International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 09 – Issue 12 || December 2024 || PP. 01-09

Smart Medicinal Plant Cultivation Model Applying IoT and AI Technologies

Nguyen Tai Tuyen[1], Luong Khac Dinh[2], Vu Duy Hung[2], Nguyen Quang Ngoc[1]
[1]*Post and Telecommunication Institute of Technology, Hanoi, Viet Nam*
[2]*Ha Long University, Uong Bi, Quang Ning, Viet Nam*

Abstract: This paper presents the research results of building an intelligent cultivation model for the development of medicinal materials using IoT and AI technology and environmental monitoring sensors in cultivation to enhance the efficiency and quality of crops. The system is capable of automatically adjusting devices according to activation thresholds and provides collected information, such as temperature, humidity, and images, via a mobile app developed by the author team. The demo model has been developed and introduced into practical teaching for university students, focusing on the application of medicinal material development in particular, and agriculture in general. It creates an environment for developing IoT applications through the demo model.

Keywords: IoT, Firebase, ESP32, Medicinal materials, Automation, PTIT, UHL

I. INTRODUCTION

1. Internet of things application development tools

a) Firebase

Firebase is a Backend as a Service (BaaS) platform developed by Google, designed for mobile and web application development. Firebase [9] offers various services such as data storage, user authentication, notifications, analytics, and more, enabling the development of backends for web and mobile applications.

b) Google Cloud API

Google Cloud API, provided by Google, helps developers interact with and build on Google Cloud services [8]. It enables developers to create, manage, and customize applications efficiently. In this article, the Google Sheets API is used to automatically update and edit data [10].

c) MIT App Inventor

MIT App Inventor (App Inventor or MIT AI2), this is a block-based programming platform [12]. It allows users to create computer applications for operating systems like Android and iOS. The platform is open-source and free to use.

d) Visual Studio 2022

Visual Studio 2022 [11] offers powerful features that support software developers in programming, debugging, and deployment. It was used to develop the research group's application program.

e) Arduino IDE

Arduino IDE is an opensource tool used primarily to write and compile, load code for IoT applications [6] such as Arduino modules, ESP [1, 2, 3, 7] and other embedded modules.

2. Modules used in developing demo models in pharmaceutical development

a) ESP32 and ESP32 CAM processing modules

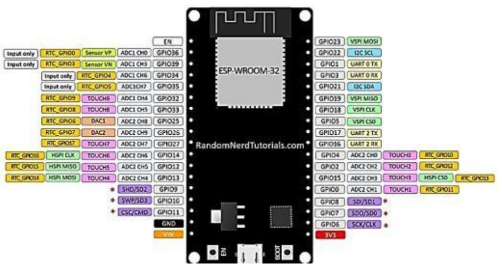

Figure 1 ESP32 Pinout

Figure 2 ESP32 CAM IoT Module

b) AHT30 sensor module and soil moisture

Figure 3 AHT30 sensor module

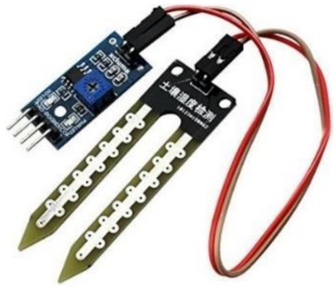

Figure 4 Soil moisture sensor

c) Relay module supplies power and Irrigation water pump

Figure 5 Relay module supplies power

Figure 6 Water pump used in the model

3. Medicinal materials

According to Clause 5, Article 2 of the 2016 Pharmacy Law: Medicinal materials are medicinal materials of natural origin from plants, animals, minerals and meet the standards for medicinal use [13].

II. System Design

1. Determine the objectives

Build an app on the phone and assemble a demo model. The operating model is compatible with many types of medicinal materials and is easy to adjust edit and improve to suit each type of medicinal plant in practice.

