**ASSIGNMENT 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **TEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | **Unit 45: Internet of Things** | | |
| **Submission date** | 8/12/2023 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Tran Cong Hoang | **Student ID** | BH00317 |
| **Class** | IT0601 | **Assessor name** | Le Van Thuan |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** |  |

**Grading grid**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P3 | P4 | P5 | P6 | M3 | M4 | M5 | D2 |
|  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

|  |
| --- |
| Table of Contents  [**I.** **Introduction** 1](#_Toc153539788)  [**II.** **Contents** 2](#_Toc153539789)  [**P3: Investigate architecture, frameworks, tools, hardware and API techniques available to develop IoT applications.** 2](#_Toc153539790)  [**1.** **IoT architectural model** 2](#_Toc153539791)  [**2.** **Project development framework and platform** 6](#_Toc153539792)  [**3.** **Tools** 6](#_Toc153539793)  [**4.** **IoT hardware used** 9](#_Toc153539794)  [**5.** **APIs and protocols used in the project** 14](#_Toc153539795)  [**P4: Discuss a specific problem to solve using IoT.** 14](#_Toc153539796)  [**1.** **Implementation plan** 14](#_Toc153539797)  [**2.** **Problem of definition** 15](#_Toc153539798)  [**3.** **Purpose of the project** 16](#_Toc153539799)  [**4.** **Function of project** 17](#_Toc153539800)  [**P5: Employ an appropriate set of tools to develop a plan into an IoT application.** 19](#_Toc153539801)  [**1.** **Deploy the application on the development environment** 19](#_Toc153539802)  [**2.** **Build and interpret source code (using Arduino IDE )** 20](#_Toc153539803)  [**3.** **Description and explanation of the project** 32](#_Toc153539804)  [**4.** **Demo video and explanation of the operating process.** 32](#_Toc153539805)  [**P6: Create a detailed test plan and examine feedback.** 33](#_Toc153539806)  [**1.** **Test plan** 33](#_Toc153539807)  [**2.** **Feedback.** 33](#_Toc153539808)  [**3.** **Project improvement** 36](#_Toc153539809)  [**III.** **Conclusion** 37](#_Toc153539810)  [**IV.** **References** 38](#_Toc153539811) |

|  |
| --- |
| List of Figires  [Figure 1: Four-layer Architecture (Jelvix, n.d.) 2](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567456)  [Figure 2: 3-layer architectural model 3](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567457)  [Figure 3: Five-layer architectural 5](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567458)  [Figure 4: Blynk 6](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567459)  [Figure 5: Arduino IDE 7](#_Toc153567460)  [Figure 6: Blynk 8](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567461)  [Figure 7: Arduino IDE 19](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567462)  [Figure 8: Download Arduino IDE 19](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567463)  [Figure 9: Circuit diagram 20](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567464)  [Figure 10: Create a new project 21](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567465)  [Figure 11: Add library for board esp 8266 21](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567466)  [Figure 12: Install library Blynk 22](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567467)  [Figure 13: Install Library esp 8266 22](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567468)  [Figure 14: Declare pins and value 23](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567469)  [Figure 15: Include libraries and Configure Blynk 23](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567470)  [Figure 16: Get value from Blynk 23](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567471)  [Figure 17: Check connection and run 24](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567472)  [Figure 18: Setup Pin 24](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567473)  [Figure 19: Car forward 24](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567474)  [Figure 20: Car put off fire 25](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567475)  [Figure 21: Car Stop 25](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567476)  [Figure 22: Car turnleft 25](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567477)  [Figure 23: Car turnright 25](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567478)  [Figure 24: Car backward 25](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567479)  [Figure 25: Car Smart 26](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567480)  [Figure 26: Car Auto 26](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567481)  [Figure 27: Set up datastream 28](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567482)  [Figure 28: Set devices 28](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567483)  [Figure 29: Set virtual Pin datastream 31](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567484)  [Figure 30: Set virtual Pin datastream 31](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567485)  [Figure 31: Feedback 1 34](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567486)  [Figure 32: Feedback 2 34](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567487)  [Figure 33: Feedback 3 35](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567488)  [Figure 34: Feedback 4 35](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_ASM2_2nd.docx#_Toc153567489)  [Figure 35: Feedback 5 36](#_Toc153567490) |

|  |
| --- |
| List of Tables  [Table 1: Hardware 14](#_Toc153570029)  [Table 2: Blynk App 29](#_Toc153570030)  [Table 3: Test plan 33](#_Toc153570031) |

# **Introduction**

Fire incidents pose significant threats to life, property, and the environment. Timely and effective response is crucial to minimize the damage caused by fires. However, traditional firefighting vehicles often face challenges in accessing densely populated areas with narrow streets and heavy traffic. To address this issue, the integration of Internet of Things (IoT) technologies into firefighting vehicles has emerged as a promising solution. This report presents a comprehensive analysis and development plan for a smart firefighting vehicle that leverages IoT capabilities. The objective of this project is to design and implement an innovative solution that enhances the efficiency, mobility, and firefighting capabilities of traditional firefighting vehicles.

The project focuses on investigating the architecture, frameworks, tools, hardware, and API techniques available for developing IoT applications. By utilizing these technologies, the smart firefighting vehicle will possess advanced features such as real-time fire detection, autonomous navigation, and automated fire suppression. These capabilities will enable the vehicle to reach fire incidents in densely populated areas more swiftly and effectively. In addition to addressing the problem of limited accessibility, the project aims to solve specific challenges associated with firefighting operations. By integrating IoT sensors and actuators, the smart firefighting vehicle can gather and analyze environmental data, monitor fire behavior, and make intelligent decisions to optimize firefighting strategies. This will result in improved safety for firefighters and increased effectiveness in extinguishing fires.

To develop the IoT application for the smart firefighting vehicle, a careful selection of tools and technologies will be made based on their suitability for the project requirements. The project will employ an appropriate set of tools, including integrated development environments (IDEs), programming languages, IoT platforms, hardware development kits, simulation and testing tools, and version control systems. A detailed test plan will be created to ensure the quality and reliability of the IoT application. The plan will encompass various test scenarios and test cases to validate the functionalities, performance, and robustness of the smart firefighting vehicle. Feedback from users, stakeholders, and beta testers will be gathered and analyzed to further improve the application's usability and effectiveness.

In conclusion, this report presents an overview of the smart firefighting vehicle project, highlighting the significance of integrating IoT technologies into firefighting operations. The subsequent sections delve into the investigation of architecture, frameworks, tools, hardware, and API techniques, as well as the development plan, test plan, and feedback analysis. The project aims to revolutionize the firefighting industry by providing an advanced solution that enhances the accessibility, efficiency, and safety of firefighting operations through the power of IoT.

