



**Master thesis of Paris-Saclay University**

**Thesis supervision**

**Committee**

Arnaud BOURNEL Chairman

*Université Paris-Saclay*

Pierre DUHAMEL Reporter

*CNRS, Laboratory of Signals and Systems*

NGUYEN Linh Trung Examiner

*VNU University of Engineering and Technology*

BUI Duy Hieu Examiner

*VNU Information Technology Institute*

Erwan LIBESSARTExaminer

*CentraleSupélec*

BUI Duy Hieu Thesis supervisor

*VNU Information Technology Institute*

TRAN Thi Thuy Quynh Co-supervisor of the thesis

*VNU University of Engineering and Technology*

Erwan LIBESSARTCo-supervisor of the thesis

*CentraleSupélec*

DEVELOPING A REAL-TIME OBJECT DETECTION SYSTEM ON FPGA

Specialization: M2 Communication and Data Engineering

**Master student: NGUYEN Trung Kien**

# ABSTRACT

# AUTHORSHIP

|  |  |  |
| --- | --- | --- |
|  | | *Hanoi*, January 22nd, 2024 |
|  | | Author |
|  | |  |
|  |

# ACKNOWLEDGEMENT

Contents

[ABSTRACT i](#_Toc201398440)

[AUTHORSHIP ii](#_Toc201398441)

[ACKNOWLEDGEMENT iii](#_Toc201398442)

[List of Abbreviations vi](#_Toc201398443)

[List of Figures viii](#_Toc201398444)

[List of Tables ix](#_Toc201398445)

[Introduction x](#_Toc201398446)

[Chapter 1. Real-time Object Detection System 1](#_Toc201398447)

[1.1. Introduction to the Internet of Things (IoT) and Its Applications for Fire Monitoring Alert Systems 1](#_Toc201398448)

[1.2. Current situation and Limitations of the monitoring fire alert system 2](#_Toc201398449)

[1.3. Related Works 3](#_Toc201398450)

[1.4. Conclusions 5](#_Toc201398451)

[Chapter 2. System Architecture Design 6](#_Toc201398452)

[2.1. Hardware System Architecture 7](#_Toc201398453)

[2.1.1. Sensor Node Architecture 7](#_Toc201398454)

[2.1.2. Central Processing Unit System Architecture 11](#_Toc201398455)

[2.2. Software & Firmware System Architecture 13](#_Toc201398456)

[2.2.1. Firmware at the Node 14](#_Toc201398457)

[2.2.2. Software at the Central Processing Unit 20](#_Toc201398458)

[2.3. 25](#_Toc201398459)

[2.4. Conclusions 25](#_Toc201398460)

[Chapter 3. Implementation and Evaluations 26](#_Toc201398461)

[3.1. Experiment setup environment 26](#_Toc201398462)

[3.1.1. Mean Average Precision 26](#_Toc201398463)

[3.1.2. Frame rate 26](#_Toc201398464)

[3.2. Experimental results 27](#_Toc201398465)

[3.2.1. Input: PETS09-S2L1 [19] 27](#_Toc201398466)

[3.2.2. Input: TUD-Stadtmitte [20] 27](#_Toc201398467)

[3.2.3. Input: TUD-Campus [21] 27](#_Toc201398468)

[3.3. Conclusions 28](#_Toc201398469)

[Conclusions and Perspective 28](#_Toc201398470)

[References 29](#_Toc201398471)

# List of Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Definition |
| IoT | Internet of Things |
| ADAS | Advanced Driver Assistance Systems |
| AI | Artificial Intelligence |
| AMD | Advanced Micro Devices |
| mAP | Mean Average Precision |
| ASIC | Application Specific Integrated Circuit |
| AVC | Advanced Video Coding |
| CBAM | Convolutional Block Attention Module |
| CNN | Convolutional Neural Network |
| COCO | Common Objects in Context |
| CPU | Central Processing Unit |
| DMA | Direct Memory Access |
| FPGA | Field Programmable Gate Array |
| FPS | Frames Per Second |
| GB | Gigabyte |
| GE | Gate equivalents |
| GPU | Graphics Processing Unit |
| HD | High Definition |
| HOG | Histogram of Oriented Gradients |
| LTS | Long Term Support |
| MJPEG | Motion JPEG |
| MOT15 | Multiple Object Tracking Benchmark 2015 |
| MOTC | Multiple Object Tracking Challenge |
| MPSoC | MultiProcessor System On Chip |
| NMS | Non-Maximum Suppression |
| OS | Operating System |
| PASCAL | Pattern Analysis Statistical Modelling and Computational Learning |
| PETS09 | Performance Evaluation of Tracking and Surveillance 2009 |
| PTZ | Pan-Tilt-Zoom |
| RAM | Random Access Memory |
| RCNN | Regional Convolutional Neural Network |
| RGB | Red Green Blue |
| SIFT | Scale-Invariant Feature Transform |
| SPP | Spatial Pyramid Pooling |
| SRAM | Static Random Access Memory |
| SSD | Single Shot Detector |
| SVM | Support Vector Machine |
| TSMC | Taiwan Semiconductor Manufacturing Company |
| UHD | Ultra High Definition |
| VNU | Vietnam National University |
| VOC | Visual Object Classes |
| YOLO | You Only Look Once |
| ZCU106 | Zynq UltraScale+ ZCU106 Evaluation Kit |

# List of Figures

[Figure 1. An IoT system with interconnected smart devices. 2](#_Toc201223746)

[Figure 2. Flame sensor PISAFE. 8](#_Toc201223747)

[Figure 3. Flame sensor KY-026. 8](#_Toc201223748)

[Figure 4: Honeywell 5809SS heat detector. 9](#_Toc201223749)

[Figure 5: DHT11 - Temperatue & Humidity Sensor. 9](#_Toc201223750)

[Figure 6: ESP32 Microcontroller. 10](#_Toc201223751)

[Figure 7: STM32F103C8T6 Microcontroller. 11](#_Toc201223752)

[Figure 8: HC-12 Wireless Communication Module. 12](#_Toc201223753)

[Figure 9: Single board Computer - Raspberry Pi 5. 13](#_Toc201223754)

[Figure 12: Touchscreen Monitoring. 14](#_Toc201223755)