2. System requirements

- Automatic control
- Automatic collection of information and statistics
- Apply AI to analyze the growth status and diseases of medicinal plants

3. System operation

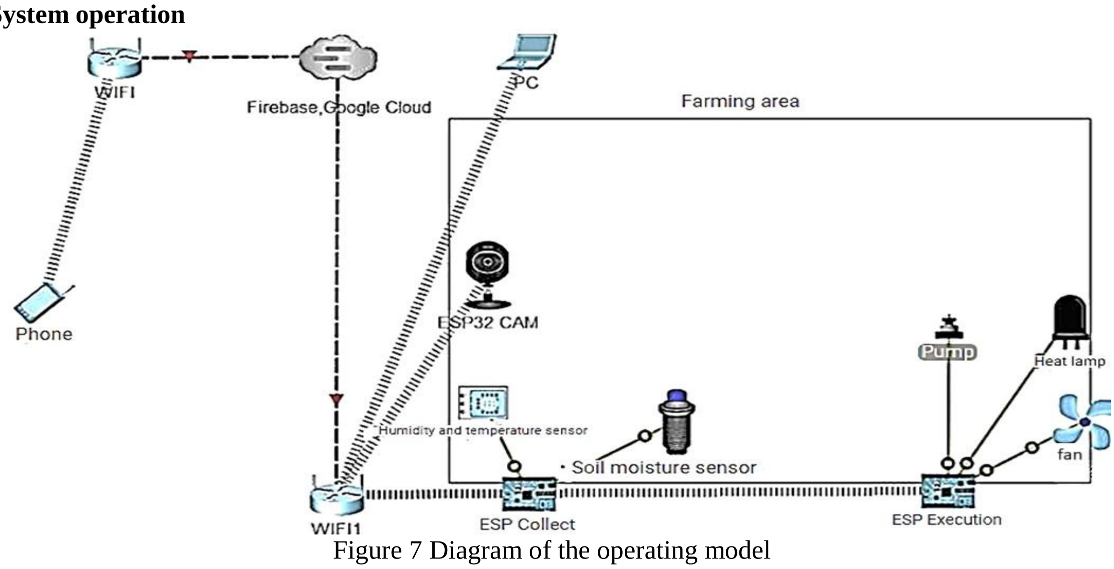

Figure 7 Diagram of the operating model

Users can control and monitor their medicinal herb garden through the app installed on their smartphones by the development team.

Data will be stored in the cloud. The ESP module will connect to Wi-Fi to send and receive data via Firebase.

The ESP32-CAM module is responsible for sending images to web servers. The computer retrieves these images from the web server and uses them to diagnose the status of the medicinal plants.

Device operations, such as turning devices on or off, are executed by comparing the actual values with the threshold values set by the user in the system.

III. Applied Algorithms in Smart Medicinal Herb Cultivation

1. System initialization

Step 1: Configure Firebase
- Connect ESP32 to Firebase (URL and API key).
- Initialize necessary libraries on Arduino IDE (FirebaseESP32, WiFi, AHT30).
- Define variables to store sensor data and device status.

Step 2: Install sensors and devices
- Attach AHT30 sensor, soil moisture sensor to ESP32.
- Connect ESP32 CAM to send images.
- Connect L298N module to fan, pump, heating lamp.

Step 3: Configure phone application
- Design interface to display parameters (soil moisture, temperature, air humidity) and device status.
- Integrate Firebase into the application to receive and send real-time data.

2. Collect sensor data

Step 1: ESP32 reads data from sensor
- Read soil moisture value from soil moisture sensor (ADC).
- Read temperature and air humidity from the AHT30 sensor.
- Convert analog signal to percentage or degree Celsius value.

Step 2: ESP32 CAM sends image
- Take photos of medicinal plants.
- Send a photo to Firebase via HTTP protocol or store on the server.

3. Synchronize data with Firebase

Step 1: Save sensor data
- Send sensor values (soil moisture, temperature, air humidity) to Firebase.
- Send device status (pump, fan, heating lamp) to Firebase.

Step 2: Synchronize with application
- Update phone application interface when data on Firebase changes.
- Allow users to set thresholds through the application, update thresholds to Firebase.

4. Data analysis and decision making

Step 1: Check the sensor value against the threshold:
- If soil moisture < threshold, turn on the water pump.
- If temperature > threshold, turn on the ventilation fan.
- If temperature < threshold, turn on the heating lamp.

Step 2: Update the device status to Firebase:
- Record the on/off status of the pump, fan, and lamp on Firebase.
- Display the device status on the application.

Step 3: Analyze the tree image (Python application):
- Download the image from ESP32 CAM.
- Use TensorFlow to identify the tree status (green, water-deficient, diseased).
- Send diagnostic results to Firebase to display on the app.

5. Update data

Step 1: Send data to Google Sheets:
- Set up Google Cloud API.
- Automatically send sensor parameters and device status to Google Sheets periodically.