# **Contents**

## **P3: Investigate architecture, frameworks, tools, hardware and API techniques available to develop IoT applications.**

### **IoT architectural model**

#### **4-layer architectural model**

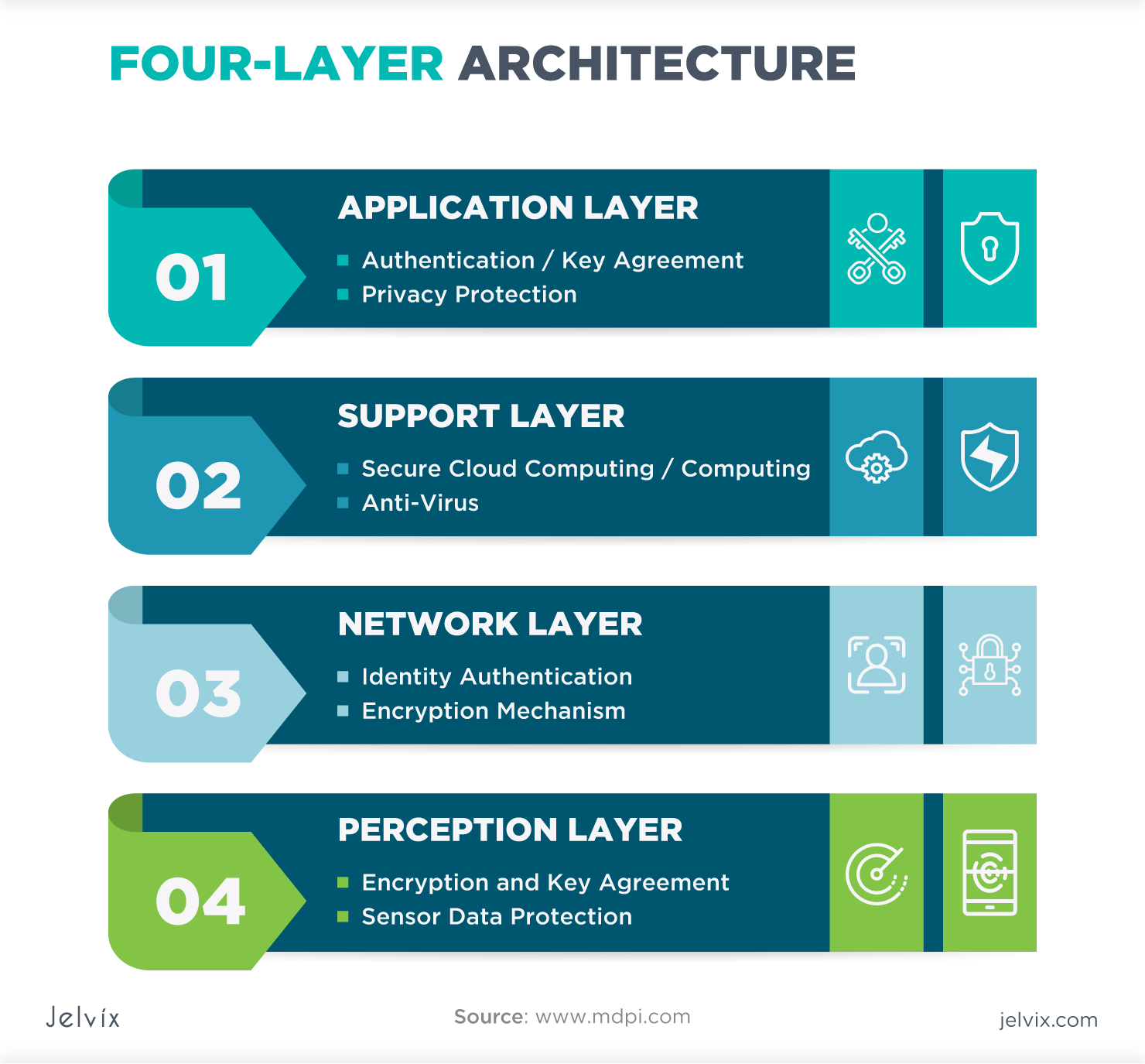
IoT stands for Internet of Things, which is an ecosystem of devices that are connected to the internet and can communicate and cooperate with each other. The IoT architectural model is the structure that allows these devices to communicate with each other. There are many different IoT architecture models, but a popular model is the 4-layer model, which includes the following layers (Jena, 2020):

Figure : Four-layer Architecture (Jelvix, n.d.)

* **Sensor layer**: This layer includes sensors and motors placed in the environment to collect information about temperature, humidity, light, sound and other physical parameters. These devices are connected to the network layer through wired or wireless communication protocols.
* **Network layer**: This layer includes the protocols and technologies that allow devices to connect and communicate with each other and the wider internet. Examples of network technologies commonly used in IoT are WiFi, Bluetooth, Zigbee, and cellular networks such as 4G and 5G. Additionally, the network layer may also include gateways and routers that mediate between devices and the broader internet, and may also include security features such as encryption and authentication to protect against unauthorized access. unauthorized access.
* **Data processing layer**: This layer includes software and hardware components responsible for collecting, analyzing and interpreting data from IoT devices. This layer is responsible for receiving raw data from devices, processing it and making it available for further analysis or action. The data processing layer includes many technologies and tools, such as data management systems, analytics platforms, and machine learning algorithms. These tools are used to extract meaningful information from data and make decisions based on that data.
* **Application layer**: This layer is the highest layer that interacts directly with the end user. It is responsible for providing user-friendly interfaces and functions that allow users to access and control IoT devices. This layer includes various software and applications, such as mobile applications, web portals, and other user interfaces designed to interact with the IoT infrastructure.

#### **3-layer architectural model**

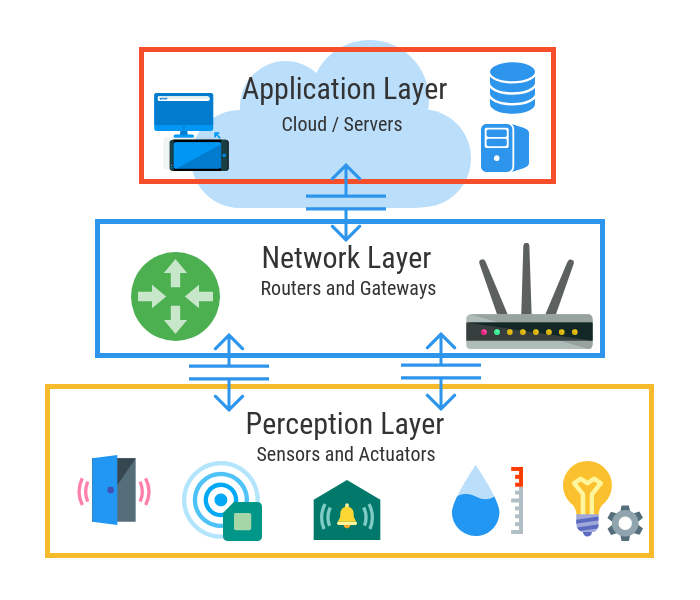
The 3-layer model is a simplified model from the 4-layer model by combining the data processing layer and application layer into a single layer1. This model is suitable for simple and small IoT applications that do not require many data processing and analysis features1. This model has advantages such as ease of design, deployment and management, cost and resource savings, and high scalability2. However, this model also has disadvantages such as not ensuring security, reliability and high performance, not supporting complex and diverse applications, and having the risk of becoming a bottleneck in the system, it includes the following layer:

Figure : 3-layer architectural model

* + **Sensor layer**: This layer contains devices such as sensors and motors to collect data from the environment. These devices can use wired or wireless communication protocols to connect to the network layer. These devices may also have security features such as encryption and authentication to protect data from unauthorized access.
  + **Network layer**: This layer contains technologies and protocols for connecting and transferring data between devices and the internet. Network technologies commonly used in IoT are WiFi, Bluetooth, Zigbee, and mobile networks such as 4G and 5G. The network layer can also include gateways and routers to mediate between devices and the broader internet (IOTLINK, 2023).
  + **Processing/application layer**: This layer contains tools and applications to process, analyze and display data to users. This layer can use data management systems, analytics platforms, and machine learning algorithms to extract meaningful information from data and make decisions based on that data. This layer also provides user-friendly interfaces and functions to access and control IoT devices, such as mobile applications, web portals, and other user interfaces (Thietbimang.com, n.d.).

