JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Home Surveillance and Monitoring System Through ESP32

[1]Dr. K Prasanthi Jasmine, [2]T.Naveen, [3]K.Anand, [4]B.Epaphra

[1]Professor, [2]Student, [3]Student, [4]Student [1]Electronics and Communication, [1]ALIET, Vijayawada, India

Abstract: **This paper presents the design and development of a cost-effective, multi-sensor security system for home environments. The system leverages an ESP32-CAM microcontroller as the central processing unit, integrating various sensors to detect a range of security threats. Magnetic door/window sensors provide detection of unauthorized entry, PIR sensors offer motion detection within designated areas, and an IR flame sensor enables early fire warnings. Additionally, the system can be expanded to include an MQ-6 gas leak sensor for comprehensive environmental monitoring. The ESP32-CAM processes sensor data and triggers pre-programmed responses such as sending smartphone alerts, capturing images/videos with its onboard camera, or activating an audible alarm. The paper details the functionalities of the chosen sensors, their integration with the ESP32-CAM, and the overall system design. This project provides a practical and user-friendly solution for homeowners seeking to enhance their security at an affordable cost.**

Keywords **– Home Security System, ESP32-CAM, Sensor Integration, Internet of Things (IoT), Security Threats.**

I. Introduction

Ensuring the safety and security of our homes is a paramount concern. Traditional security systems can be expensive and complex, often requiring professional installation and monitoring services. This project investigates the development of a cost-effective and user-friendly alternative utilizing readily available components and open-source software. The core of the system lies in the ESP32-CAM microcontroller. This powerful and versatile device incorporates a Tensilica dual-core processor, Wi-Fi and Bluetooth connectivity, and an onboard camera, making it ideal for building internet-connected projects like this security system. By integrating various sensors with the ESP32-CAM, the system can detect a range of security threats, offering a more comprehensive approach to home protection.

This paper will discuss the motivation behind this project, outlining the specific security concerns it addresses and the functionalities it offers. We will then delve into the chosen sensors and their roles within the system, followed by an explanation of the ESP32-CAM's role as the central processing unit. The paper will then detail the hardware and software components used, along with the system's design and implementation. Finally, we will present potential applications, limitations, and considerations for future development.

II. Proposed System

The core of this multi-sensor security system lies in several key components working together seamlessly. The ESP32-CAM microcontroller acts as the brains of the operation. This versatile device boasts a powerful processor, Wi-Fi connectivity, and an onboard camera. These features make it ideal for building internet-connected projects like this security system. The ESP32-CAM receives data from the various sensors, interprets it, and triggers pre-programmed responses based on the detected events.

The effectiveness of the system hinges on the functionalities of the chosen sensors. Magnetic door/window sensors form the first line of defense. They detect the opening or closing of doors and windows, alerting the system to potential unauthorized entry attempts. PIR motion sensors add another layer of security. These sensors can identify movement within designated areas, such as hallways or rooms, triggering an alert if someone is present when they shouldn't be.

For proactive fire protection, the system incorporates an IR flame sensor. This sensor is crucial for early fire warnings. It detects the unique infrared signature of flames, allowing the system to issue an alert before a fire becomes a major threat. This early warning can potentially save lives and minimize property damage. The system can be further enhanced by incorporating an MQ-6 gas leak sensor (or similar). This sensor helps detect the presence of hazardous gases, offering a more comprehensive approach to environmental monitoring within the home.

Finally, the software plays a critical role in bringing the system to life. Custom code, developed using the Arduino IDE or a similar platform, runs on the ESP32-CAM. This code incorporates libraries for various purposes, including communication with the sensors, Wi-Fi connectivity for sending smartphone alerts, and potentially image/video capture using the ESP32-CAM's camera. The code continuously monitors the sensor data and triggers pre-defined actions based on specific events.

III. SYSTEM DESIGN

The system design centres around a ESP32-CAM microcontroller integrated with various sensors to create a security system for a home environment. Here's a breakdown of the key components and their functionalities:

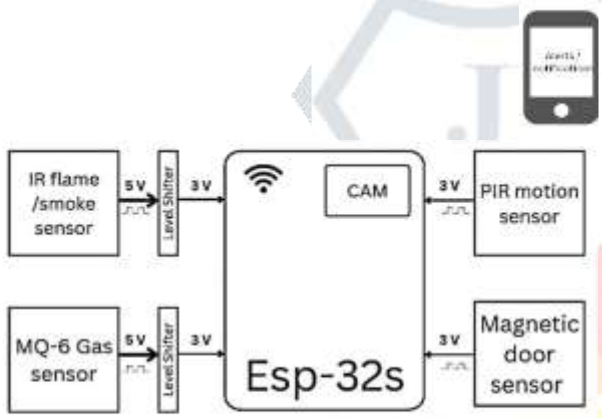

Fig.1 System Architecture Block Diagram

3.1 Hardware:

Central Processing Unit (CPU):

ESP32-CAM Microcontroller: The ESP32-CAM is a tiny powerhouse for wireless camera applications. This development board combines an ESP32 chip with built-in Wi-Fi and Bluetooth with a small OV2640 camera module, all on a single board. With its compact size and low power consumption, it's ideal for creating internet-connected camera projects.

