

**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

**FACULTY OF COMPUTER NETWORKING AND COMMUNICATIONS**

-----🙞🙜🕮🙞🙜-----

**PROJECT REPORT**

**PROJECT  
Wireless Embedded Network Systems**

**TOPIC: DEVELOPMENT OF SECURITY ALARM SYSTEM USING ESP32 AND HC-SR501.**

*Supervisor:*

Quan LE-TRUNG, Assoc. Prof. Dr.techn

*Students:*

Nguyen Hoang Diep Phi 21522453

Nguyen Trong Phuc 21522476

**Ho Chi Minh City, November 28, 2023**

THANK YOU  
  
   
  
After studying and training at the Embedded wireless network system at the University of Information Technology - Vietnam National University, Ho Chi Minh City, our group has been equipped with basic knowledge and practical skills to be able to complete into the group's Project.  
  
To complete this map, please be deeply grateful. The team would like to send sincere thanks to:  
  
Board of Directors of the University of Information Technology - Vietnam National University, Ho Chi Minh City for creating favorable conditions for students to search and research information with a modern library system, diverse types of books and documents.  
  
Closer, the best words would like to send to Associate Professor Dr. Le Trung Quan for taking advantage of the situation to help, orient thinking and guide scientific work. These are extremely valuable advice not only during the project implementation process but also as the next action steps for me in my future study and work process.  
  
Finally, I wish the teachers at the School of Information Technology in general good health and confidence to continue to carry out their noble mission.  
  
Ho Chi Minh City, November 29, 2023

**TEACHER COMMENT**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**...............................................................................................................................................................**

**​Table of Contents**

[**I.**](#_heading=h.gjdgxs) **INTRODUCTION 5**

[1.](#_heading=h.30j0zll) Reason for Choosing the Topic 5

[1.1.](#_heading=h.1fob9te) Why we choose ESP32 and HC-SRC501 6

[2.](#_heading=h.3znysh7) Target Audience 7

[3.](#_heading=h.2et92p0) Hardware Components 8

[3.1.](#_heading=h.tyjcwt) ESP32 Dev Board 8

[3.2.](#_heading=h.3dy6vkm) HC-SR501 11

[4.](#_heading=h.1t3h5sf) Software 14

[4.1.](#_heading=h.4d34og8) Arduino IDE 14

[4.2.](#_heading=h.2s8eyo1) ESP RainMaker 15

[**II.**](#_heading=h.17dp8vu) **RELATED WORK/LITERATURE REVIEW 16**

[**III.**](#_heading=h.3rdcrjn) **METHODLOGY 16**

[**IV.**](#_heading=h.26in1rg) **CODE ANALYSIS 16**

[**V.**](#_heading=h.lnxbz9) **CONCLUSIONS 18**

[1.](#_heading=h.35nkun2) Evaluate 18

[1.1.](#_heading=h.1ksv4uv) Advantages 18

[1.2.](#_heading=h.44sinio) Disadvantages 18

[2.](#_heading=h.2jxsxqh) Conclusion 18

[2.1.](#_heading=h.z337ya) Result 18

[2.2.](#_heading=h.3j2qqm3) Advantage 18

[2.3.](#_heading=h.1y810tw) Defect 19

[3.](#_heading=h.4i7ojhp) Development orientation 19

[**VI.**](#_heading=h.2xcytpi) **REFERENCES 19**

# INTRODUCTION

## Reason for Choosing the Topic

The decision to select the "Security Alarm System" topic is driven by the contemporary demand for enhanced security solutions. In today's dynamic environment, where safety is a paramount concern, the rationale for this choice can be articulated as follows:

1. Increasing Security Concerns:
   * The prevalent rise in security threats and concerns necessitates the development of advanced and accessible security solutions.
   * Instances of unauthorized access, burglaries, and security breaches have underscored the need for reliable and efficient alarm systems.
2. Affordable and Customizable Solutions:
   * The project aims to provide an affordable yet effective security solution that can be customized to meet the specific needs of users.
   * Recognizing the financial constraints of individuals and small businesses, an adaptable security alarm system becomes crucial.
3. Integration with Smart Homes:
   * The growing trend towards smart homes and IoT (Internet of Things) solutions accentuates the demand for integrated security systems.
   * A security alarm system that can seamlessly integrate with existing smart home infrastructure enhances overall home automation and security.
4. Accessibility and User-Friendly Implementation:
   * Open-source Community: ESP32's open-source nature and a supportive community offer opportunities for customization and continuous improvement.
   * Expandable Features: The project can be extended with additional sensors, sirens, or integration with existing smart home systems, enhancing its capabilities.
5. Educational Value:
   * Developing a security alarm system provides valuable educational opportunities, allowing individuals to gain hands-on experience in areas such as sensor integration, microcontroller programming, and system design.
   * It serves as a practical and applied learning experience for students and enthusiasts interested in the intersection of security and technology.