#### **5-layer architectural model**

This model is a more complex model, including 5 layers: sensor layer, network layer, data processing layer, presentation layer and application layer. The sensor layer and network layer are similar to the 3-layer model. The data processing layer is a new layer, dedicated to processing, analyzing, and storing data from IoT devices. The presentation layer is a new layer, specialized for representing data in different formats, like charts, tables, images, etc. The application layer is a new layer, dedicated to providing user-friendly interfaces and functions to access and control IoT devices. This model is suitable for large and complex IoT applications that need a lot of data visualization and interaction features. This model has the advantage of ensuring security, reliability and high performance, supporting diverse and rich applications, and taking advantage of the power of cloud and edge computing. However, this model also has disadvantages: it is difficult to design, deploy and manage, consumes a lot of costs and resources, and can encounter compatibility and integration problems. The 5-layer model is a complex IoT architecture model, including the following 5 layers:

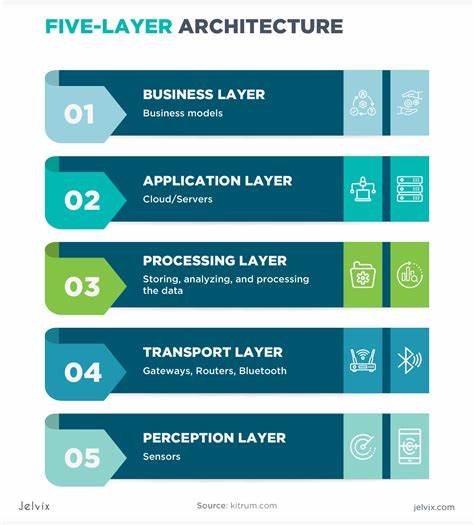
* + **Sensor layer**: This layer contains devices such as sensors and motors to collect data from the environment. These devices can use wired or wireless communication protocols to connect to the network layer. These devices may also have security features such as encryption and authentication to protect data from unauthorized access.
  + **Network layer**: This layer contains technologies and protocols for connecting and transferring data between devices and the internet. Network technologies commonly used in IoT are WiFi, Bluetooth, Zigbee, and mobile networks such as 4G and 5G. The network layer can also include gateways and routers to mediate between devices and the broader internet.
  + **Data processing layer**: This layer is specialized for processing, analyzing and storing data from IoT devices. This layer can use data management systems, analytics platforms, and machine learning algorithms to extract meaningful information from data and make decisions based on that data⁵. This layer can also leverage the power of cloud and edge computing to process data close to the source.

Figure : Five-layer architectural

* + **Presentation layer**: This layer is specialized for representing data in different formats, such as charts, tables, images, etc.⁷. This class helps users have an intuitive and easy-to-understand view of the data. This layer can also provide data manipulation and interaction features.
  + **Application layer**: This layer provides user-friendly interfaces and functions to access and control IoT devices. This layer includes various software and applications, such as mobile applications, web portals, and other user interfaces. This layer can also provide value-added services to users, such as notifications, recommendations, and support.

#### **Selected architectural model**

In my project, I apply a three-layer model including:

* **Sensor Layer**: Includes fire sensor and infrared sensor to detect and monitor fire conditions. The fire sensor sends data about the fire status to the data processing layer, and the infrared sensor detects the line to control the vehicle's direction.
* **Network and Data Processing Layer**: Use the ESP8266 to connect to the Wi-Fi network and transmit data from the sensor to the Blynk application on the phone. The ESP8266 serves as a bridge between the sensor and the Blynk application.
* **Application and Service Layer**: Use the Blynk application on your phone to control components and send signals to fire engines. Blynk provides a user interface for interacting with the fire truck model and displaying fire status.

### **Project development framework and platform**

* **Blynk**: Blynk is an IoT platform that allows you to build IoT applications that interact with devices through mobile phones. In my project, I use the Blynk app on my phone to control the fire truck model and monitor information from the sensors.
* **ESP 8266**: ESP 8266 is a compact and powerful Wi-Fi microcontroller. In my project, the ESP 8266 is used to connect the fire truck model to the Wi-Fi network and transmit data from the sensor to the Blynk application.

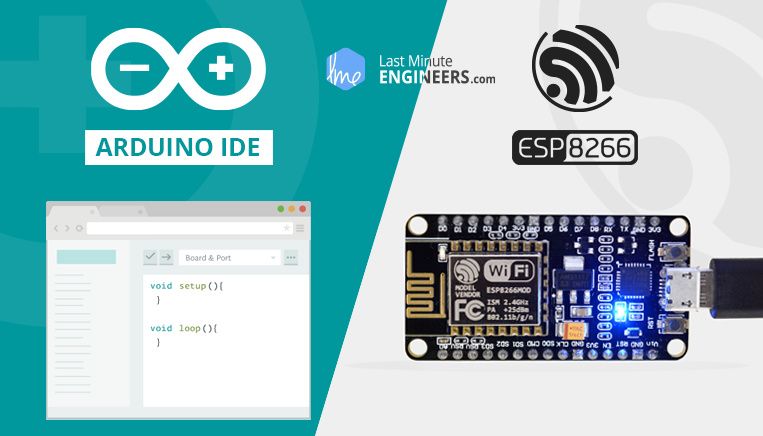
Figure : Blynk

### **Tools**

#### **Arduino IDE**

The Arduino Integrated Development Environment (IDE) is an open-source software application that allows you to write, upload, and manage code on Arduino-compatible boards. Arduino is a popular open-source electronics platform based on easy-to-use hardware and software.

The core functionality of the Arduino IDE resides in its code editor, providing a workspace for crafting programs using the C and C++ programming languages. This simplicity is particularly advantageous for beginners, allowing them to explore the realm of electronics without grappling with complex programming languages. Additionally, the IDE integrates a compiler responsible for translating the written code into machine-readable instructions, enabling seamless execution of tasks on the Arduino microcontroller.

Figure : Arduino IDE

A standout feature of the Arduino IDE in the context of smart parking development is its adept handling of libraries. The built-in Library Manager empowers users to effortlessly add, remove, and update libraries that extend the functionalities of their Arduino-based smart parking projects. These libraries encapsulate pre-written code snippets, streamlining the integration of existing solutions and saving developers valuable time and effort. The IDE's Board Manager ensures that the compiled code aligns with the specific microcontroller present on the selected Arduino board, a crucial aspect in the development of smart parking systems.

As developers advance in their smart parking projects, the Arduino IDE offers a powerful Serial Monitor tool. This feature facilitates communication between the computer and the Arduino board, enabling developers to monitor data exchange over the serial port. Whether debugging code or observing real-time sensor readings crucial for smart parking functionality, the Serial Monitor proves to be an invaluable tool.

In summary, the Arduino IDE serves as the central hub for programming and developing smart parking projects, fostering a community of innovators and makers committed to pushing the boundaries of what's achievable with microcontrollers and electronics in the realm of modern parking solutions.

#### **Blynk**

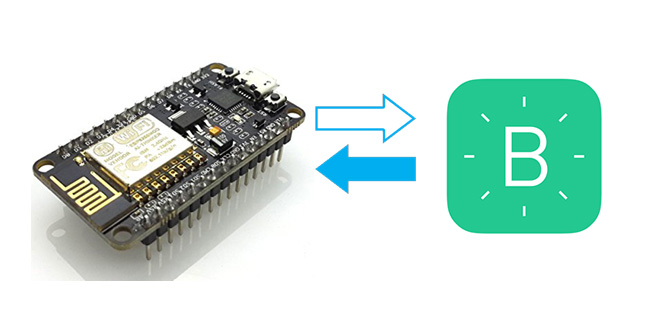
Blynk is a versatile platform designed for developing Internet of Things (IoT) applications. At its core, Blynk offers a straightforward method for connecting and managing devices and projects via a mobile application. The platform empowers users to construct customized dashboards containing various widgets that facilitate the remote control of connected hardware.In the realm of smart parking solutions, the primary avenue for interacting with Blynk is through a versatile mobile application, compatible with both iOS and Android operating systems. This app serves as a centralized hub, empowering users to craft customized graphical interfaces or dashboards tailored specifically to their IoT-driven smart parking projects. Within these dashboards, users can seamlessly integrate an array of widgets, including buttons, sliders, graphs, and displays, fostering intuitive interaction with connected devices.