Step 2: Store and analyze data:

- Save measurement history and device status for system analysis and optimization.

6. Repeat the process

Step 1: Send data to Google Sheets:

- Set up Google Cloud API.
- Automatically send sensor parameters and device status to Google Sheets periodically.

Step 2: Store and analyze data:

- Save measurement history and device status for system analysis and optimization.

IV. Develop Control Applications on Smartphones

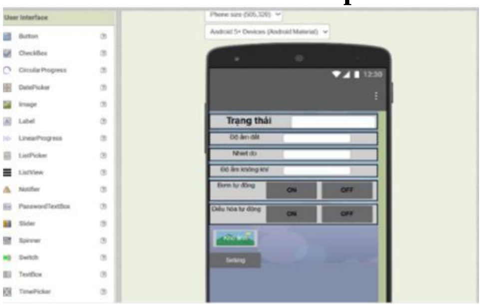

Figure 8. Mobile Application Development Interface

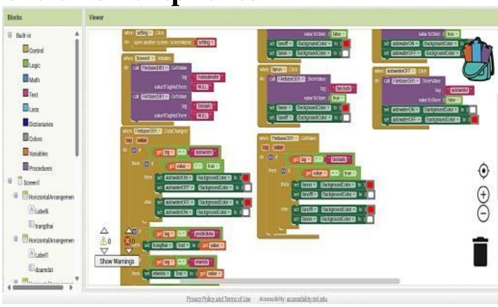

Figure 9. Logic Processing

The in figure 8 interface includes:

+ Medicinal herbs status line
+ Soil moisture line
+ Temperature line
+ Air humidity line
+ Pump mode on/off buttons and thermostat when used in greenhouses.
+ Keys to set activation thresholds

The block checks data from Firebase, when the data on Firebase changes, the data displayed in the control application on the phone will also change accordingly.

The block pushes data to firebase and at the same time changes the color of the on and off buttons to help users recognize the operating status of the system.

7. Python application

Use the Tensor flow library and collect images to train the model. Has the function of diagnosing images from the camera to infer the status of medicinal plants.

Figure 10. Python application diagnostic interface

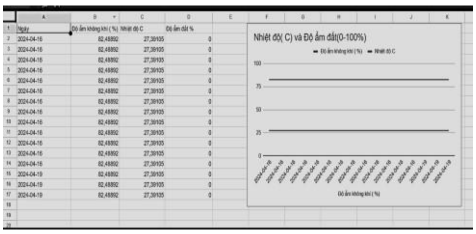

Figure 11. Data is uploaded to google sheet

First step, the user uploads images and creates labels according to the file then click the "Learn" button to train the data.

Enter the IP of ESP32 CAM to get images from ESP32 CAM then clicks the "Auto upload" button to send images to Firebase. Click the "Run auto-diagnosis" button to diagnose the status of the medicinal herbs. The application will analyze the image based on the learned model and give the results then display the status of the control application on the phone app. Next, the user will access the page https://docs.google.com/spreadsheets/create and create a new file then share it to the account on the system set up. Data will be automatically sent to google sheet without human interaction.

8. Upload code to ESP32 and ESP32 CAM

a. ESP32 Module

Create a new project on Arduino IDE. The code after being developed will be loaded into the module using the Arduino IDE tool, at the development interface.

Figure 12. Create new project Arduino IDE

For the ESP32 project, the Board selected is the DEVKIT V1 Board and loaded with Code. ESP32 reads analog data with values from 0-4095 (4096 levels) and humidity will have to convert soil humidity data (physical quantity) in analog form to percentage using the formula: $y = (x / 4095) * 100\%$

In which: x is the analog value from the soil humidity sensor and y is the soil humidity in percentage (%) received.

For air humidity and temperature, the AHT30 library already has it and automatically converts it to degrees Celsius and (%) humidity.

b. ESP32 CAM Module

With the ESP32 CAM Module, you need to select the AI THINKER ESP32 CAM Board, select the COM port and the corresponding board to load code for the module.

c. Pairing ESP with other devices

With L298, the power source must be separated from ESP32. Because the external power source must control L298N for 2 motors or relays, a better power supply is needed.