[Figure 13: NoName A Touchscreen. 14](#_Toc201223756)

[Figure 14 - Decomposition of a vector into the form of two vectors [18]. **Error! Bookmark not defined.**](#_Toc201223757)

# List of Tables

[Table 1 - Sine and Cosine quantized value. **Error! Bookmark not defined.**](#_Toc157527576)

[Table 2 - Truth Table Confusion Matrix. **Error! Bookmark not defined.**](#_Toc157527577)

# Introduction

# Real-time Object Detection System

This chapter provides an overview of the application of Internet of Things (IoT) technology in monitoring and alerting fire detection systems, primarily in the context of increasingly serious fire problems that require critical attention. It begins with an introduction to IoT and the importance of its role in improving monitoring skills and its value in fire situations. In addition, this chapter also focuses on the main objective of the thesis: to explore and employ the full potential of IoT-connected hardware devices in both IoT applications and real-world scenarios. This chapter is organized into three sections. Section 1.1 provides an overview of the IoT and its applications in fire monitoring systems. [*Section 1.2*](#Section1_2) discusses the current situation and limitations of existing fire monitoring systems. [*Section 1.3*](#Section1_3) presents modern fire detection methods using current technologies, along with their limitations.

## Introduction to the Internet of Things (IoT) and Its Applications for Fire Monitoring Alert Systems

In recent years, fire accidents have become increasingly dangerous, causing critical damage to both human life and property. Traditional fire monitoring systems, which often rely on manual checks or local alerts, are limited in their ability to respond quickly and effectively, especially in complex or remote environments. To handle this issue, we need smart and effective solutions to improve it in time.

The IoT is one exciting technical development that resolves these issues. As its name suggests, IoT refers to a network of smart devices that can communicate and connect with each other, exchanging data. These devices can collect, transmit, and process data in real-time, thereby enabling automation and making personal decisions.

Unlike standalone devices that work independently, a true IoT system (Figure 1) is a scalable and integrated system where multiple devices work together for a common goal. In the context of fire detection, this goal is always monitoring environmental conditions, identifying early signs of fire, and sending real-time alerts, even when no one is present at that location.

By providing features such as remote monitoring, automated notifications, and log data, IoT-based systems significantly enhance responsiveness and overall safety. Therefore, using IoT for fire monitoring systems is both useful and essential for enhancing early detection and minimizing damage.

[](equipment-services.blogspot.com)

Figure . An IoT system with interconnected smart devices.

## Current situation and Limitations of the monitoring fire alert system

Fire accidents are now common and can occur in various types of environments, ranging from crowded residential areas to industrial zones, forests, etc.... In an industrial environment, the use of flammable materials, high electricity consumption, and poor ventilation are common reasons that increase the risk of fire. In a residential area, the complex and overloaded electrical network increases the chance of short circuits and electrical fires. In a forest environment, rising global temperatures make it more flammable, and also human activities, such as camping or purposeful burning of the forest, increase the risk of fire. As mentioned, the fire can occur anywhere, any time, whether in crowded urban areas or isolated natural areas.

Therefore, deploying fire monitoring systems with sensors and data-collecting modules in large areas is expensive and challenging. Traditional systems, which rely on wired connections or require continuous human monitoring, are not realistic, especially in large areas or remote and harsh environments. These systems will face problems such as processing complex data and false fire alerts. This is really a big challenge for the fire warning system. The solution using IoT with sensor data can help provide more accurate and useful results.

Currently, traditional fire monitoring and alert systems are still widely used and have not been fully replaced by IoT-based systems. Although some improvements have been made, many issues and limitations still exist, such as detection delays, limited coverage range, and high maintenance requirements. As a result, developing a fire detection and monitoring system that uses fire, temperature, and humidity sensors combined with wireless communication technologies is a creative, possible, worthy solution to consider.

## Related Works

The development of fire monitoring and alert systems based on IoT has gained increasing attention in recent years. This section presents several existing works and discusses their limitations, highlighting the gap that this project aims to address.

Nowadays, IoT systems typically use various sensors, ranging from tiny microcontrollers to large ones, and are on the way to merging machine learning in the new era of AI, which is now very popular. I plan to use AI in my future work. Back to this section – “Related Works”, I believe that anyone who reads this thesis someday will find it easy to refer to the paper I will share below:

In the paper [1], the authors proposed an IoT-based fire detection and monitoring system using temperature, flame, and smoke sensors with an ESP32 microcontroller, and implemented a LoRa Network to detect and alert on forest and farm. In paper [2], the authors have used an Arduino-based home fire alarm system with a GSM module and temperature sensor to detect fire and send SMS alerts, enhancing user safety and property protection. In the paper [3], the authors developed a smart IoT-based system using the same list of sensors as paper 1 to detect fire, but in addition, using Arduino UNO and a gas sensor. The system was implemented on the Blynk platform and uses GSM to send alerts. Hơi cụt)Paper [3], the authors designed an STM32-based wireless fire detection system using a block of sensors and embedded wireless communication technologies like Wi-Fi to detect fire. Next paper [4] (2019) shows a big improvement in the application of robots. Pages 267–278 of the paper focus on the use of robots in fire monitoring and alert systems. The project utilizes STM32 and STC89C51 microcontrollers, drones, to connect air and ground monitoring systems and facilitate communication through ZigBee. This paper [5] (2021) focuses more on the alerting part. The authors use ESP32 and PIR sensors to detect fire and movement, sending alerts through a Telegram bot to a smartphone, with feature alarms by image and temperature parameters. When AI becomes popular, some papers nowadays are applying it in IoT systems to detect and alert fire. For example, paper [6] (2023) presents a system that combines IoT devices with YOLOv5 to enable early forest fire and real-time detection, aiming to reduce false alarms and enhance safety during dry seasons. While the approach in [6] is already technically advanced, [7](2025) adds a suggestive improvement to fire monitoring and alert systems by using IoT and machine learning, and also combining various sensor types, to improve detection accuracy and support remote monitoring. Although this recent work represents the development, the core concept of fire detection systems has already been explored in earlier studies.. Indeed, a paper from 2013 [9] discusses a real-time fire alarm system developed using Raspberry Pi and Arduino Uno. This system features include smoke detection, room image capture, and alert through SMS and a web interface. It only sends alerts to firefighters after user confirmation to reduce false alarms. Although this paper was from 2013, its implementation was quite complete – features like login were already added. However, the web design is still basic and a little bit outdated. However, about implementation, it was a meaningful success, and the authors’ idea was forward–thinking.