Figure : Blynk

Blynk's versatility spans a diverse array of hardware platforms, including popular choices such as Arduino, Raspberry Pi, ESP8266, and ESP32. Integration involves users installing the Blynk library on their hardware, configuring it to establish seamless communication with the Blynk server. The communication channel between the mobile app and connected hardware operates through Blynk's straightforward and efficient protocol, enabling users to exchange data, send/receive commands, monitor sensor data, and control actuators effortlessly in the context of IoT-driven smart parking projects.

Blynk projects encapsulate both the mobile app configuration (dashboard layout, widgets, etc.) and the hardware code. Users initiate a project within the Blynk app, utilizing the provided authentication token to establish a secure connection with their hardware. Essentially, Blynk streamlines the prototyping process and remote control of hardware, making it a preferred choice among makers, hobbyists, and IoT enthusiasts in the context of smart parking solutions. Its user-friendly approach liberates individuals to concentrate on the creative aspects of their projects, eliminating the need to develop a dedicated mobile app for each endeavor

### **IoT hardware used**

* + **Fire sensor and infrared sensor**: Fire sensor is used to detect fire, while infrared sensor is used to detect and control fire truck model movement.
  + **Servo**: Servo is used to control the direction of movement of the fire extinguisher on the vehicle model. The ESP8266 sends a control signal to the servo to rotate the fire arm in the desired direction.
  + **5V Relay Module**: The 5V Relay Module is used to control other devices in the fire truck model, such as the water pump. It acts as an electrical switch to turn on/off power to those devices.
  + **Pump**: Pump is used to provide water for firefighting. When needed, the ESP8266 will activate the pump through the relay module to pump water out.
  + **DC and L298N gear reduction motors**: DC and L298N gear reduction motors are used to control the movement of the fire truck in your model. The ESP8266 will send control signals to the L298N to adjust the DC deceleration motor, thereby controlling the movement of the fire truck.

The following is all the hardware used in the smart fire truck project:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N.O** | **Hardware** | **Image** | **Quantity** | **Price** |
| **1** | ESP 8266 | A black circuit board with a white background  Description automatically generated | 1 | 2$ |
| **2** | Esp 8266 foot base | A black circuit board with many small pins  Description automatically generated | 1 | 2$ |
| **3** | Flamer Sensor | A blue electronic device with a black cap  Description automatically generated | 1 | 0.4$ |
| **4** | 4-wheeled robot chassis + 4 DC motors | A plastic and plastic model of a car  Description automatically generated with medium confidence | 1 | 6$ |
| **5** | Jumper ware | A bunch of colorful wires  Description automatically generated | 40 | 2$ |
| **6** | IR sensor | A close-up of a blue circuit board  Description automatically generated | 2 | 1$ |
| **7** | Cap USB micro | A blue usb cable with a white background  Description automatically generated | 1 | 0.4$ |
| **8** | Pump | A small white pump with black wire  Description automatically generated | 1 | 0.5$ |
| **9** | 3.7V battery | A group of blue batteries  Description automatically generated | 4 | 4.5$ |
| **10** | Battery base box | A battery with two blue batteries  Description automatically generated | 2 | 1$ |
| **11** | TiO ware | A close-up of a tube  Description automatically generated | 1 | 0.3$ |
| **12** | Breakboard | A white and blue circuit board  Description automatically generated | 1 | 0.5$ |
| **13** | Module Relay 5V | A blue electronic device with red and green lights  Description automatically generated | 1 | 0.6$ |
|  |  | **Total** |  | **23$** |

Table : Hardware

### **APIs and protocols used in the project**

* + **Blynk API**: Blynk provides APIs to interact with IoT components and devices from mobile applications. This API allows controlling components and receiving information from sensors in the Blynk application on mobile phones.
  + **Wi-Fi:** In my project, a Wi-Fi connection is used to connect the ESP8266 to the network and the Blynk application on the phone. ESP8266 uses Wi-Fi connection to transmit data from the sensor to the Blynk application and receive control commands from the Blynk application.

## **P4: Discuss a specific problem to solve using IoT.**

### **Implementation plan**

**Step 1: Research and Development**: Conduct a thorough analysis of existing IoT technologies, firefighting equipment, and vehicle design principles. Identify and select appropriate components and systems for IoT firefighting vehicles.

**Step 2: Vehicle design and manufacturing**: Develop a specialized vehicle design that incorporates selected IoT technologies. Build the vehicle with the compartments, equipment and sensors needed for firefighting operations in tight, densely populated areas.

**Step 3: IoT integration**: Integrate fire detection sensors, tracking positioning systems, fire warning systems and Wi-Fi connections into the vehicle's control system. Ensures seamless communication and data exchange between the vehicle and external devices.

**Step 4: Mobile Application Development**: Design and develop user-friendly mobile applications using the Blynk platform. This application will enable remote control of IoT firefighting vehicles, real-time monitoring of fire incidents and data analysis.

**Step 5: Testing and Evaluation**: Conduct rigorous testing of IoT firefighting vehicles to ensure their functionality, reliability, and compliance with safety standards. Evaluate its performance in simulated firefighting situations and make necessary adjustments.

**Step 6: Deployment and training**: Deploy IoT firefighting vehicles in targeted densely populated areas. Provide training to users, firefighters, and emergency personnel on how to effectively operate and utilize the vehicle's IoT capabilities.

Developing an IoT firefighting vehicle specifically designed for densely populated areas is critical to address the challenges that traditional firefighting vehicles face in such environments. By combining advanced technologies, effective positioning systems, and real-time monitoring capabilities, IoT firefighting vehicles aim to enhance response times, improve firefighting operations, and ultimately Protect the lives and property of residents in densely populated areas. Successful implementation of this project will contribute to the overall safety and resilience of the community in the face of fire incidents.

### **Problem of definition**

#### **Problem**

According to a report from the Ministry of Public Security of Vietnam, in the first 4 months of 2023, there were 1,512 fire and explosion incidents nationwide, killing 56 people, injuring 132 people and causing damage estimated at more than 1,000 billion VND (Anon, 2023). Crowded but narrow residential areas in Vietnam are one of the places with a high risk of fire and explosion due to causes such as electricity, gas, chemicals, etc. However, it is difficult for normal fire engines to reach these areas due to narrow streets, heavy traffic, etc. This can lead to serious consequences for people's lives and property. Some examples of fires and explosions in crowded but narrow residential areas in Vietnam in 2023 include a 9-story apartment fire in Hanoi on September 13, 2023, killing 56 people and 37 injured (Statista, n.d.); the Gas explosion in Ward 12, Binh Thanh District, Ho Chi Minh City on October 22, 2023, killed 3 people and injured 5 people (Reuters, 2023).

#### **Target users**

Households in crowded but narrow residential areas in Vietnam, especially large cities such as Hanoi, Ho Chi Minh, Da Nang, etc. are people with high demand for a solution to prevent and promptly handle fires and explosions. According to statistics from the General Statistics Office of Vietnam, by 2020, the country will have 9.8 million households in crowded but narrow residential areas, accounting for 35.8% of the total number of households. In particular, large cities such as Hanoi, Ho Chi Minh, and Da Nang have a higher proportion of households in crowded but narrow residential areas than the national average, 48.7 percent respectively, and 60.4 percent and 41.8 percent (Tatarski, 2023).