In conclusion, the choice of the "Security Alarm System" topic is driven by the contemporary landscape where security is a paramount concern. The project aims to address this concern by providing an affordable, customizable, and accessible security solution that aligns with the evolving needs of individuals and businesses.

### Why we choose ESP32 and HC-SRC501

The selection of the "Security Alarm with ESP32 and HC-SR501" topic stems from the increasing need for cost-effective and efficient security solutions in various settings. This project integrates the powerful capabilities of ESP32, a versatile microcontroller, with the precision of the HC-SR501 Passive Infrared (PIR) motion sensor.

Versatility of ESP32:

* + Connectivity: ESP32 provides robust connectivity options, including Wi-Fi and Bluetooth, enabling remote monitoring and control of the security system.
  + Processing Power: The dual-core processor and ample memory of ESP32 allow for complex algorithm implementation, enhancing the overall functionality of the security system.

Precision of HC-SR501:

* + Motion Sensing: HC-SR501 PIR sensor excels in detecting human presence through infrared radiation, making it ideal for security applications.
  + Cost-effectiveness: HC-SR501 offers a cost-effective solution without compromising on accuracy, making it suitable for a wide range of security projects.

Practical Application:

* + Home Security: The project addresses the practical need for a reliable and affordable home security solution, allowing users to monitor and receive alerts in case of unauthorized access.
  + Educational Value: Implementing a security alarm system with widely-used components like ESP32 and HC-SR501 provides valuable hands-on experience for learning about IoT, sensor integration, and security systems.

1. Customization and Expandability:
   * Open-source Community: ESP32's open-source nature and a supportive community offer opportunities for customization and continuous improvement.
   * Expandable Features: The project can be extended with additional sensors, sirens, or integration with existing smart home systems, enhancing its capabilities.

In conclusion, the "Security Alarm with ESP32 and HC-SR501" project aligns with the growing demand for accessible and customizable security solutions. The combination of ESP32's versatility and HC-SR501's precision makes this project a practical and educational endeavor with potential real-world applications.

## Target Audience

The project is designed to cater to a diverse audience with varying needs and interests. The primary target audience includes:

1. Homeowners:
   * Individuals seeking a cost-effective yet robust security solution for their homes.
   * Homeowners interested in enhancing their home automation setup with a reliable security alarm system.
2. Small Business Owners:
   * Small businesses looking for an affordable security system to protect their premises during non-operating hours.
   * Entrepreneurs aiming to integrate a security solution into their existing infrastructure.
3. DIY Enthusiasts and Students:
   * DIY enthusiasts interested in building and customizing their security systems using widely available components like ESP32 and HC-SR501.
   * Students studying electronics, IoT, or related fields, who can utilize the project for practical learning and hands-on experience.
4. Security-Conscious Users:
   * Individuals prioritizing security in their daily lives and seeking an accessible and customizable solution.
   * Users looking for a user-friendly security system with features such as remote monitoring, alerts, and integration with other smart devices.

## Hardware Components

The project "Security Alarm with ESP32 and HC-SR501" utilizes a combination of essential hardware components to ensure the effectiveness of the security system. Here are the key components:

### ESP32 Dev Board

#### Introduction

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica’s 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.Ảnh có chứa đồ điện tử, Linh kiện điện, Thành phần mạch điện, Kỹ thuật điện

Mô tả được tạo tự động

ESP32 is that it is manufactured using TSMC’s ultra-low-power 40 nm technology. So, designing battery operated applications like wearables, audio equipment, baby monitors, smart watches, etc., using ESP32 should be very easy.

#### Specifications of ESP32

The ESP32 surpasses its predecessor, the ESP8266, with a myriad of enhanced features. While it's challenging to encompass all specifications, we've highlighted key aspects below:

* Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
* 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
* Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
* Support for both Classic Bluetooth v4.2 and BLE specifications.
* 34 Programmable GPIOs.
* Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
* Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
* Ethernet MAC for physical LAN Communication (requires external PHY).
* 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
* Motor PWM and up to 16-channels of LED PWM.
* Secure Boot and Flash Encryption.
* Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.

#### Diverse Programming Options for ESP32

A high-quality hardware platform like the ESP32 becomes even more user-friendly when it offers versatility in programming options. Notably, the ESP32 supports multiple programming environments, catering to a wide range of preferences and requirements.