As mentioned at the beginning of Section 1.3, this thesis does not apply AI in the IoT system. However, this will be explored in future work. To recognize and expand on the ideas presented from the science papers, many of which use wireless technologies such as Wi-Fi or GSM. This thesis proposes enhancing current approaches by applying radio communication to an IoT-based fire detection and alert system. Once again, I am grateful to these “giants” whose pioneering work has allowed this small idea of a thesis to be developed.

## Conclusions

An IoT system is an essential technology with applications in monitoring and fire detection. Achieving real-time monitoring, high accuracy, processing all scenarios, and overcoming hardware limitations creates serious difficulties. The use of IoT brings a promising solution, enabling real-time monitoring, remote access, and automated alert notifications. This project aims to inherit and integrate the ideas from those previous works, while also contributing further improvements to develop a more responsive and reliable fire alert system. It also introduces new innovations to improve the system’s flexibility and performance in real-life situations. In Chapter 2 *System Architecture Design*, a proposal will be presented, focusing on the integration of hardware and software components, including sensor modules, communication systems, and control logic. All of which will be implemented through software to build the proposed IoT-based fire monitoring solution.

# System Architecture Design

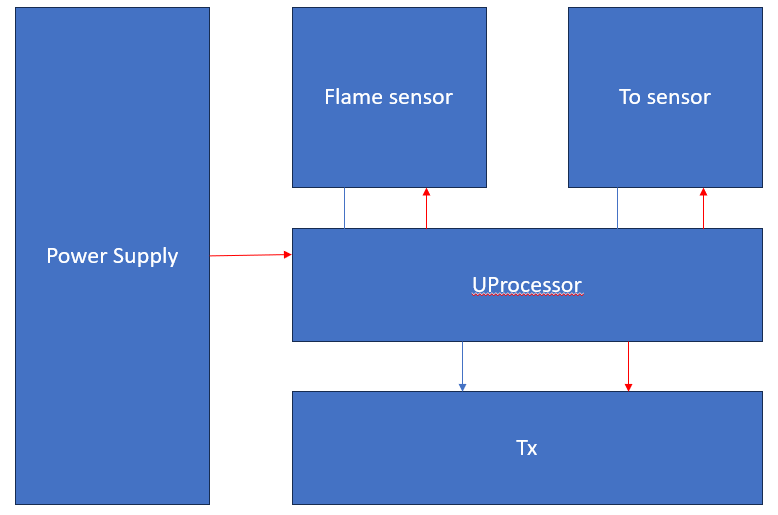
The first chapter of this thesis provided an overview of IoT technology and its applications in fire detection and monitoring. It also discussed the current state of fire accidents occurring in many places today, highlighting the need for more efficient solutions. Furthermore, with the current state of fire, the thesis proposed a new direction by enhancing existing ideas with the use of radio communication. While many previous systems deploy on Wi-Fi or GSM, this thesis focuses on using *low-cost, low-power* communication wireless modules. The main goal is to develop an IoT-based system that monitors, detects fires, and alerts in real-time.

Building on this ideal, *Chapter 2* presents the proposed system architecture, which consists of both hardware and software components designed to work together for efficient fire monitoring. The system includes: Raspberry Pi and STM32 microcontrollers, HC-12 modules for communication, and a variety of sensors to collect fire-related data. This chapter has three main sections: *Section 2.1 Hardware System Architecture*: in this section, an overview of the hardware components used in this thesis will be presented, including sensors, microcontrollers, and the central processing unit. Each component will be explained with the reason why it is used and a summary of its advantages and disadvantages.*Section 2.2: Software & Firmware System Architecture*: This part details how the software & firmware work across the system. It begins with sensor-side firmware, which includes collecting data, encoding, and transmission. On the server side, it explains how data is received, decoded, stored, and visualized through a Flask-based web application. This section also includes user authentication, account management, and how software sends alerts (using Fuzzy Logic and Threshold Evaluation). *Section 2.3 Conclusions:*This final section summarizes all components introduced in *Chapter 2* and prepares for the system implementation and evaluation that will follow in later chapters, *Chapter 3: Implementation Results and Evaluation*.

## Hardware System Architecture

This section introduces the hardware components used at both the *sensor node* and *the central processing unit*. It explains what each component is, its role in the system, and the reasons behind choosing it for the demo. Subsections cover specific components, including fire sensors, temperature and humidity modules, STM32 for edge processing, and the Raspberry Pi for central control and web server hosting. The following *section 2.2* will discuss the data flow in depth, and how each part works together.

Kiến trúc phần cứng



### Sensor Node Architecture

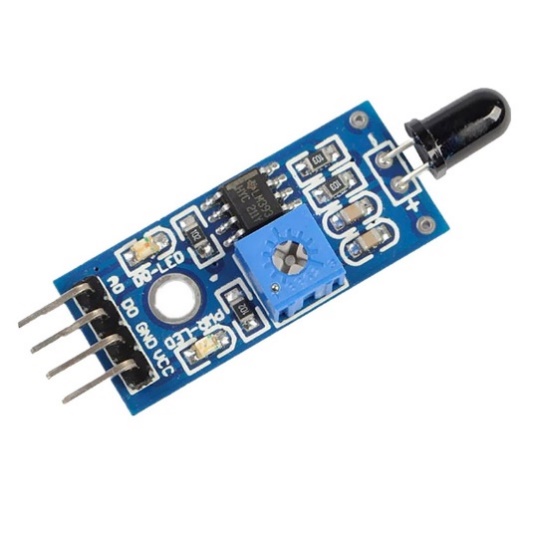
#### Flame Sensor using KY—026

Flame sensor là gì và tại sao tôi chọn KY-026

Flame sensors are a type of detector used to identify the presence of fire or flames, which is possible because fires typically occurs various types of radiation – including *visible light* (such as red, orange, and blue), *infrared radiation* (heat), and maybe *ultraviolet radiation* – and flame sensors are specifically designed to detect the infrared spectrum in the 760 nm to 1100 nm range.