#### **Needs**

Households need a solution to prevent and promptly handle fires and explosions, protecting themselves and the community. Households want a device that can detect fire sources early and promptly warn people and authorities, and can quickly and effectively put out fires using water or other substances. , can move flexibly and intelligently along lines or according to signals from the user, can connect to the Internet to send and receive data about the status of the device and the location of the fire source.

### **Purpose of the project**

#### **Objective**

Develop an IoT application for smart fire trucks that can detect fire, move along lines, extinguish fires and warn of fires automatically and effectively. This application will use technologies such as sensors, motors, water pumps, esp 8266 boards, WiFi, etc. to collect, transmit and process data from IoT devices and control the responding devices. The application will also have a user interface on a smartphone or computer so that users can remotely monitor and control the smart fire engine.

#### **Benefits**

IoT applications for smart fire trucks can help households:

* Minimize the risk of fire and explosion by early detection of fire sources and timely warnings to people and authorities. According to a study by the Vietnam Institute of Science and Technology, more than 80% of fires and explosions can be prevented if detected and handled within the first 3 minutes (Anon, 2023). IoT applications for smart fire trucks can reduce response time and increase warning effectiveness by using sensitive fire sensors speakers and LED lights to emit sound and light fire warnings. The app can also send notifications to users and authorities via WiFi and user interface, including the location and intensity of the fire source.
* Minimize loss of life and property by quickly and effectively extinguishing fires using water pumps and moving along lines. According to a study by Hanoi University of Science and Technology, water is one of the most effective fire extinguishers for most types of fires, such as fires caused by electricity, gas, chemicals, etc. Water can cool the temperature of the fire source, reduce the oxygen supply to the fire, and wet flammable materials (Statista, n.d.). IoT applications for smart fire trucks can use a water pump to spray water on the fire source and a relay module to turn the water pump on and off due to signals from the ESP 8266 board. This application can also move along the road. White lines on a black background to find and approach the fire source automatically and flexibly using two infrared sensors to read the value from the line and send signals to the ESP 8266 board to control four motors DC and four wheels adopt L298N motor.
* Enhance access and response to fires in hard-to-reach areas using compact, flexible, and smart fire engines. According to a study by the Institute of High Technology Research and Development, smart fire trucks can reach areas that are difficult for normal fire trucks to reach, such as narrow alleys and high floors of buildings apartments, areas with many obstacles, etc (Bulletin, 2023). Smart fire trucks have a compact size and can move flexibly. The smart fire truck can also be controlled remotely using a user interface on a smartphone or computer, allowing users to easily and conveniently monitor and control the smart fire truck.

### **Function of project**

Below are some main functions of the project:

* **Fire detection**: Use the fire sensor to measure the intensity of infrared light from the fire source and send a signal to the Esp 8266 board. The fire sensor is a type of photoelectric sensor, which can detect pink light. Infrared wavelengths range from 760 nm to 1100 nm, corresponding to the emission spectrum of fire (Roberto Garcia-Martin, 2019). The fire sensor can be connected to the ESP 8266 board via analog or digital pins, depending on the desired sensitivity and accuracy. The ESP 8266 board is a WiFi module with integrated microcontroller, programmable in Arduino language (Pisonero, 2023). The ESP 8266 board can receive signals from the fire sensor and process the data to make appropriate decisions.
* **Move along the line**: Use two infrared sensors to read the value from the white line on the black background and send signals to the ESP 8266 board to control four DC motors and four wheels via the L298N motor . Infrared sensor is a type of photoelectric sensor, which can detect infrared light with wavelengths from 700 nm to 1400 nm. The infrared sensor can be connected to the ESP 8266 board via digital pins, and can output high or low values depending on the color of the reflecting surface. A DC motor is a type of motor that converts electrical energy into mechanical energy, and can control the speed and direction of rotation by changing the voltage and polarity (ieeexplore.ieee.org, n.d.). The DC motor can be connected to the esp 8266 board via the L298N motor, which is a motor control module that can power and control two DC motors at the same time.
* **Extinguishing fire**: Use a water pump to spray water onto the fire source and a relay module to turn the water pump on and off due to signals from the ESP 8266 board. The water pump is a mechanical device that can convert mechanical energy into hydraulic energy, and can create pressure and water flow to spray out. The water pump can be connected to the ESP 8266 board through the relay module, which is an electronic device that can use a small voltage signal to control a large voltage circuit (Pisonero, 2023).
* **Fire warning**: Uses a speaker to emit a fire warning sound and an LED to emit a fire warning light due to a signal from the esp 8266 board. The speaker is an audio device, which can be switched electrical signals into sound waves, which can produce different sounds depending on the frequency and amplitude of the electrical signal. The speaker can be connected to the ESP 8266 board via the PWM pin, which is a binary pulse width control technique to generate electrical signals of different widths and periods. LED light is a photoelectric device, which can convert electrical signals into light, which can produce different colors depending on the wavelength of light. The LED can be connected to the ESP 8266 board via digital pins and can output high or low values depending on the on/off status of the LED.
* **WiFi connection**: Use the esp 8266 board to connect to the WiFi network via the Blynk application and send data about the status of the smart fire truck via the Serial port. WiFi is a wireless technology that can transmit and receive data via radio waves and can connect IoT devices to the Internet. Blynk is an IoT platform that can create user interfaces on smartphones or computers to control IoT devices over WiFi. Serial is a communication protocol that can transmit and receive data in bit form and can connect IoT devices to computers.

## **P5: Employ an appropriate set of tools to develop a plan into an IoT application.**

### **Deploy the application on the development environment**

#### **Identify the development environment (Arduino IDE).**

**Arduino IDE**: I use Arduino IDE as the development environment for my project. Arduino IDE provides a simple programming interface and source code support for the ESP8266 board and other components in your project.

Figure : Arduino IDE

#### **Install and configure the development environment.**

A screenshot of a computer

Description automatically generatedArduino IDE: Download Arduino IDE from the official Arduino website and install it on the computer. After installation, I configured the development environment by selecting the board type "NodeMCU 1.0 (ESP-12E Module)" in the "Board Manager" section of the Arduino IDE. You also need to install the ESP8266 library for Arduino IDE to support programming the ESP8266 board.

Figure : Download Arduino IDE

### **Build and interpret source code (using Arduino IDE )**

#### **Circuit diagram**

Figure : Circuit diagram

#### **Create the necessary files and libraries in the development environment.**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedBelow are the steps for creating the files and libraries used in the project, as well as choosing the board:

Figure : Create a new project

Figure : Add library for board esp 8266

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure : Install library Blynk

Figure : Install Library esp 8266

#### **Build source code for IoT applications.**

Step 1: Configure the connection to Blynk and declare the necessary libraries

A screenshot of a computer program

Description automatically generatedStepA screenshot of a computer program

Description automatically generated 2: Declare pins and values received from Blynk