* + - * 1. Among the commonly utilized programming environments are:
        2. Arduino IDE
        3. PlatformIO IDE (VS Code)
        4. LUA
        5. MicroPython
        6. Espressif IDF (IoT Development Framework)
        7. JavaScript

Given the familiarity of Arduino IDE, it will serve as our primary programming environment in this projects. However, the ESP32's flexibility allows users to explore and experiment with other programming environments according to their preferences and project needs.

#### Layout:

We will see what a typical ESP32 Development Board consists of by taking a look at the layout of ESP32 DevKit Board.

Ảnh có chứa văn bản, Kỹ thuật điện, mạch điện, đồ điện tử

Mô tả được tạo tự động

As you can see from the image, the ESP32 Board consists of the following:

* ESP-WROOM-32 Module
* Two rows of IO Pins (with 15 pins on each side)
* CP2012 USB – UART Bridge IC
* micro–USB Connector (for power and programming)
* AMS1117 3.3V Regulator IC
* Enable Button (for Reset)
* Boot Button (for flashing)
* Power LED (Red)
* User LED (Blue – connected to GPIO2)
* Some passive components

#### ESP32 PINOUT

The board which we use has 30 Pins (15 pins on each side). There are some board with 36 Pins and some with slightly less Pins.

Ảnh có chứa văn bản, ảnh chụp màn hình, Nhiều màu sắc, Phần mềm đa phương tiện

Mô tả được tạo tự động

An interesting point about the USB-to-UART IC is that its DTR and RTS pins are used to automatically set the ESP32 in to programming mode (whenever required) and also rest the board after programming.

#### Conclusion

The ESP32 Dev Board serves as the central processing unit for the security alarm system. It is a powerful and versatile microcontroller that provides Wi-Fi and Bluetooth connectivity, making it suitable for IoT applications.

### HC-SR501

#### Introduction

The HC-SR501 is a widely used Passive Infrared (PIR) motion sensor that plays a pivotal role in detecting movement within its range. It uses infrared waves for detection of an object. It is an automatic control device, also have large sensitivity and high reliability. It is used in auto-sensing control devices, where we need to perform motion detection. HC-SR501 is used in industrial projects and buildings for security purposes.Ảnh có chứa bánh xe, đồ chơi

Mô tả được tạo tự động

Developed by HENGXING Technology Co., Ltd., the HC-SR501 has gained popularity for its simplicity, effectiveness, and affordability in various applications, particularly in the realm of security systems and home automation.

#### HC-SR501 PINOUT

HC-SR501 has a total of three pinout, which are:

* **PIN 1:** This pin is Vcc, it is used for input voltage. Its input voltage varies from 5V to 12V.
* **PIN 2:** It's the OUT Pin which is fed to the microcontroller.
* **PIN 3:** We have to apply ground on this pin.

Ảnh có chứa mạch điện, Linh kiện điện, Kỹ thuật điện, Thành phần mạch điện

Mô tả được tạo tự động

#### Working of HC-SR501

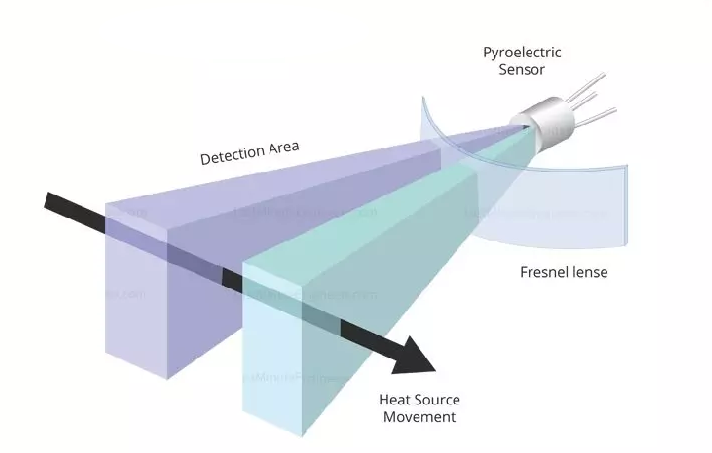
Every living object with a temperature above Absolute Zero (0 Kelvin / -273.15 °C) emit heat energy in the form of infrared radiations.

The hotter an object is the more radiation it emits. Human body works on a similar pattern and emits heat energy.

HC-SR 501 sensor is designed to detect such level of infrared radiation. It basically consists of two main parts:

* **A Pyroelectric Sensor.**
* A special lens called **Fresnel lens** which focuses the infrared signals onto the pyroelectric sensor.