In Figure 2, the PISAFE (Figure 2) flame sensor is a solution for detecting fire using Wi-Fi connectivity and wide-area fire detection (up to 100 meters), while also meeting fire safety certification standards. However, the cost of it is really high, so this sensor will not be used in the context of this thesis.

Instead, to align with the project’s budget and for demo purposes, the **KY-026** flame sensor (Figure 3) is selected. Despite its small size and lower cost, the KY-026 is a good option for demo and educational applications for flame detection in this thesis.

[](https://store.arrowdot.io/product/ky-026-flame-sensor-module-detects-infrared-light-emitted-2/)  
Figure . Flame sensor KY-026.

#### Temperature & Humidity Sensor using ….

Cái này là gì và tại sao lại chọn cái kia

A temperature and humidity sensor is a type of sensor used to collect environmental data. When a fire occurs, such as when flammable materials or electrical wires are burning, the temperature of the fire will increase continuously, and the humidity drops quickly. In the process of fire detection and monitoring, just using the flame sensor’s evaluation is not enough. Therefore, combining data from both the Temperature & Humidity Sensor and the Flame Sensor can provide a more accurate result and enhance the fire detection system's reliability, helping to reduce false and prevent incorrect fire alarms. Nowadays, there are many types of sensors collecting environmental data, like Honeywell 5809SS [9] (Figure 4). Similar to the flame sensor, this system is designed to be low-power and low-cost, and as the main objective of the thesis is for demo purposes. Therefore, it uses the DHT11 (Figure 5) sensor to collect temperature and humidity data. The DHT11 is a widely used and accessible sensor in educational environments, especially in university-level projects.

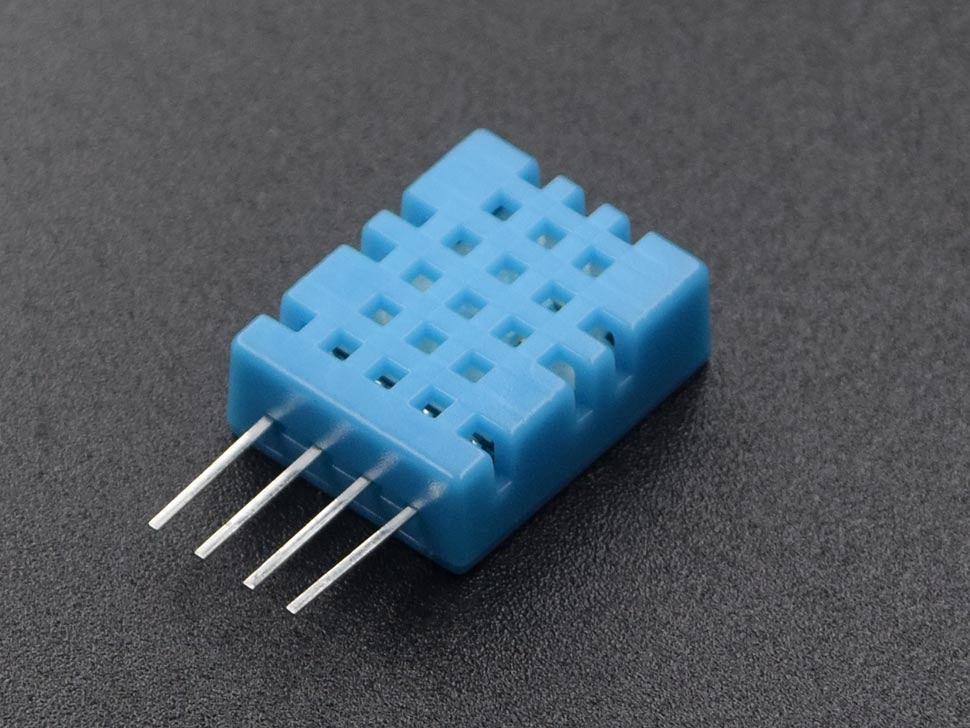
[](https://circuit.rocks/products/sensors-temperature-and-humidity-dht11-sensor-html)

Figure : DHT11 - Temperatue & Humidity Sensor.

#### Microprocessor using STM32F103C8t6

Nó là gì tại sao lại chọn cái này

In an IoT system, the edge processor is a lightweight microcontroller responsible for collecting data from a block of sensors, such as flame sensors, temperature and humidity sensors. Each microcontroller that handles the data from one or more sensors can call a sensor node within the system.

Usually, edge processors use microcontrollers like the ESP32 (Figure 6), ESP8266, or other microcontrollers, which are favored because of its programming and integration. However, in this thesis, the selected edge processor is the STM32, specifically the STM32F103C8T6 (Figure 7). Although programming the STM32 may be more complex compared to ESP-based microcontrollers, but it is chosen to support the research and application of this work. Additionally, STM32’s UART communication mode is effective for serial data transmission, making it an right choice for handling sensor data and transmitt data in this system.

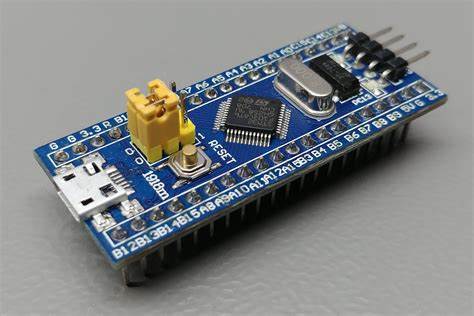
[](https://www.epanorama.net/blog/2022/06/18/introduction-to-the-stm32-blue-pill-stm32duino-and-other-stm32-boards/)

Figure : STM32F103C8T6 Microcontroller.

#### Wireless Communication Module using HC-12

Tương tự

This module plays an important role in the main idea of this thesis. In the proposed IoT system, the wireless communication module is the part that transmits data between sensor nodes and the central processing unit. Several types of wireless communication technologies were reviewed in *Section 1.3 Related Work*, including Wi-Fi, Zigbee, Cellular, and LoRa.