Figure : Declare pins and value

Figure : Include libraries and Configure Blynk

A white background with black text

Description automatically generatedStep 3: Read value from Blynk

Figure : Get value from Blynk

A screenshot of a computer

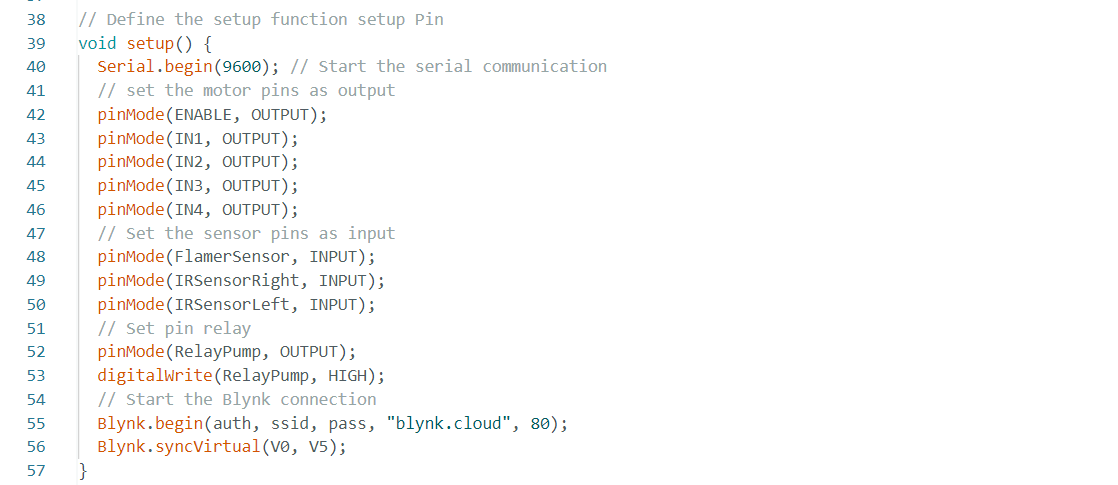
Description automatically generatedStep 4: Setup Pin

Figure : Check connection and run

Figure : Setup Pin

Step 5: Check the connection to wifi and Blynk.

Step 6: Initialize necessary functions.

A white background with black dots

Description automatically generated

Figure : Car forward

A computer screen shot of a computer screen

Description automatically generatedA white background with black text

Description automatically generatedA white background with black and white clouds

Description automatically generatedA close-up of a computer screen

Description automatically generatedA white background with a black cross

Description automatically generated

Figure : Car put off fire

Figure : Car Stop

Figure : Car turnleft

Figure : Car turnright

Figure : Car backward

A screenshot of a computer

Description automatically generated A screen shot of a computer code

Description automatically generated

Figure : Car Smart

Figure : Car Auto

#### **Explain the important parts of the source code and how they work.**

Important parts of this source code include configuring the connection to Blynk, handling events from virtual pins in Blynk, and controlling vehicle movement through the smartcar(), carforward(), carbackward functions. (), carturnleft(), carturnright() and carStop(), as well as the car's automatic mode via the carAuto() function.

* **Blynk Configuration**: Đoạn mã này chứa thông tin cấu hình kết nối với Blynk IoT platform. Các thông số bao gồm BLYNK\_TEMPLATE\_ID, BLYNK\_TEMPLATE\_NAME, BLYNK\_AUTH\_TOKEN, BLYNK\_PRINT, auth, ssid, và pass. Chúng được sử dụng để thiết lập kết nối WiFi với Blynk và xác thực thiết bị.
* **Setup Function**: Trong hàm setup(), các chức năng cần thiết cho việc khởi tạo và cấu hình các chân của vi mạch ESP8266 được thực hiện. Điều này bao gồm thiết lập chế độ của các chân như đầu ra (OUTPUT) hoặc đầu vào (INPUT) thông qua pinMode(). Ngoài ra, kết nối với Blynk cũng được khởi động thông qua Blynk.begin(auth, ssid, pass, "blynk.cloud", 80).
* **Blynk Virtual Pin Handlers**: Các hàm BLYNK\_WRITE(V0) và BLYNK\_WRITE(V5) được sử dụng để xử lý sự kiện khi giá trị của các virtual pin trong Blynk thay đổi. Đầu vào param chứa giá trị của virtual pin và chúng được sử dụng để cập nhật các biến direction và Speed tương ứng.
* **Loop Function**: Trong hàm loop(), Blynk.run() được gọi để thực thi các tác vụ liên quan đến Blynk. Sau đó, hàm smartcar(direction, Speed) được gọi để điều khiển chiếc xe thông qua các thông số direction và Speed.
* **Smartcar Function**: Hàm smartcar(direction, speed) nhận vào các thông số direction và speed để điều khiển chiếc xe theo hướng và tốc độ tương ứng. Switch-case được sử dụng để xác định hướng di chuyển của xe dựa trên giá trị của direction. Theo từng trường hợp, các hàm điều khiển chi tiết như carforward(), carbackward(), carturnleft(), carturnright() và carStop() được gọi để thực hiện các hành động tương ứng.
* **Control Functions**: Các hàm carforward(), carbackward(), carturnleft(), carturnright(), và carStop() được sử dụng để điều khiển động cơ và hướng di chuyển của xe. Chúng thực hiện việc thiết lập các chân đầu ra (IN1, IN2, IN3, IN4) theo cấu hình tương ứng để điều khiển động cơ.
* **carAuto Function**: Hàm carAuto() được sử dụng để chế độ tự động của xe. Nó kiểm tra trạng thái của các cảm biến như FlamerSensor và IRSensorRight, IRSensorLeft để xác định hành động của xe. Nếu phát hiện cháy (FlamerSensor == LOW), xe sẽ dừng lại và kích hoạt cơ chế dập lửa thông qua hàm putOffFire(). Nếu không có cháy, xe sẽ tiếp tục di chuyển theo đường đi dựa trên trạng thái của các cảm biến hồng ngoại.
* **putOffFire Function**: Hàm putOffFire() được sử dụng để tắt lửa bằng cách tắt động cơ (carStop()) và kíchhoạt bơm nước (digitalWrite(pump, LOW)). Điều này đảm bảo rằng khi xe phát hiện cháy, nó sẽ ngừng di chuyển và tự động kích hoạt quá trình dập lửa.

#### A screenshot of a computer Description automatically generated**Set up Blynk.**A screenshot of a computer Description automatically generated

Figure : Set up datastream

Figure : Set devices

Set up in App Blynk on SmartPhone:

Table : Blynk App

|  |  |  |
| --- | --- | --- |
| Không có mô tả. | Không có mô tả. | Không có mô tả. |
| **Control Speed** | **Car Forward** | **Car Backward** |
| Không có mô tả. | Không có mô tả. | Không có mô tả. |
| **Car TurnLeft** | **Car TurnRight** | **Car Auto mode** |
| Không có mô tả. | | |
| **Application Control Interface** | | |

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure : Set virtual Pin datastream

Figure : Set virtual Pin datastream

### **Description and explanation of the project**

#### **Describe project**

Your smart fire truck project uses the ESP8266 board as the main board to control the entire system. This fire truck is designed to automatically detect and handle fire incidents effectively. It will move to the fire location and provide extinguishing agent to extinguish the fire.

#### **Project explanation**

* **ESP8266 board**: The ESP8266 board is used as the main board to control the entire system. It has the ability to connect to Wi-Fi and communicate with other hardware components in the project. This board also runs programming programs to control the functions of the fire engine.
* **DC motor and wheels**: The fire truck is equipped with four DC motors and four wheels. The DC motor helps the vehicle move and the wheels help it move smoothly over the surface.
* **L298N Motor**: L298N Motor is a motor control module used to control DC motors. It receives commands from the ESP8266 board and controls the motor to move in the desired direction and speed.
* **Infrared sensors**: Two infrared sensors are used to let the vehicle move along the line. They help the vehicle detect and track the line to maintain the correct direction of travel.
* **Fire Sensor**: Fire sensor is used to detect fire in the surrounding area. When a fire is detected, it sends a signal to the ESP8266 board to activate fire incident handling measures.
* **Pump and relay module**: Pump is used to supply water or fire extinguishing agent when a fire occurs. The relay module is used to control the pump on/off, enabled by the ESP8266 board.
* **Wi-Fi connectivity through Blynk**: Blynk is an IoT platform that allows you to remotely connect, control, and monitor IoT devices. Wi-Fi connectivity through Blynk allows you to control and monitor your fire truck remotely via your mobile phone or computer.

