}
```

Figure 3.6: Connection code for BLYNK

```
BLYNK_WRITE(V5) { //Bedroom LED
  int pinValue = param.asInt();

  if (pinValue == 1)
  digitalWrite(bedroomLED, HIGH);
  else
  digitalWrite(bedroomLED, LOW);
}
```

Figure 3.7: Code to control LEDs via BLYNK

3.3 Results

This section discusses the results of the smart home project, where any outcome from the smart home collected to evaluate the performance of the smart home. One result shows that the successful integration of BLYNK IoT / framework platform. Implementation of BLYNK allowed end-users to remotely control every sensor available on the smart home using a smartphone. Additionally, data sent from Arduino Mega's sensor to BLYNK will have a consistent pattern which is every 5 seconds data been transmitted. This delay prevents overload of the Arduino Mega and improves the system performance by effectively data transmitted from BLYNK to Arduino Mega, enhancing the overall performance of the smart home.

Besides that, the temperature sensors where end-users can now monitor their temperature of the smart home via their smartphone. As mentioned above, it will update the temperature every 5 seconds due to the fixed value of the delay. This enabled users to monitor their smart home and can detect any abnormal events in their homes like having an abnormal temperature. Overall, this sensor will enhance user awareness in their house and able to control their living environment without any hassle.

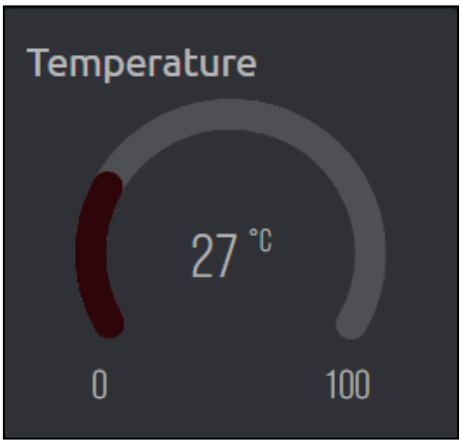


Figure 3.8: Temperate gauge from DHT11 sensor

3.3.1 Analysis

The objective of the project was to control all sensors and enable real-time monitoring of specific data in a smart home. This objective was successfully accomplished by integrating the BLYNK IoT platform. BLYNK provides a good solution for remote control of LED lights in every room of the smart home and allows monitoring of sensor data, such as temperature. The BLYNK framework includes a chart feature, where the DHT11 sensor sends data to BLYNK and generates a chart that enables end-users to monitor the temperature via their smartphones. Figure 13 shows that the temperature averaged between 25-30 degrees Celsius for a duration of 30 minutes. End-users can easily control the LED lights, turning them on or off with smartphones. Moreover, they can monitor their smart home with real-time updates from sensors like temperature and fire sensors, enabling them to detect any abnormal events or changes. The integration of BLYNK with the smart home project has successfully achieved the objective of controlling all sensors through a smartphone and providing monitoring capabilities for specific sensor outputs.

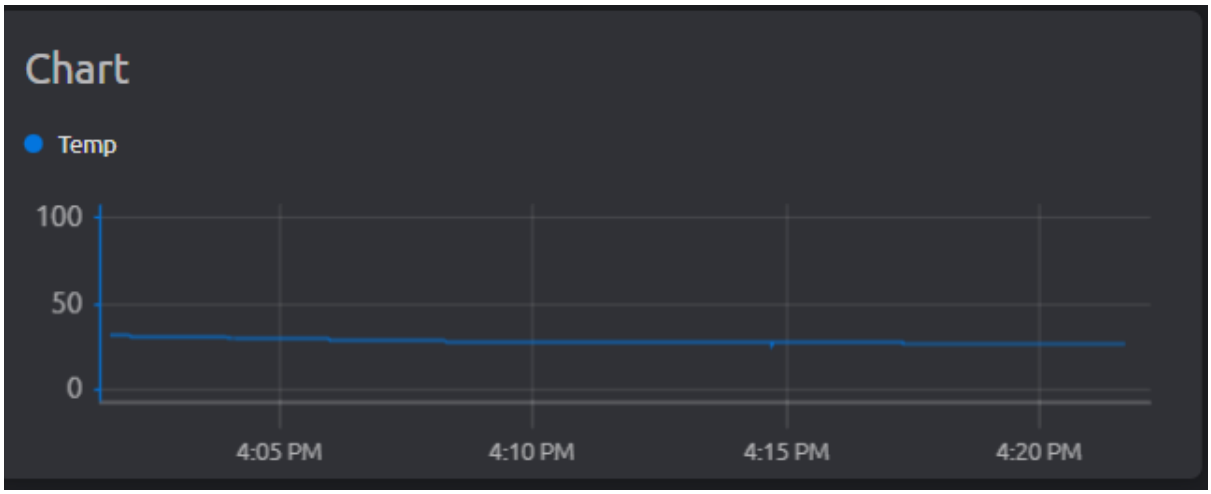


Figure 3.9: Temperature chart for DHT11 sensor

Other than that, the decision to analyse the integration of the fire sensor with the BLYNK platform in the smart home project was driven by the importance of safety. This integration has significantly enhanced the safety of the smart home. When a fire is detected, the fire sensor integrated with BLYNK IoT platform sends an emergency notification to the user's smartphone, alerting them about the detected fire. This will allow user to quickly respond such as contacting emergency number and alerting nearby neighbours. Additionally, BLYNK sends an email notification if any fire has been detected within the house shown in Figure 14. This has enhanced the overall safety of the smart home.