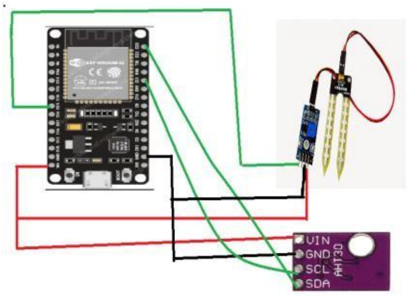

Figure 13. Pairing ESP32 with AHT30, soil moisture sensor

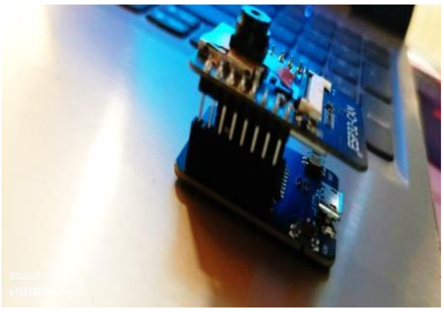
Figure 14. Pairing the ESP32 CAM device

9. Installation and testing

In the testing section, we will perform the steps of installing the program, configuring the system and deploying the test with the smart farming model in medicinal plant development.

a. Set up Firebase

Create a project and get the API key

Go to Service accounts to get the json used for the python app. When running the python app, the service account used will be linked to google cloud for good response. You can run 1 service account for many applications at the same time.

b. Choose the medicinal plant to test

Choose the Phyllanthus urinaria plant for testing because it is not as difficult to find as the medicinal plants in the area, easy to grow. The small plant is suitable for use in the demo model presented in this article.

Figure 15 Phyllanthus urinaria

According to oriental medicine, the Vietnamese plant Phyllanthus urinaria has a slightly bitter taste, cool properties, and has antiseptic, anti-inflammatory, detoxifying, blood-clearing, and diuretic effects.

c. Upload the code and assemble the demo model

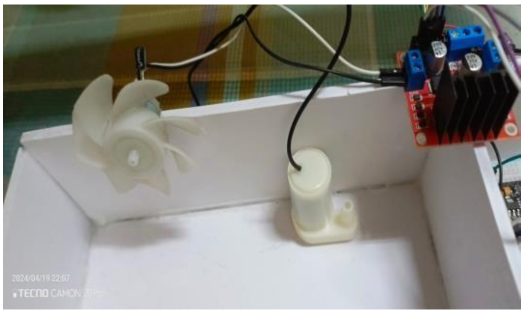

Figure 16 Connect the fan and pump to L298N

Figure 17 Connect the L298N and the sensor to ESP

The positive pin is connected to IN1 and IN3, the negative pin is IN2 and IN4.
Connect the sensor pin to esp32.GND with the corresponding GND pin. The sensor will prioritize using the 3v3 pin. Connect the long-legged LED to GPIO and the short leg will be GND.

d. Control the program test result

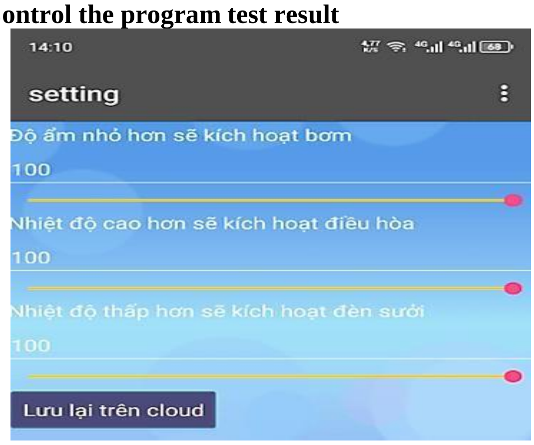

Figure 18 Custom trigger threshold interface

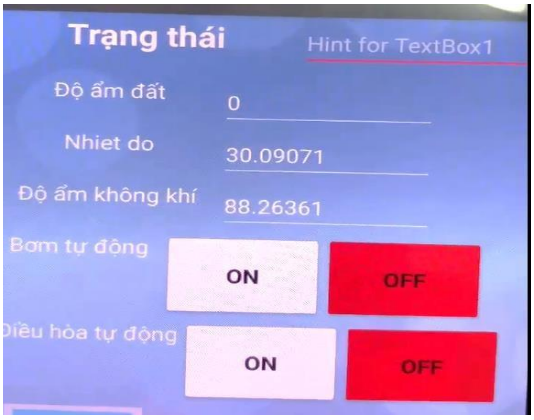

Figure 19 The interface displays the status control buttons.