### **Demo video and explanation of the operating process.**

Link video demo on Youtue:

## **P6: Create a detailed test plan and examine feedback.**

### **Test plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case N.O** | **Test Description** | **Acceptance criteria** | **Status** |
| **1** | Check the Wi-Fi connection between the esp 8266 board and the Blynk | Connection successful | Completed |
| **2** | Check the pump on/off control feature through the relay module | Works correctly | Completed |
| **3** | Test fire detection with fire sensor | Successful fire detection | Completed |
| **4** | Check the DC motor and wheel control features with the ESP 8266 board and the L298N motor | Works correctly | Completed |
| **5** | Check the line moving feature using infrared sensors and DC motors | Successful migration | Completed |
| **6** | Check the automatic stop feature when encountering fire and activate the pump to extinguish fires. | Works correctly |  |
| **7** | Test the feature of sending alerts via the Blynk app when fire is detected. | Alert sent successfully | Rechecking needed |
| **8** | Check for alerts via the Blynk app when you run out of fuel or another problem occurs . | Alert sent successfully | In progress |
| **9** | Test the security features of the esp 8266 board and the Blynk application. | Reliable security | Rechecking needed |
| **10** | Check the stability and reliability of the entire intelligent fire truck system. | Stable operation | Completed |

Table : Test plan

### **Feedback.**

After developing and completing the fire truck project, I gave users product cards, and received user reviews through the following form: <https://forms.gle/Ruv4a5jvWkT8Xpuf7> .

"**How do you feel about using the product?**": This question aims to collect users' opinions and feelings about using the product. Users are asked to share personal feelings, comments or opinions about the experience of using the product. Most users are very satisfied with the product.

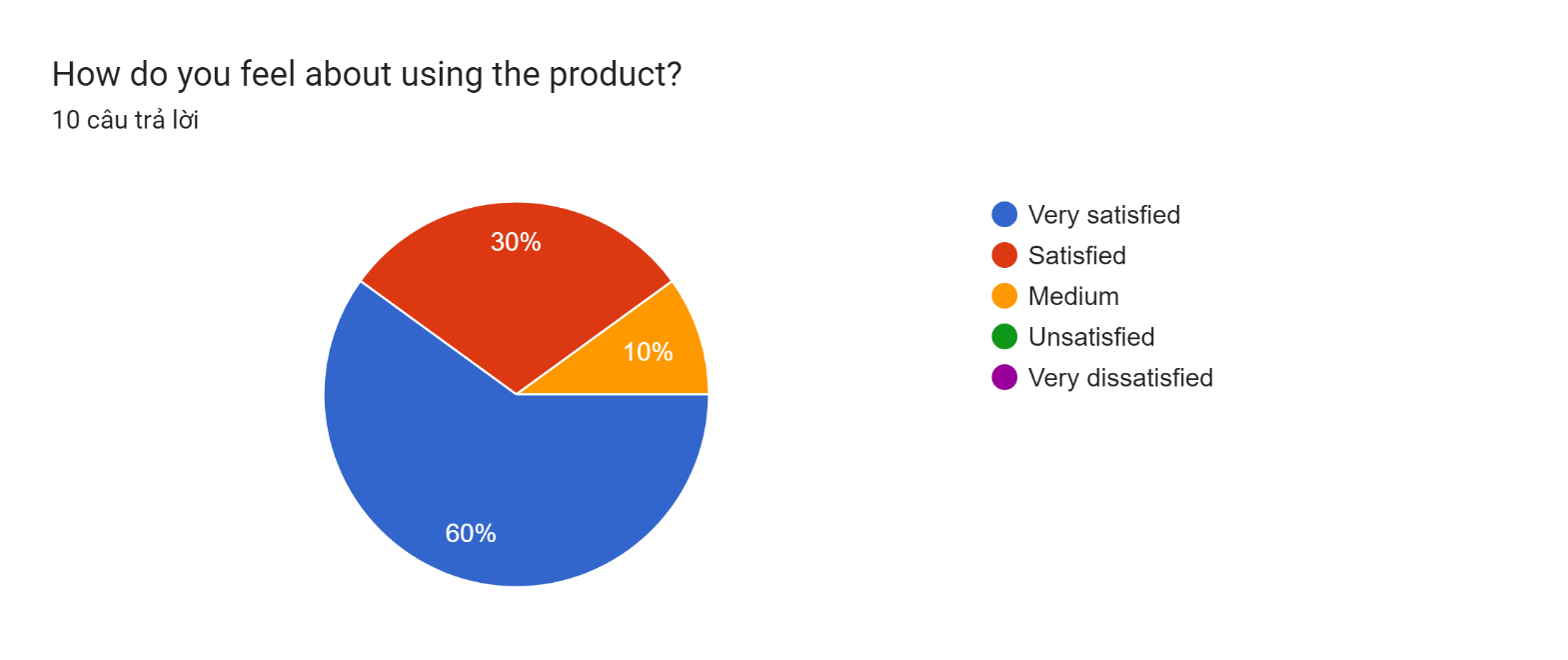


Figure : Feedback 1

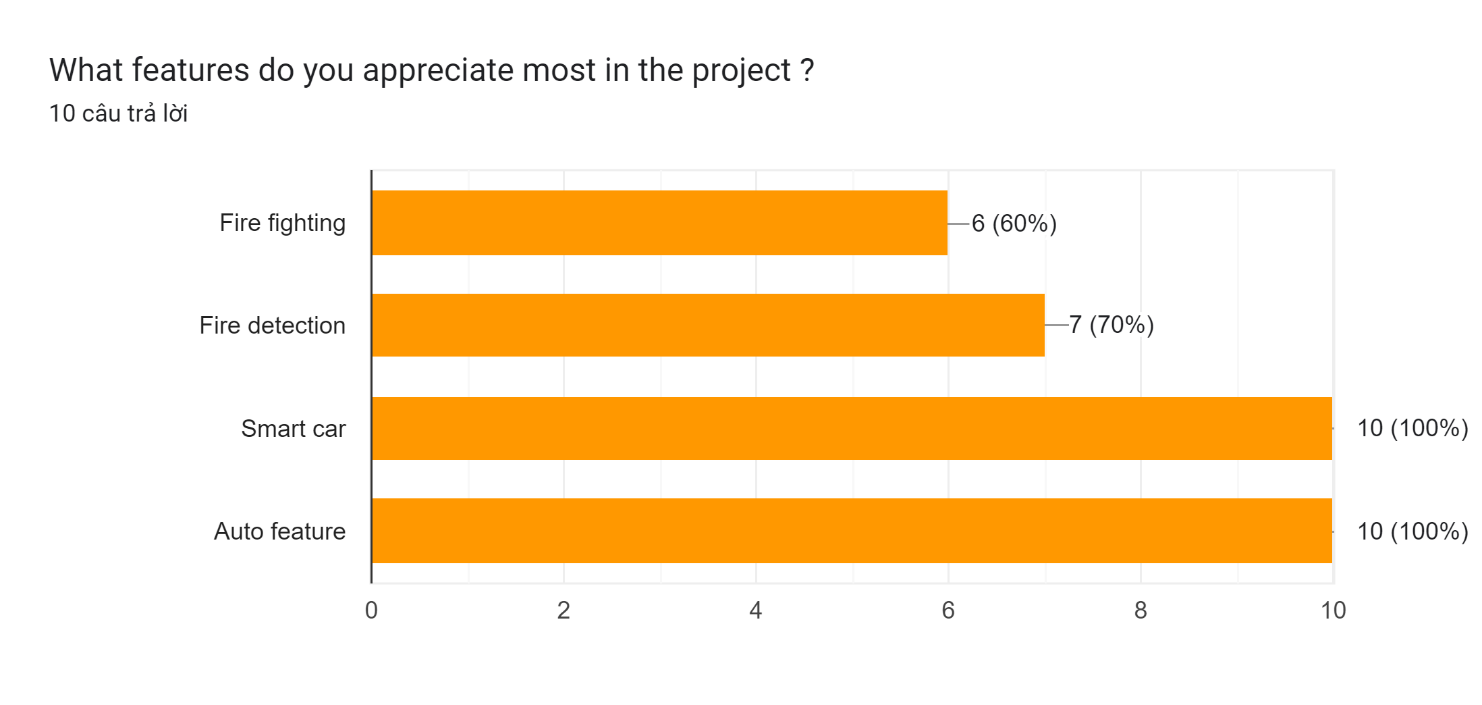
**"What features do you appreciate most in the project?":** This question requires users to choose the features in the smart fire truck project that they appreciate the most. This helps understand which elements are valued by users and create a positive impression. The features highly appreciated by users in the project are the line tracking ability and the car's auto mode, this is smart.