A pyroelectric sensor has two rectangular slots in it, which made of such material which allow infrared radiation to pass through it.



Behind these two slots, there are two sensor electrodes

* One responsible for positive output.
* Second for negative output.

The two electrode wire up so that they cancel each other out. If one half sees less or more infrared radiations then other, the output will swing high or low.

**When the sensor is idle:** If there is no movement around the sensor, both slots detect the same amount of infrared radiations, resulting in a zero output signal.

**When a warm body like a human or animal passes by:** If someone pass by as sensor then, it first intercept one half of the sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happen, then the sensor generates a negative differential change. The Corresponding pulse of signals results in the sensor setting its output pin high.

#### HC-SR501 Features

* Wide range of voltages we can apply on its input varying from 4.V to 12V (+5V recommended).
* Its best feature is that it can distinguish between men movement and object movement.
* Its Output voltage is High/Low (3.3V TTL).
* It cover a distance of 7 meter and an area of 110 degrees.
* It's operating temperature is from -20° to +80° Celsius.

#### HC SR501 Functional Description

When someone comes under infrared waves regions, then sensor detect variations in infrared waves it trip alarm and tell about someone appearance at that point. We can its adjustment according to our requirements.

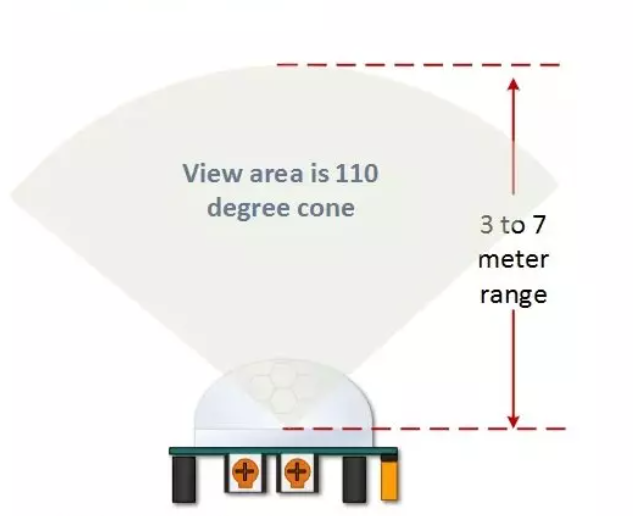
* HC-SR501 Initialization:

When we start function on it, it require a minute for the initiate. During this period, it does not work properly. During this period for it to work properly, we need a circuit or controller to take this initialization period into consideration.

* HC-SR501 Area of Detection:

In the coming line, we will discuss about its detection area.

It works in the range of 110 degree cone area and 3 to 7 meters.



#### Conclusion

The HC-SR501 Motion Sensor is a versatile and cost-effective component, widely employed in security systems, lighting control, and various motion-sensing applications due to its simplicity, reliability, and adjustable parameters.

## Software

### Arduino IDE

Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules.

It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

It is available for all operating systems i.e. MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing and compiling the code.

A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.

Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

This environment supports both C and C++ languages.

### ESP RainMaker

The ESP RainMaker is a collection of device-agent SDK, a transparent cloud application and iOS/Android phone apps. This lets developers write the firmware and instantaneously interact with it through the phone-apps, or through other services.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động

ESP RainMaker app offers the following :

* Signing up for the ESP RainMaker account
* Provisioning ESP RainMaker devices
* Controlling and Managing the devices remotely or over local network
* Auto rendering of the UI based on the device descriptions in the firmware
* Scheduling to allow actions to be triggered on devices at a given time on a specified day or days of the week
* Creating abstract or logical grouping of devices
* Sharing to allow controlling of devices to other users

# RELATED WORK/LITERATURE REVIEW

# METHODOLOGY

In order to develop a flexible and cost-effective security system using ESP32 and HC-SR50, we adopted an approach based on three stages, namely, planning, conducting, and reporting.

### Primary studies

To increase the emergence of search results, primary studies were selected based on keywords in the search function of a publication or a search engine. The search strings are as follows:

("Internet of things" OR "IoT") AND "Security"  
("Internet of things" OR "IoT") AND ("ESP32" AND "HC-SR501")  
("Internet of things" OR "IoT") AND "Threats"

We will consider the following publishers online platforms for the papers retained for this Literature Review: IEEE Xplore, Google Scholar, Sunfounder, Electronics Hub, ACM Digital Library.

Based on the above mentioned, we thoroughly searched these platforms for papers that comply with them in their titles or abstracts or keywords.