Among them, Wi-Fi, Zigbee, and Cellular require an existing communication infrastructure such as routers, gateways, or mobile network towers. On the other hand, LoRa can operate in two modes: LoRaWAN, which is based on centralized infrastructure (gateways and network servers), and LoRa (peer-to-peer), which can communicate directly between nodes without any infrastructure.

Despite the advantages of LoRa in terms of long-range communication and energy efficiency, it often comes at a higher cost and requires more complex configuration. In opposition, the HC-12 module offers a low price and is easy to configure, making it suitable for the goals and scope of this thesis project.In the *2.1.1 Node System Architecture*, the HC-12 module is placed in the transmission (Tx) block. After the sensor data is collected and processed by the STM32F103C8T6 microcontroller, the resulting data packet is transmitted to the HC-12 module (Figure 8), which then sends the data wirelessly via radio frequency (RF) communication.

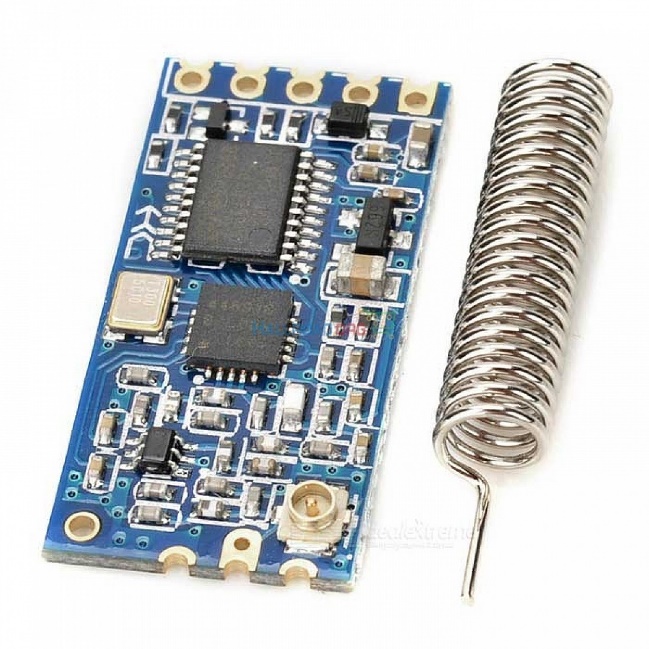
[](https://www.flyrobo.in/hc-12-433-si4463-wireless-serial-module)

Figure : HC-12 Wireless Communication Module.

### Central Processing Unit System Architecture

#### Wireless Communication Module using …

In the *2.1.2 Central Processing Unit System Architecture*, the HC-12 module (Figure 8) is placed in the receiver (Rx) block. After the data is transmitted from the node to the central processing unit through radio communication, the HC-12 module at the receiver receives incoming signals and transmits them to the central processor for handling.

#### Central processing unit using ….

In this section, the central processing unit is using for handling incoming data after it has been wirelessly transmitted to the receiver block (in this case, the HC-12 module). There are many types of CPUs suited of performing logical data processing and running a web server. However, with the objective of this thesis, the system using the Raspberry Pi 5 (Figure 9) to explore its practical application in real-world IoT scenarios.

The Raspberry Pi 5 is a single-board computer (SBC) — as the name implies, it is a complete computer built on a single circuit board. With only a power supply and an external monitor, it can working as a fully like a computer. It supports many type of operating systems like : Raspberry Pi OS, Raspberry Pi Lite, Ubuntu,…etc…, among which Pi OS being the most commonly used.

For storage, it uses a microSD card, supporting capacities up to 1 TB, however a normal and reliable setup runs from 32 to 256 GB. The Pi 5 is powered by a quad-core ARM Cortex-A76 processor, provides high performance for embedded applications. In this thesis, the Raspberry Pi 5 equipped 8 GB of RAM, making it the same to many standard desktop PCs.

About connectivity, it provides USB ports, Ethernet, Wi-Fi, Bluetooth, a 3.5 mm audio jack, camera ports and also 2 micro HDMI for connecting screen and support to 4k HD. Importantly, it also features a 40-pin GPIO header, including 26 programmable GPIO pins, with the rest used for power (3.3V/5V - VCC) and ground connections (GND). These GPIO pins allow the Pi to connect with sensors, LEDs, buzzers, and communication modules like the HC-12.

With its wide features, fast performance, and flexibility, the Raspberry Pi 5 is an excellent choice for developing modern IoT systems.

[](https://www.theregister.com/2023/09/28/raspberry_pi_5_revealed/?td=amp-keepreading)

Figure : Single board Computer - Raspberry Pi 5.

#### Touchscreen Monitor using …….

In a typical IoT-based monitoring system, external touchscreen displays are often connected to the Central Controller Unit (Raspberry Pi 5), especially when building a web-based application. These interfaces enable users to easily interact with the system and monitor its status in real time (Figure 10).

Although this project does not focus on physical user interfaces, adding a touchscreen in the demo setup improves system visualization and user interaction.

Therefore, a touchscreen NoName A (Figure 11) is used. It works well with the custom monitoring software in this project. This addition helps users understand the system more clearly, test it more easily, and consider improvements for future development.



Figure : Touchscreen.

## Software & Firmware System Architecture

Section 2.1 focuses on introducing the hardware components used in this thesis, along with the reasons for their selection and their unique benefits. *Section 2.2. Software & Firmware System Architecture* discusses the working flow of the full system implementation. *Section 2.2* is divided into two subsections: *Section 2.2.1. Firmware at the Node* explains how data is handled at the sensor node after collecting data from flame, temperature, and humidity sensors. It also provides the standard requirements for protecting the privacy and security of data during wireless transmission. *Software at the Central Processing Unit* uses software to receive, decode, and process incoming data packets. The decrypted data will be uploaded in real-time onto a web-based platform. This subsection also covers user administration, data logging, notification handling, and other system operations, which will be described in further depth.

### Firmware at the Node

#### Sensor Data Collection

**Input**: Value from the environment.

**Output**: Temperatue, Humidity, Fire state data.

During the Sensor Data Collection stage, environmental data is collected using two sensors: the DHT11 for temperature and humidity and the KY-026 for fire detection.

Algorithm : Read DHT11 Data and Fire Sensor State.