Figure : Feedback 2

A screenshot of a computer

Description automatically generated**"Problems you encountered during the project?":** This question asks users to share about problems or difficulties they encountered while using or participating in the project. This helps developers identify issues that need to be fixed and improve user experience.All users have difficulty controlling the water spray direction because of the limited spray angle.

Figure : Feedback 3

**"Would you recommend the smart fire truck project to others**?": This question asks users' opinions on whether they would recommend the smart fire truck project to others. User opinions can provide important information about project satisfaction and reliability, which can influence the decisions of others when choosing a product. 70% of users will recommend the product to others, 30% of users are still confused by some limitations that still exist on the product.

Figure : Feedback 4

**"Thank you for taking the time to submit your feedback! Do you have any other comments or information you would like to share?":** This question invites users to share any other comments, suggestions or information that they want to share with developers. This gives users an opportunity to suggest ideas, provide additional opinions, and share additional information that was not asked in the previous question. I have received feedback from users about adding the function of adjusting the spray direction and water spray angle to optimize fire fighting.

Figure : Feedback 5A screenshot of a survey

Description automatically generated

These questions provide users with a platform to share opinions, feelings and suggestions about the smart fire truck project. The information gathered from the answers will help developers better understand user opinions and improve the project to better meet user needs.

### **Project improvement**

This is the product after improvement:

# **Conclusion**

The development of a smart firefighting vehicle utilizing IoT technologies has significant potential to revolutionize the firefighting industry. This report has presented a comprehensive analysis and development plan for such a vehicle, focusing on investigating architecture, frameworks, tools, hardware, and API techniques for IoT application development. By leveraging IoT capabilities, the smart firefighting vehicle can overcome the challenges of limited accessibility in densely populated areas with narrow streets and heavy traffic. Real-time fire detection, autonomous navigation, and automated fire suppression functionalities enable the vehicle to respond swiftly and effectively to fire incidents.

The integration of IoT sensors and actuators allows the vehicle to gather and analyze environmental data, monitor fire behavior, and make intelligent decisions to optimize firefighting strategies. This enhances the safety of firefighters and improves the overall effectiveness of fire suppression operations.

Throughout the development process, careful consideration was given to selecting appropriate tools and technologies. Integrated development environments, programming languages, IoT platforms, hardware development kits, simulation and testing tools, and version control systems were employed to streamline the development and ensure the successful implementation of the IoT application.

A detailed test plan was created to validate the functionalities, performance, and robustness of the smart firefighting vehicle. By conducting thorough testing and analyzing the obtained feedback, any issues and areas for improvement were identified and addressed, leading to an optimized and reliable IoT application.

In conclusion, the smart firefighting vehicle project represents a significant advancement in firefighting operations. By harnessing the power of IoT, the vehicle offers enhanced accessibility, efficiency, and safety, ultimately leading to better fire incident response and reduced damage to life and property. The implementation of this project highlights the immense potential of IoT technologies in various industries, including emergency response and public safety. It is expected that further advancements in IoT and related technologies will continue to drive innovation in firefighting and contribute to the creation of safer and more resilient communities.

# **References**

Anon, 2023. *The CTIF World Fire Statistics Report № 27 now available for download | CTIF - International Association of Fire Services for Safer Citizens through Skilled Firefighters..* [Online]   
Available at: https://ctif.org/news/ctif-world-fire-statistics-report-no-27-now-available-download.  
[Accessed 3 12 2023].

Bulletin, M., 2023. *Number of PH fire incidents rises by 8.3% in first 4 months of 2023- BFP.* [Online]   
Available at: https://mb.com.ph/2023/5/22/ph-fire-incidents-rises-by-8-3-in-first-4-months-of-2023-bfp.  
[Accessed 3 12 2023].

ieeexplore.ieee.org, n.d. *Design and Implementation of IoT based Multi Degree Rotating Fire Extinguisher System | IEEE Conference Publication | IEEE Xplore.* [Online]   
Available at: https://ieeexplore.ieee.org/document/10140798  
[Accessed 3 12 2023].

IOTLINK, 2023. *IoT là gì? Kiến thức cơ bản về Internet of Things bạn cần phải biết.* [Online]   
Available at: https://iotlink.com.vn/iot-la-gi-kien-thuc-co-ban-ve-internet-of-things-ban-can-phai-biet/  
[Accessed 10 12 2023].

Jelvix, n.d. *What Makes an IoT Solution Comprehensive? 7 Layers of IoT Architecture.* [Online]   
Available at: https://jelvix.com/blog/iot-architecture-layers  
[Accessed 8 12 2023].

Jena, S., 2020. *Architecture of Internet of Things (IoT).* [Online]   
Available at: https://www.geeksforgeeks.org/architecture-of-internet-of-things-iot/  
[Accessed 8 12 2023].

Pisonero, J. G.-G. E. G.-M. R. a. G.-A. D., 2023. *Smartvessel: A New Extinguisher Prototype Based on New Materials and IoT Sensors. Sensors.* [Online]   
Available at: https://www.mdpi.com/1424-8220/23/6/3134  
[Accessed 3 12 2023].

Reuters, 2023. *Fire kills at least 56 in nine-story building in Vietnam.* [Online]   
Available at: https://edition.cnn.com/2023/09/13/asia/vietnam-hanoi-apartment-fire-intl-hnk/index.html  
[Accessed 3 12 2023].

Roberto Garcia-Martin, A. G.-B. J. M., 2019. *SmartFire: Intelligent Platform for Monitoring Fire Extinguishers and Their Building Environment. Sensors.* [Online]   
Available at: https://www.mdpi.com/1424-8220/19/10/2390  
[Accessed 3 12 2023].

Statista, n.d. *Vietnam: number of deaths by traffic accidents 2018.* [Online]   
Available at: https://www.statista.com/statistics/986123/vietnam-number-deaths-traffic-accidents/.  
[Accessed 3 12 2023].

Tatarski, M., 2023. *Vietnam: 56 people dead after fire at nine-floor apartment building. The Guardian..* [Online]   
Available at: https://www.theguardian.com/world/2023/sep/13/vietnam-apartment-fire-hanoi-deaths  
[Accessed 3 12 2023].

Thietbimang.com, n.d. *IoT là gì? Kiến trúc hệ thống mạng IoT Internet vạn vật kết nối mọi thiết bị | thietbimang.com.* [Online]   
Available at: https://thietbimang.com/tin-tuc/IoT-la-gi-Kien-truc-he-thong-mang-IoT.html  
[Accessed 10 12 2023].