Figure 3.10 Screenshot of email notification for fire sensor

3.3.2 Limitations

Integrating with BLYNK API for the smart home system might a success and it achieved the objective of the project but there are certain limitations that encountered while doing this project. First of all, lack of customization and design for BLYNK IoT platform / framework. By that, a lot of features that offered by BLYNK are all behind paywall such as having limited widgets, lack of advanced features, and lack on customizable and customization for the user interface of BLYNK's application.

Furthermore, one of the limitations of the smart home system is the outdated firmware of the Wi-Fi module, ESP-01s. The factory-installed firmware provided by the manufacturer has limited connectivity and requires manual configuration of UDP and TCP, which can be time-consuming and not efficient for the module itself. Additionally, UDP and TCP is one of the required aspects to send and receive a HTTP requests. Finding an up-to-date firmware for the module is not possible due to discontinuation of the module's production by the different manufacturer. While there are workarounds such as flashing the module with a different manufacturer's firmware, there is a chance that the module would lead to corruption and rendering it unusable. As a result, outdated firmware for the ESP-01s Wi-Fi module is one of the limitations of this project.

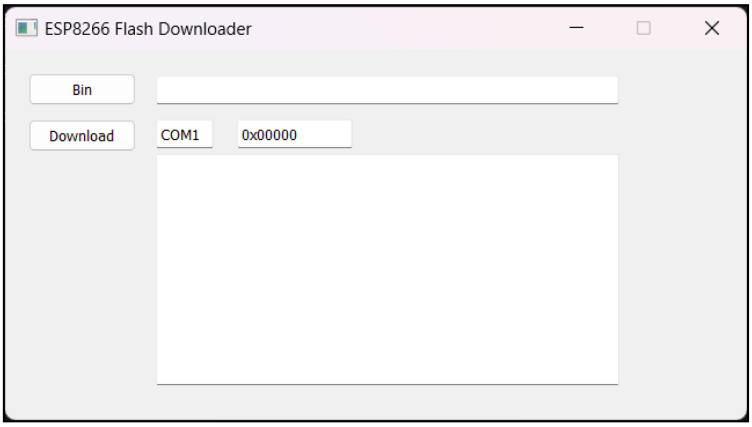


Figure 3.11: ESP8266 Flash Tool

Other than that, the use of microcontroller Arduino Mega is one of the limitations to this project. Being an older microcontroller board, the Arduino Mega has limited in terms of power and memory compared to newer microcontroller that is available to the market. This limitation can affect the overall performance of the sensors installed on the smart home. Additionally, Arduino Mega lacks simple features such as integrated Wi-Fi and Bluetooth modules where most new microcontroller come with it already. So, the use of Arduino Mega in the project is limited by its power and memory capabilities.

Lastly, the smart home project is limited by its reliance on a stable internet connectivity. The BLYNK framework are integrated to the project and it serves as an IoT platform for monitoring and controlling the sensors. However, any disruption on the internet connection will draw the smart home uncontrollable. As a result, this will impact the smart home in terms of safety and security.