# CODE ANALYSIS

Function: **sysProvEvent()**

* This function is called whenever an event occurs related to the provisioning process.
* There are two types of events that can occur:
  + ARDUINO\_EVENT\_PROV\_START: This event is raised when the provisioning process starts.
  + ARDUINO\_EVENT\_WIFI\_STA\_CONNECTED: This event is raised when the device connects to the Wi-Fi network.
* The sysProvEvent function simply prints a message to the serial monitor indicating what event has occurred.

Function: **write\_callback()**

* This function is called whenever a value is written to a parameter of a device that has been registered with ESP RainMaker.
* The function receives the following parameters:
  + device: The device to which the value was written.
  + param: The parameter to which the value was written.
  + val: The value that was written to the parameter.
  + priv\_data: A pointer to any private data that was associated with the device when it was registered.
  + ctx: A pointer to the write context.
* The write\_callback function simply prints a message to the serial monitor indicating the device, parameter, and value that were updated.

Function: **setup()**

* This function is called once at the beginning of the program.
* The function performs the following tasks:
  + Initializes the serial monitor.
  + Sets the pin modes for the RESET\_PIN, WIFI\_LED, PIR\_PIN, and BUZZER\_PIN.
  + Creates a new node called my\_node.
  + Adds the SecuritySwitch device to the my\_node.
  + Enables OTA (Over-the-Air) updates.
  + Set the time zone.
  + Prints the chip ID and service name to the serial monitor.
  + Starts ESP RainMaker.
  + Registers the sysProvEvent function to be called whenever a provisioning event occurs.
  + Starts the provisioning process.

Function: **loop()**

* This function is called repeatedly after the setup function has finished.
* The function performs the following tasks:
  + Reads the state of the RESET\_PIN. If the pin is LOW, it means that the reset button has been pressed. The function then calls the RMakerFactoryReset function to reset the device to factory defaults.
  + Check the status of the Wi-Fi connection. If the device is not connected to Wi-Fi, it turns off the WIFI\_LED. Otherwise, it turns on the WIFI\_LED.
  + Calls the detectMotion function to detect motion.
  + Calls the controlBuzzer function to control the buzzer.

Function: **detectMotion()**

* This function checks the state of the PIR\_PIN to see if motion has been detected.
* If motion has been detected, the function prints a message to the serial monitor, raises an alert, turns on the buzzer, and sets the BUZZER\_STATE variable to true.

Function: **controlBuzzer()**

* This function checks the BUZZER\_STATE variable to see if the buzzer is supposed to be turned on.
* If the BUZZER\_STATE variable is true, the function checks to see if 5 seconds have elapsed since the buzzer was turned on. If it has, the function turns off the buzzer and sets the BUZZER\_STATE variable to false.

# CONCLUSIONS

## Evaluate

### Advantages

* There are many spending management applications on the market for my team to refer to, thereby helping my team have many ideas in the process of designing models and displaying data.
* Documents on project implementation tools are available in full and detailed information online. The community helps solve common problems when using project-related technology quickly.

### Disadvantages

* It takes a lot of time to get used to the new tool and how it works.
* Arrange each member's working time to hold meetings. During the process, there were disagreements.

## Conclusion

### Result

After researching and implementing the project, the group has achieved the following results:

* Successfully built a sensor system with the following functions::
* Sensing a moving object within range and sending a notification to the application
* Users can perform manual shutdown
* Perform partial testing of functions
* Students have learned how to make a system with IOT devices.
* Through the project, the group learned more about new technologies for embedded programming, data transmission, functional code for each pin of the device, etc. to serve the group's future work.
* Improve programming, teamwork and problem solving skills.

### Advantage

* The interface is simple, bright, and users can easily follow the application.
* Fast response speed to operations, priority of model performance.
* Can make a useful contribution to life.

### Defect

* The remaining model has a single, incomplete basis.
* A broader model that can link multiple devices and have more coverage has not yet been built

## Development orientation

* Expand the model and be more optimized, ....
* Optimize code for easy updates and feature expansion.

# REFERENCES

The team used reference materials from the Arduino.vn homepage. Besides, our team also references many discussion topics, illustrative examples about HC-SR501, code, applications on websites such as Stackoverflow, Github, ...

<https://www.espressif.com/en/products/socs/esp32>

[https://www.youtube.com/ESP32](https://youtu.be/MMtYchNrcL4?si=D8O2q89mxBOw0gUC)

[https://www.youtube.com/HC-SR501](https://youtu.be/yq-rur7rbRY?si=_T34kPYLMEFhtE4b)

<https://github.com/espressif/arduino-esp32>