Fig.2 ESP32-CAM

The ESP32-CAM boasts features like video streaming, image capture, and microSD card storage. Popular applications include DIY security cameras, wildlife monitoring systems, and even facial recognition projects. With the help of development environments like Arduino IDE, you can program the ESP32-CAM to suit your specific needs, making it a versatile tool for the creative tinkerer.

This device serves as the brain of the system, receiving signals from the sensors and interpreting the data. The ESP32-CAM can then trigger pre-programmed responses based on the type of event detected by the sensors.

3.2 Sensors:

Magnetic Door/Window Sensor: This sensor detects the opening or closing of doors and windows, potentially indicating unauthorized entry. When a door or window is opened or closed, the sensor sends a signal to the ESP32-CAM.

PIR Motion Sensor: The Passive Infrared (PIR) sensor is the cornerstone of our project's motion detection system. Unlike traditional motion detectors that use radar or microwave technology, PIR sensors are completely passive, meaning they don't emit any radiation themselves. Instead, they continuously measure the infrared radiation levels in their designated field of view. This radiation comes from all objects in the environment, with warmer objects emitting more infrared energy. When a warm object, like a human body, enters the sensor's range, it disrupts the established thermal pattern. This sudden change in infrared radiation triggers a signal from the PIR sensor. The ESP32-CAM microcontroller is programmed to interpret this signal and initiate pre-defined actions, such as capturing an image using the ESP32-CAM's camera and sending an alert notification via Telegram. This functionality allows for remote monitoring of your home's security, keeping you informed of potential security breaches.

IR Flame Sensor: This sensor plays a vital role in early fire detection. It detects the infrared signature of flames, allowing the system to trigger an alarm before a fire becomes a significant threat. The earlier a fire is detected, the more time there is to evacuate and potentially minimize property damage.

MQ-6 Gas Leak Sensor: The MQ-6 gas sensor acts as an invisible guardian within your smart home security system, specifically designed to detect combustible gas leaks. Unlike traditional smoke detectors, the MQ-6 boasts high sensitivity for Liquefied Petroleum Gas (LPG), propane, methane, and butane, commonly found in household appliances and fuels. This targeted detection capability complements the existing fire safety measures in your project. The MQ-6 is strategically connected to the ESP32-CAM's digital output pin (dout). This digital connection allows for clear and immediate readings from the sensor. The code continuously monitors these readings, translating them into gas concentration levels. If a significant rise in gas concentration is detected, exceeding a predefined safety threshold, the system springs into action. An urgent notification is sent via Telegram, alerting you to a potential gas leak within your home. This immediate notification empowers you to take swift action, such as shutting off gas valves and evacuating the premises, potentially preventing a dangerous gas-related incident. By incorporating the MQ-6 sensor, your smart home security system gains a vital layer of defense, safeguarding your home from both fire hazards and combustible gas leaks.

Level Shifter:

Integrating the MQ-6 gas sensor and IR flame sensor into your ESP32-CAM based project unlocks additional safety features like combustible gas leak detection and fire hazard monitoring. However, these sensors introduce a potential hurdle: voltage incompatibility. The MQ-6 and IR flame sensor typically operate at 5V logic, while the ESP32-CAM functions on a 3.3V logic level. This mismatch can lead to serious consequences. Connecting them directly could cause erratic behavior, permanent damage, or even complete failure of these components. To bridge this voltage gap and ensure safe, reliable communication within your system, a level shifter becomes an essential element. Think of it as a translator: it takes the 3.3V signals from the ESP32-CAM and converts them to 5V for the sensors, and vice versa. This level translation safeguards your delicate electronics by ensuring they operate within their intended voltage tolerances. By incorporating a level shifter, you create a harmonious environment where all components can communicate effectively without the risk of voltage-related issues. This not only protects your investment in these components but also guarantees the overall functionality and long-term reliability of your smart home security system.

3.3 Software:

The ESP32-CAM's code acts as the conductor of the security system's symphony. Written in Arduino IDE or a similar platform, the code incorporates libraries for seamless communication with the various sensors. It leverages Wi-Fi connectivity to send smartphone alerts, keeping you informed of any security events. The code can also harness the ESP32-CAM's camera to capture images or videos, providing visual verification of potential threats. For each sensor input, the code is programmed with specific logic to trigger pre-defined responses. This might involve sending alerts, activating alarms, or even controlling connected devices in case of emergencies. By continuously monitoring sensor data and triggering appropriate actions, the code ensures a comprehensive and responsive security system.

The code establishes a secure connection to your WiFi network using the WiFi library. This connection enables the system to transmit alerts and sensor data to your smartphone or designated monitoring center. Next, the code retrieves the Telegram bot token and chat ID, which act as unique identifiers for your Telegram bot. These credentials are essential for sending notification messages and receiving commands directly through the Telegram app on your smartphone. This two-way communication channel between the security system and your smartphone empowers you to stay informed about security events and potentially control functionalities remotely.