In conclusion, the integration of the BLYNK API for the smart home IoT has been successful, providing end-users with control and monitoring capabilities for the sensors. However, the smart home does face several limitations which are outdated firmware of the Wi-Fi module, BLYNK's paywall, outdated Arduino microcontroller, and the requirement for a stable internet connection. These limitations should be considered when evaluating the overall functionality and usability of the smart home system.

3.3.3 Recommendations

Based on the limitation that have encountered while doing Smart Home IoT project, there are several recommendations that can improve the Smart Home in many fields such as functionality, performance, and user experience of the smart home. By implementing this recommendation, this will greatly improve quality-of-life in terms of reliability, easy access, and much more for end-users.

Firstly, upgrading the Arduino Mega into a much more up-to-date microcontroller board by replacing the Arduino Mega with a newer microcontroller board that has raw processing power, memory and comes with integrated features like WI-FI and Bluetooth connectivity. For example, ESP32 and Raspberry PI is one of the most popular microcontrollers that are being offered right now due to it having much more raw power. This will improve the performance of the smart home and provide opportunities to add more complex / advanced sensor to the smart home project.

Furthermore, using an alternative Wi-Fi module instead of ESP-01s. By that, using a Wi-Fi module that is compatible with the smart home and offer up-to-date firmware version. In addition, choosing Wi-Fi module that offers better support for HTTP requests will allow the smart home project to connect different type of APIs like Telegram, WhatsApp, and much more. Having these kinds of APIs will enhance the notification alert instead of relying on BLYNK.

Other than that, adding new APIs for the smart home project is one of the recommendations to improve the project. There are many different APIs that offer a wide range of functionality and features that fit the smart home project. For example, APIs that can enhance the quality-of-life for end users are Google Assistant API, Amazon Alexa Skills Kit, SmartThings API, Apple HomeKit API, and IFFT (If this Then That) API.

Lastly, a recommendation to improve the smart home project is to explore alternative platforms or available framework for controlling and monitoring the smart home instead of using BLYNK. There are plenty of IoT platforms or frameworks like OpenHAB, Node-RED, and Home Assistant where it can offer many more advanced features that can improve the smart home automation and connectivity. By exploring other framework or platform, it will provide more customization in many fields such as user interface, features, home automation, and much more. In addition, this will also enhance flexibility and increase quality-of-life for end-users.

Conclusion

The Smart Home IoT project has undergone significant enhancements, transforming it into a more sophisticated, automated, and intelligent smart home. The project utilized various sensors, including the RFID sensor, DHT11 sensor, photoresistor sensor, and flame sensor, which were implemented previously. Additionally, Wi-Fi connectivity, LEDs, and integration of the smart home with BLYNK IoT platform have been integrated into every sensor of the smart home. These additions have greatly improved the quality of life and functionality for end-users.

References

1. NorAinShapiqa 2023, Smart-Home-IOT-project, GitHub, viewed 22nd May 2023, https://github.com/NorAinShapiqa/Smart-Home-IOT-project.

2. Gillis, A 2021, 'What is internet of things (IoT)?', IOT Agenda, viewed 22nd May 2023, https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT.

3. SmartThings n.d., Smart Home Automation, viewed 22nd May 2023, https://www.smartthings.com/smart-home.

4. Singh, A 2019, Smart Home Automation Using IoT, Amazon Digital Services LLC - KDP Print US.

5. ResearchGate n.d., The Internet of Things: A Survey, ResearchGate, viewed 27th May 2023, https://www.researchgate.net/publication/222571757_The_Internet_of_Things_A_Survey.

6. Caragliu, A, Del Bo, C & Nijkamp, P 2011, 'Smart cities in Europe', Journal of Urban Technology, vol. 18, no. 2, pp. 65-82, doi:10.1080/10630732.2011.601117.

7. Han, S, Chen, X, Zhang, M, Liu, Y & Xu, Y 2018, 'A smart home energy management system based on internet of things and distributed computing', IEEE Transactions on Smart Grid, vol. 9, no. 1, pp. 153-164.

8. Chen, J, Zhang, C & Guo, H 2019, 'A survey on smart home management technologies and applications', Mobile Networks and Applications, vol. 24, no. 4, pp. 1116-1125.

9. Smith, J & Johnson, L 2017, 'Arduino-based home automation: An energy-efficient approach', International Journal of Smart Home, vol. 11, no. 3, pp. 85-96.

10. Johnson, A, Smith, B & Anderson, C 2018, 'Comparative analysis of microcontrollers for home automation', Journal of Smart Home