The soil moisture is lower than the threshold, the water pump is turned on, otherwise the pump will turn off. The temperature is higher than the threshold, the system will turn on the ventilation fan until the temperature reaches the required level, the fan system will turn off. The temperature is lower than the trigger threshold, the heating system will automatically turn on and operate until the temperature reaches the set threshold, the heating system will turn off.

The user will observe the state of the garden and turn on/off the automatic mode here.

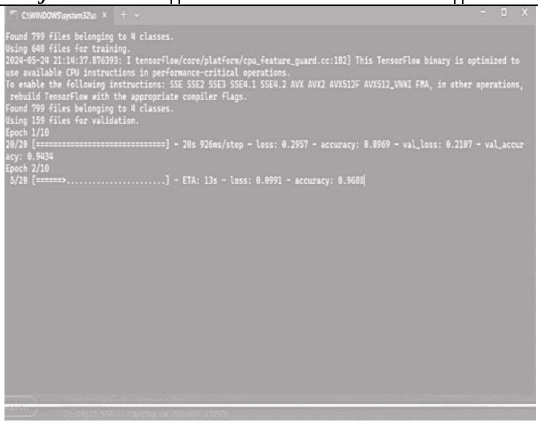

Figure 20. The Python application uses the Tensor flow library.

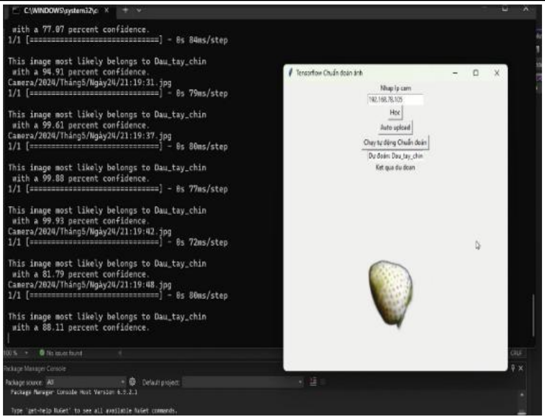

Figure 21. This is an image with the background removed and entered into the diagnosis (e.g. strawberry).

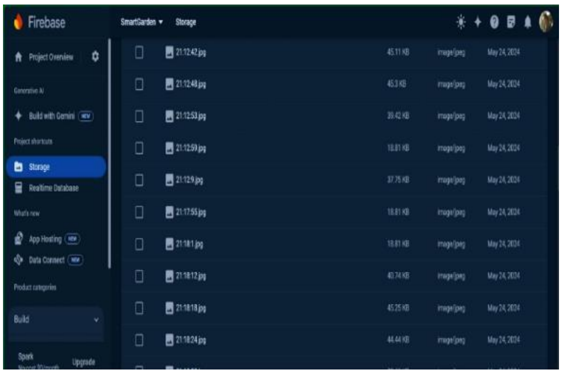

Figure 22 Above is an image of a herb that is in the process for storage of training data for disease diagnosis.

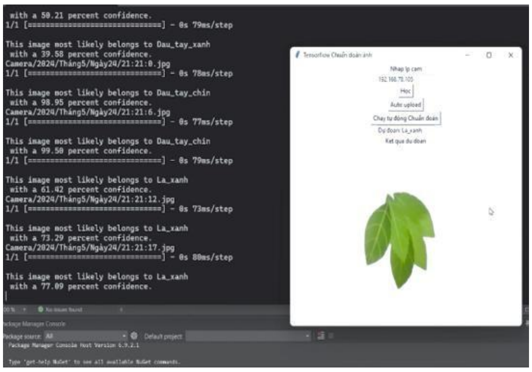

Figure 23 This is an image sent to the cloud and remote observation via phone.

Using leaves to test the growth status diagnosis feature of the herb Phyllanthus urinaria. The data is recognized as still green.

V. Discussion

1. Tools and platforms used

- Firebase helps store and synchronize data in real time, ensuring continuous updates of new data from sensors and IoT devices.
- Arduino IDE and Google Cloud API allow convenient processing, programming and device management.
- App Inventor creates a simple but effective user interface, suitable for smartphone control applications.
- Integrating TensorFlow with Python helps analyze images and apply AI to diagnose the status of medicinal plants.