3.3.1 Library Selection:

The software development leverages a collection of libraries to interact with various hardware components and functionalities. The WiFi library enables communication with your WiFi network for internet access. The Telegram library allows sending notifications and receiving commands via the Telegram bot you create. To control the ESP32-CAM's camera, including image capture and resolution settings, the camera library is essential. Depending on your chosen sensors, you might need specific libraries for interaction, such as an IR sensor library.

3.4 Overall System Operation:

The system is designed to continuously monitor the home environment through the various sensors. If a sensor detects an event (door opening, motion detected, presence of flame, or gas leak), it sends a signal to the ESP32-CAM. The ESP32-CAM then interprets the signal based on the programmed logic and triggers pre-programmed responses, which might include:

- Sending smartphone alerts to designated recipients.
- Capturing an image or short video using the ESP32-CAM's camera to provide visual verification of the event.
- Activating an audible alarm to deter potential intruders or warn occupants of a fire or gas leak.
- In case of a fire or gas leak, the system might be programmed to trigger connected smart plugs to cut power to appliances as a safety precaution

IV. Output

The system outputs real-time security information through Telegram notifications. These alerts keep you informed about motion detection, door/window breaches, gas leaks (with level shifter), and fire hazards. The system can also optionally capture images for visual verification alongside notifications. Additionally, depending on your implementation, it might offer status updates and remote control via Telegram messages for managing sensor monitoring, notification

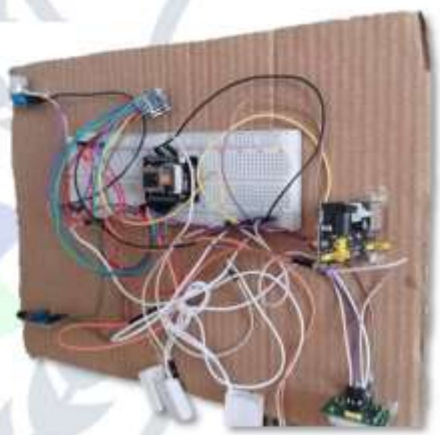

Fig.3 Components on Board

Fig.4 Telegram Bot

V. Conclusion

This paper presented the design and implementation of a cost-effective and user-friendly smart home security system utilizing an ESP32-CAM microcontroller, various sensors, and Telegram communication. The system effectively leverages the ESP32-CAM's processing power, camera capabilities, and WiFi connectivity to provide comprehensive security features for your home environment.

The integration of multiple sensors, including PIR motion detection, magnetic door/window monitoring, MQ-6 gas detection (with level shifter), and IR flame detection, empowers the system to detect a wide range of security threats. Upon detecting suspicious activity, the system promptly transmits alerts and potentially captures images via the Telegram app, keeping you informed and enabling swift action.

Furthermore, the user-friendly interface through Telegram messages allows for remote interaction with the system. Users can enable or disable specific monitoring features, customize notification preferences, and potentially receive real-time video streams (depending on implementation) for enhanced situational awareness.

This project demonstrates the feasibility of developing a robust and feature-rich smart home security system using readily available components and open-source libraries. By incorporating additional functionalities like real-time video monitoring, siren activation upon security breaches, or integration with smart home platforms, the system's capabilities can be further extended, fostering a safer and more secure living environment.

VI. Future Scope

The future holds exciting possibilities for expanding this smart home security system's capabilities. To bolster security, two-factor authentication and data encryption can be implemented, adding extra layers of protection. Integrating real-time video monitoring with cloud storage allows for live viewing of your home environment from anywhere and the ability to review past security events. Expanding the system's reach to platforms like Alexa or Google Home unlocks the potential for voice commands and broader interaction with other smart home devices within your ecosystem. Cloud storage of sensor data and video footage opens doors for advanced cloud analytics. By analyzing this data, the system can identify patterns and trends, potentially improving the accuracy of threat detection. Finally, incorporating AI algorithms to analyze sensor data and video footage can further refine the system's intelligence. AI can learn to distinguish between genuine threats and false alarms, enhancing the system's overall effectiveness.

VII. REFERANCE

[1] M. Sharma and S. C. Gupta, "An Internet of Things Based Smart Surveillance and Monitoring System using Arduino," 2018 Int. Conf. Adv. Comput. Commun. Eng., no. June, pp. 428–433, 2018

[2] A. Jain, S. Basantwani, O. Kazi, and Y. Bang, "Smart surveillance monitoring system," 2017 Int. Conf. Data Manag. Anal. Innov. ICDMAI 2017, pp. 269–273, 2017.

[3] S. Hargude and S. R. Idate, "I-Surveillance : Intelligent Surveillance System Using Background Subtraction Technique," Proc. - 2nd Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2016, vol. 1, 2017.

[4] ArduCam ESP32 UNO User Guide – ArduCam

[5] CMOS OV2640 Camera Module Datasheet – OmniVision

[6] Pertab Rai, Murk Rehman, "ESP32 Based Smart Survellance System," 2019 Int. Conf. Com. Math. and Engineer. Tech. iCoMET 2019.

[7] https://randomnerdtutorials.com/.

[8] https://www.hackster.io/make2explore/