The DHT11 employs a 1-Wire interface, with the STM32 initiating communication through GPIOA Pin 6 set as output with accurate time delays. After setting up, it returns 5 bytes (humidity, temperature, and checksum), and data validity is validated by checking the checksum. The KY-026 flame sensor, which is attached to GPIOA Pin 1 (digital) and, alternatively, GPIOA Pin 0 (analog via ADC), produces HIGH (1) when no flame is detected and LOW (0) when there is a fire. STM32CubeMX, STM32CubeIDE, and STM32CubeProgrammer handle sensor setup and firmware development for debugging and deployment.

#### Encoding Message

This is a highly important and interesting part when transmitting messages over wireless communication. It ensures compliance with standard requirements for protecting privacy and securing data in wireless transmission. In wireless data security, the main components typically include (Figure 12): Frame Packing, Security Coding, Channel Coding, Spreading, and Synchronization. In the Encoding Message section, each of these components will be described in detail through corresponding steps, allowing for a clear understanding of how the message is processed and prepared for transmission.



Figure : Data protection sequence in wireless transmission.

#### Frame Packing

**Input:** Temperature, Humidity, Fire state data.

**Output:** MavLink frame.

In the Frame Packing step of the thesis, sensor data – including temperature, humidity, and fire state – is packed into a standardized communication format using the MAVLink protocol (version 1). This protocol, widely used in embedded and wireless systems, ensures reliable and efficient data exchange. Each MAVLink frame comprises three sections: a 6-byte Header, a Payload of up to 255 bytes, and a 2-byte CRC.

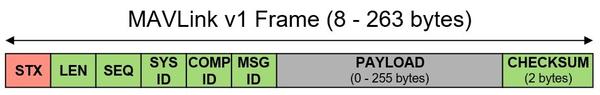


Figure 13: Mavlink v1 Frame.

As in Figure 13, the Header includes the start byte (0xFE), payload length, a message sequence number, system ID, component ID, and message ID. The Payload carries the original sensor values (in hexadecimal format), while the CRC is used to verify data integrity and detect transmission errors. This structure enables robust, consistent communication between nodes in the fire monitoring system.

#### Security Coding (AES 256)

**Input:** MavLink frame**.**

**Output:** Ciphertext.

In the Security Coding step, to enhance the protection of MAVLink messages, this thesis applies the *AES-256* block encryption algorithm, one of the most powerful and widely adopted encryption standards today. With a 256-bit key length and 14 rounds of transformation, *AES-256* provides strong resistance against modern cryptographic attacks. Because *AES-256* works with 16-byte blocks, the MAVLink message must be padded accordingly. The *PKCS7* padding is used: if the message length is not a multiple of 16, the missing bytes are filled with the number of padding bytes (e.g., if 3 bytes are missing, three bytes of 0x03 are added); if the message is already a multiple of 16, a full block of padding with the value 0x10 is appended. After padding, the message is divided into fixed-length blocks and encrypted sequentially with AES-256. Additionally, to further confuse the data, this thesis uses the bit masking technique – specifically, bit reversal, which reverses all the bits in each block.

#### Channel Coding (LDPC 1/2)

**Input:** Ciphertext.

**Output:** Channel-encoded message.

To enhance reliability during the wireless transmission of encrypted data, this thesis employs channel coding using the Low-Density Parity-Check (LDPC) method in 1/2 mode. LDPC is an error-correcting code that uses a sparse parity-check matrix (H) shared between the transmitter and receiver. In this implementation, each data byte is paired with a parity byte, effectively doubling the length of every block. This byte-pairing methodology enables localized error detection and correction, as each pair can be independently verified. This method makes LDPC highly effective in identifying and correcting transmission errors caused by external interference.

#### Spreading

**Input:** Channel-encoded message.

**Output:** Spread-spectrum signal.

In the Spreading phase, this thesis uses a Fibonacci-type Linear Feedback Shift Register (LFSR) to enhance communication safety and reliability. By generating pseudo-random sequences, the LFSR confuses the signal, making it more difficult to track, detect, and attack during wireless transmission. This step receives an LDPC-encoded message whose length has been doubled by channel coding. Each data block is XORed with the LFSR output, and the same register is shared between the transmitter and receiver for detection purposes.

#### Synchronization

**Input:** Channel-encoded message.

**Output:** Framed message.

After undergoing the processes of frame packing, encryption, channel coding, and spreading, the message becomes significantly confused, making it difficult for the receiver to identify the beginning or structure of the transmission. Therefore, as the final step, two header bytes (0xEB and 0x90) are added after LFSR encoding.

These specific bytes were chosen because 0xEB and 0x90 are commonly detectable in wireless environments. They frequently appear in the radio spectrum, making them reliable markers. By adding these two bytes at the beginning of the final message, the receiver can more easily synchronize and detect the start of a valid transmission.

#### Wireless Transmission

**Input:** Framed message.

**Output:** Message transmitted.

In the Wireless Transmission stage, the framed message is transmitted over the air using the HC-12 wireless module. The STM32 sends the message through its TX pin (GPIOA9), which is connected to the RX pin of the HC-12, where modulation and radio-frequency transmission are handled. Before transmission, the module is configured using basic AT commands to set parameters such as baud rate and channel. Once configured, the HC-12 broadcasts the message wirelessly, completing the transmission process.

### Software at the Central Processing Unit

#### Wireless Receiver

**Input:** Message transmitted.

**Output:** Message received.

On the receiver side, the HC-12 wireless module is pre-configured with the same AT command parameters, such as baud rate and channel, as described in the Wireless Transmission section. After the message is successfully transmitted, the Central Processing Unit listens for incoming data through the USB UART.

#### Decoding Message

#### Synchronization Decode

**Input:** Message received.

**Output:** Frame message unpack.

When the message is received through USB UART at the Central Processing Unit, it checks the validity of the message by inspecting the first two bytes. If the first two bytes are 0xEB and 0x90, the message is valid. The synchronization decode step then unpacks the frame by removing these two header bytes, allowing the system to proceed with further processing.

#### Spreading Decode

**Input:** Frame message unpack.

**Output:** Despread message.

Because both the transmitter and the receiver use the same register after applying the LFSR Fibonacci, to decode the spreading, XOR each block again with the LFSR Fibonacci output, just as in the LFSR encoding step.