2. System model

- The system has been tested on Phyllanthus urinaria and achieved good results in automatically collecting information from sensors, analyzing data, and activating automatic devices based on set thresholds.
- Modules such as ESP32 CAM and AHT30 operate stably, providing reliable data on the growth status of plants.

3. Practical application

- Strengths
  + Automation: The system can operate independently and connect to the cloud, allowing remote control via smartphone.
  + Smart data analysis: Integrating AI to detect the growth and disease status of plants. This is a step forward in applying technology to smart agriculture.

  + Scalability: The system is designed flexibly, easily adjusted to suit different types of medicinal plants.

- Challenges:
  + Limitations in real environments: The current model is limited to testing in a small experimental environment. Largescale deployment or in natural conditions may require additional expansion of sensors and neural network applications for sensor modules, Lora connectivity, and system security and device protection.
  + Complexity when upgrading: Adding functions or compatibility with other medicinal plants may require time and cost for reprogramming and integrating new equipment.

4. Development direction
- Model improvement
- Integrate additional sensors to monitor other indicators such as light, soil nutrient levels to better serve othermedicinal plants.
- Use solar energy for power supply, helping the system operate stably in areas far from the grid.
- Enhance AI:
  + Train AI models with larger datasets to increase accuracy in plant status analysis.
  + Integrate the ability to detect more diseases on medicinal plants.

5. Wider practical applications
- Deploy testing on other popular medicinal plants.
- Develop additional warning features via text messages or emails when plants show signs of disease or the system has problems.

VI. Conclusion

During the research and testing process, the team successfully built a demo model for developing smart medicinal plants, using IoT and AI technology. IoT application development tools such as Firebase, Google Cloud API, App Inventor, Visual Studio 2022, and Arduino IDE have been effectively used in building and developing the system.

The demo model has been tested with Phyllanthus urinaria, a popular medicinal herb, and has shown positive results. The system automatically collects information and statistics, automatically controls devices such as water pumps to compensate for water (soil moisture) for plants, ventilation fans, and heating lamps based on set thresholds.

A special feature is the application of AI to analyze the growth and disease status of medicinal plants through images collected from ESP32 CAM, allowing users to monitor and control the system remotely via an application on a smartphone.

However, the system is implemented in the form of a model. Therefore, in order to be applied in practice, it needs to be further improved and developed to be more widely applied in industry, especially with many different types of medicinal herbs.

VII. Acknowledgements

The authors would like to thank the leaders of the Faculty of Information Technology and the leaders of Ha Long University; the Board of Directors of the Academy of Posts and Telecommunications Technology for creating the best conditions for the research team to complete the article.

References

[1] Nguyen Tai Tuyen (2022), On an application in supporting practical teaching of IoT course and embedded programming, Global Journal of Engineering and Technology Advances, 2022, 13(03), 039–044.

[2] Nguyen Tai Tuyen and associates (2023), Demo Model in Smart Home Application Design, International Journal of Science, Engineering and Technology, 83-86, 12/2023.

[3] Nguyen Tai Tuyen and associates (2021), On an application of node MCU Esp8266 in opening and closing the laboratory door - online practice, Global Journal of Engineering and Technology Advances 12, 2021.

[4] Tuyen Nguyen Tai (2021), Automatically shedding loads for lighting systems in corridors and public areas in high-rise apartment buildings, Global Journal of Engineering and Technology Advances, pp. 056-061.

[5] Ai-Thinker, "ESP32-CAM", Available: https://www.ai-thinker.com/esp32-cam.
[6] Arduino, "Arduino IDE", Available: https://www.arduino.cc/en/software.
[7] Espressif "ESP32", Available: https://www.espressif.com/en/products/socs/esp32.
[8] Google Cloud, "Google Cloud API", Available: https://cloud.google.com/apis.
[9] Google Firebase, "Firebase", Available: https://firebase.google.com/.
[10] Google Sheets, "Google Sheets", Available: https://docs.google.com/spreadsheets/. [11]Microsoft, "Visual Studio 2022", Available: https://visualstudio.microsoft.com/vs/.
[12] MIT App Inventor, "MIT App Inventor", Available: http://appinventor.mit.edu/.
[13] http://vanban.chinhphu.vn/portal/page/portal/chinhphu/hethongvanban.
[14] STMicroelectronics, "L298N", Available: https://www.st.com/en/motor-drivers/l298.html.
[15] TensorFlow, "TensorFlow", Available: https://www.tensorflow.org/.