#### Channel Coding Decode

**Input:** Despread message.

**Output:** Channel decode message.

At this stage, LDPC decoding is employed to detect and correct errors caused by wireless transmission. As said in Spreading, a shared parity-check matrix H is applied to the received message to compute the syndrome. If the syndrome vector contains non-zero values, it indicates the presence of errors. The decoder then tries to correct the bit error by finding and flipping the bit related to the most incorrect parity. The syndrome is recalculated by using the calculate syndrome after each bit flip, and this process continues until the syndrome is zero or a maximum of ten iterations is reached. If problems continue after the allowed rounds, the message becomes uncorrectable and is destroyed.

#### Security Coding Decode

**Input:** Channel decode message.

**Output:** Plaintext.

After LDPC decoding, if the syndrome vector contains only zeros, the message is considered error-free and proceeds to the Security Coding Decode stage. As mentioned earlier, bit masking in this system involves a simple bit-reversal operation during encoding, so decoding applies bit reversal again to restore the original bit order, preparing the message for AES-256 decryption and PKCS7 unpadding.

In the AES-256 decryption phase, the same key used during encryption is reused, but the round keys are applied in reverse order, from the last to the first, across 14 decryption rounds. Once decryption is complete, the message enters the PKCS7 unpadding stage. Here, the decoder checks the last byte to determine the number of padding bytes added. For example, if the last byte is 0x03, the decoder verifies whether the final three bytes all equal 0x03. If so, they are removed; otherwise, the message is invalid. After successful unpadding, the original MAVLink frame is fully recovered.

#### Frame Packing Decode

**Input:** Plaintext.

**Output:** Temperature, Humidity, Fire state data.

In this final stage, the system verifies whether the plaintext corresponds to a valid MAVLink frame. If the first byte is 0xFE, it indicates the start of a MAVLink packet. The decoder then checks the LEN byte to confirm that the payload length matches the actual size, and verifies the SEQ byte to ensure message order – an important factor for logging and detecting missing or out-of-sequence packets. The SYS ID and COMP ID fields help identify which system and component sent the message. Most importantly, the CRC is recalculated from byte 2 onward (not including the start byte 0xFE) and compared with the received CRC. If they match, the frame is valid; otherwise, it is discarded. Once validated, the payload is extracted to obtain the final output: temperature, humidity, and fire state data.

a method where the primary gradient vector was split into two vectors, as shown in **Error! Reference source not found.**. The values of these two vectors for pixel I(x,y) were calculated using, which takes into account the values of the four surrounding pixels I(x+1,y), I(x-1,y), I(x,y+1), and I(x,y-1).

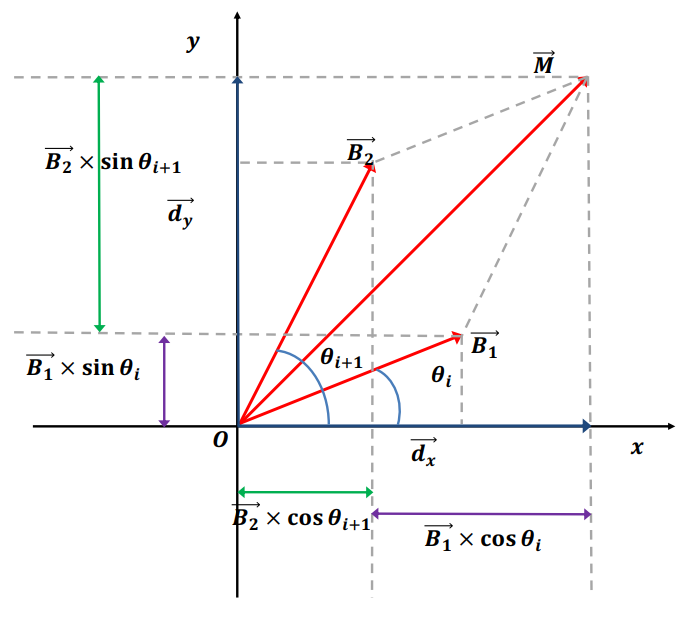


Figure 14 - Decomposition of a vector into the form of two vectors [18].

Two magnitudes are the solutions of the two equations:



This thesis uses this technique to improve the HOG-SVM algorithm and decrease the number of calculations needed to process each sliding window.

real-time applications with power constraints.

As mentioned above, the Object Detection Block will include Bilinear Interpolation Scale Generator Module, HOG-SVM module, combine module, and NMS module.

to form a block, and the cell histograms are normalized using block data.

##### Gradient calculation and Histogram generation

This

).

Algorithm 2 – Gradient calculation.

will generate all the histograms for the input image used for the next step.

Algorithm 3 – Histogram generation.

##### Descriptor generation and Result

After

generation.

The combining

remove redundancies. This block is crucial to this thesis's proposed object detection system.

## 

## Conclusions

In this

architecture offers a promising pipeline for efficient and accurate human detection for real-world applications.

# Implementation and Evaluations

In the previous

## Experiment setup environment

### Mean Average Precision

#### Mean Average Precision (mAP)

an object

#### Mean Average Precision (mAP)

Average

Here is a summary of the steps to calculate the AP:

1. *Generate the prediction scores using the model.*
2. *Convert the prediction scores to class labels.*
3. *Calculate the confusion matrix—TP, FP, TN, FN.*
4. *Calculate the precision and recall metrics.*
5. *Calculate the area under the precision-recall curve.*
6. *Measure the average precision.*

The mAP is calculated by finding Average Precision(AP) for each class and then averaging over several classes.



### Frame rate

## Experimental results

### Input: PETS09-S2L1 [19]

#### Input description

-S2L.

#### Result

and mAP graphs per frame of input PETS209-S2L.

#### Analysis

low mAP score.

### Input: TUD-Stadtmitte [20]

#### Input description

Input racteristic in the photo is that the moving object is only human moves in groups.

#### Result

In of 51

.

#### Analysis

The system can process images in near-real-time but faces significant challenges in accurately identifying and tracking human groups in urban pedestrian settings. The moderate precision combined with low recall and mAP scores suggests that the system might misidentify groups or miss them altogether.

### Input: TUD-Campus [21]

#### Input description

TUD-Campus.

#### Results

In the

D-Campus.

#### Analysis

The system

fast.

## Conclusions

# Conclusions and Perspective

# References

|  |  |
| --- | --- |
| [1] | Pradeep, Aneesh; Latifov, Akmaljon; Yodgorov, Ahborkhuja; Mahkamjonkhojizoda, Nurislombek, "Hazard Detection using custom ESP32 Microcontroller and LoRa," *IEEE,* no. 26 May 2023, 2023. |
| [2] | N. I. M. E. N. M. Z. a. K. S. S. K. M. N. N N Mahzan, "Design of an Arduino-based home fire alarm system with GSM module," *Journal of Physics: Conference Series,* vol. 1019, 2917. |
| [3] | N. I. M. E. N. M. Z. a. K. S. S. K. M. N. N N Mahzan, "Design of an Arduino-based home fire alarm system with GSM module," *Journal of Physics: Conference Series,* vol. 1019, 2017. |
| [4] | H. Y. ·. J. L. ·. L. L. ·. Z. J. ·. Y. L. ·. D. Zhou, "Development of Four Rotor Fire Extinguishing System for Synchronized Monitoring of Air and Ground for Fire Fighting," *ICIRA,* p. 267–278, 2019. |
| [5] | N. Komalapati, V. C. Yarra, L. A. Vyas, Kancharla and T. N. Shankar, "Smart Fire Detection and Surveillance System Using IOT," *IEEE,* 2021. |
| [6] | A. E. H. 1. A. S. S. 1. K. 2. B. A. 1. a. I. C. y Kuldoshbay Avazov 1, "Forest Fire Detection and Notification Method Based on AI and IoT Approaches," *Future Internet (ISSN: 1999 - 5903),* vol. 15, no. 2, 2023. |
| [7] | L. WHITE and R. AJAX, "Improved Fire Detection and Alarm Systems.," 2025. |
| [8] | "spntelecom," [Online]. Available: https://spntelecom.vn/cam-bien-nhiet-pisafe-dau-bao-nhiet-khong-day-phong-chay-canh-bao-chay-thong-minh-wifi-dat-tieu-chuan-pccc-phat-hien-chay-no-tu-xa-qua-dien-thoai. |
| [9] | "ManualMachine," Honeywell Home, [Online]. Available: https://digitalassets.resideo.com/damroot/Original/10002/L\_5809SSD\_D.pdf. |
| [10] | Nguyen, N. S., Bui, D. H., & Tran, X. T., "Reducing temporal redundancy in MJPEG using Zipfian estimation techniques," *2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). IEEE,* pp. 65-68, 2014. |
| [11] | A. Manzanera, "σ-δ background subtraction and the Zipf law," *Progress in Pattern Recognition, Image Analysis and Applications: 12th Iberoamericann Congress on Pattern Recognition, CIARP 2007,* pp. 42-51, 2007. |
| [12] | Lacassagne, L., Manzanera, A., & Dupret, A, "Motion detection: Fast and robust algorithms for embedded systems," *IEEE international conference on image processing (ICIP),* pp. 3265-3268, 2009. |
| [13] | Richefeu, A. M. J., "A robust and computationally efficient motion detection algorithm based on σ-δ background estimation," *Indian Conference on Computer Vision, Graphics & amp; Image Processing,* vol. 9, pp. 16-18, 2004. |
| [14] | Dalal, N., & Triggs, B., "Histograms of oriented gradients for human detection," *IEEE computer society conference on computer vision and pattern recognition,* 2005. |
| [15] | Dollár, P., Belongie, S., & Perona, P., "The fastest pedestrian detector in the west," 2010. |
| [16] | Dollar, P., Wojek, C., Schiele, B., & Perona, P., "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence, 34(4),* pp. 743-761, 2011. |
| [17] | Cristianini, N., & Shawe-Taylor, J., An introduction to support vector machines and other kernel-based learning methods, Cambridge university press, 2000. |
| [18] | Ho, H. H., Nguyen, N. S., Bui, D. H., & Tran, X. T, "Accurate and low complex cell histogram generation by bypass the gradient of pixel computation," *4th NAFOSTED Conference on Information and Computer Science,* pp. 201-206, 2017. |
| [19] | Ferryman, J. & Shahrokni, A., "PETS2009: Dataset and challenge.," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance,* 2009. |
| [20] | Andriluka, M., Roth, S. & Schiele, B. , "Monocular 3D Pose Estimation and Tracking by Detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2010. |
| [21] | Andriluka, M., Roth, S. & Schiele, B., "People-Tracking-by-Detection and People-Detection-by-Tracking," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2008. |
| [22] | A. Suleiman, "An energy-efficient hardware implementation of HOG-based object detection at 1080HD 60 fps with multi-scale support.," 2016. |
| [23] | Nguyen, N. D., Bui, D. H., & Tran, X. T., "A novel hardware architecture for human detection using HOG-SVM co-optimization," *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05).,* vol. 1, pp. 886-893, 2019. |
| [24] | Nguyen, T. A., Tran-Thi, T. Q., Bui, D. H., & Tran, X. T, "FPGA-Based Human Detection System using HOG-SVM Algorithm," *International Conference on Advanced Technologies for Communications (ATC),* pp. 72-77, 2023. |
| [25] | [Online]. Available: https://learnopencv.com/non-maximum-suppression-theory-and-implementation-in-pytorch/. |
| [26] | [Online]. Available: https://www.v7labs.com/blog/mean-average-precision. |
| [27] | T. Panda, R. Banerjee, A. Pal, S. K. Bishnu and A. Chakraborty, "IOT-Based Home Automation for LPG Gas and Fire Detection System With Automated Safety Measures," *IEEE,* no. 16 April 2025, 2025. |
| [28] | Shanghai Xian Dai Architecture,Engineering&Consulting Co. ,China , "Information fusion technology based on wireless fire detection and alarm system," *Journal of Physics: Conference Series,* pp. 883-887, 21-11-2013. |
| [29] | M. S. B. Bahrudin, "Development of Fire Alarm System using Raspberry Pi and Arduino Uno," *2013 International Conference on Electrical, Electronics and System Engineering (ICEESE),* 2013. |