## CHAPTER 1 :- INTRODUCTION

### 1.1 Overview

In the current era of technological advancement, security is not a luxury—it is a necessity. As incidents of theft, unauthorized access, and human intrusion continue to rise across residential, commercial, and public areas, there is a growing demand for effective, affordable, and intelligent surveillance systems. Traditional surveillance systems like CCTV cameras and alarm-based systems have long been the primary methods of ensuring safety and security. However, these systems are not without their shortcomings. They generally require continuous power, constant human monitoring, large storage for 24x7 video recordings, and often fail to differentiate between actual threats and irrelevant movements such as pets, fluttering curtains, or natural elements like wind.

To overcome these challenges, this project introduces a modern, intelligent, and compact system titled “Human Detection System using PIR and ESP32-CAM.” This system is designed to detect human motion using a Passive Infrared (PIR) sensor and respond intelligently by capturing an image via an ESP32-CAM module and sending it directly to the user’s mobile device through the Telegram messaging platform. By incorporating face recognition functionality into the ESP32-CAM, the system can also differentiate between familiar and unfamiliar individuals, thus reducing false alarms and enhancing real-time decision- making.

The project leverages the concept of the Internet of Things (IoT), where physical objects like sensors, cameras, and microcontrollers are connected to the internet to gather, process, and transmit data. This approach allows for real-time monitoring and response without requiring complex infrastructure. The PIR sensor detects motion in its range by sensing changes in infrared radiation emitted by warm-bodied objects. Once motion is detected, the ESP32-CAM captures a high- quality image and processes it using a lightweight face recognition algorithm. If the face is unknown, the system sends a notification to the user via Telegram, including the captured image. If the face is recognized as a family member or trusted individual, the system takes no further action, thereby minimizing unnecessary notifications.

The uniqueness of this system lies in its simplicity and intelligence. Unlike conventional CCTV systems that record continuously and require human operators to manually review footage, this system records only when motion is detected. This not only saves memory and power but also makes the system highly responsive and efficient. The use of Telegram for sending alerts is both practical and scalable. Telegram bots can deliver notifications instantly, support multimedia (like images), and require no special mobile application other than the Telegram app, which is widely available and free.

The hardware components used in this project are low-cost and readily available, making the system accessible to students, hobbyists, and individuals with limited budgets. The PIR sensor is a commonly used motion sensor that works passively— it detects infrared signals without emitting any energy, making it energy-efficient and safe. The ESP32-CAM is a powerful microcontroller that integrates Wi-Fi and a built-in OV2640 camera module, allowing it to perform tasks like image capture, storage, basic processing, and cloud communication within a small form factor.

This project is a response to the limitations of current commercial surveillance systems. High-end systems with artificial intelligence and cloud integration are often too expensive and complex for small-scale or personal users. On the other hand, low-end systems lack the intelligence and responsiveness needed for reliable security. By combining affordability with smart features like selective image capturing and face recognition, this system fills the gap between high-end and basic surveillance technologies.

The broader implications of this project extend beyond individual use. It can be deployed in educational institutions, small offices, warehouses, shops, or rural areas where traditional security systems may not be feasible. Additionally, its modular structure allows for future enhancements, such as integrating cloud storage, extending to multiple cameras, or connecting with a mobile application for more advanced control.

Another major advantage of this system is its energy efficiency. Since the camera and data transmission only activate when motion is detected, the system avoids the constant power draw seen in 24x7 monitoring setups. This is particularly beneficial for battery-powered or solar-powered security installations in remote areas.

In conclusion, the “Human Detection System using PIR and ESP32-CAM” offers a smart, scalable, and cost-effective solution for modern security needs. It blends the core principles of motion sensing, real- time communication, and artificial intelligence in a way that is practical and implementable even at the household level. By prioritizing intelligence, efficiency, and affordability, this system sets a new standard for DIY and educational surveillance projects.

## CHAPTER 2 :- LITERATURE SURVEY

* 1. **Title: ESP32-CAM Based Smart Door Security System with Face Recognition" Authors: S. N. Priya, R. S. Bharathi Year: 2022**

**Publication: International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)**

Presents a comprehensive approach to enhancing home security through the integration of facial recognition technology with the ESP32-CAM module. This study addresses the limitations of traditional security systems, such as keys, RFID cards, and fingerprint sensors, which are susceptible to theft, duplication, or environmental factors affecting their reliability.

The proposed system utilizes the ESP32-CAM, a low-cost microcontroller with an integrated camera and Wi- Fi capabilities, to capture and process facial images for authentication purposes. The system operates by capturing an image of the individual at the door, processing the image to detect and recognize the face, and subsequently granting or denying access based on the recognition results. Authorized users' facial data are stored in a database, and the system compares incoming images against this database to determine access rights.

To enhance user interaction and system responsiveness, the authors have incorporated a Telegram bot that sends real-time notifications to homeowners. This feature allows users to receive alerts and images of visitors, enabling remote monitoring and decision- making regarding access permissions. The integration of the Telegram bot with the ESP32-CAM module exemplifies the effective use of Internet of Things (IoT) technologies in modern security systems.

The system's performance was evaluated based on several parameters, including face recognition accuracy, response time, and reliability under varying lighting conditions. The results demonstrated that the system could accurately recognize authorized individuals and effectively prevent unauthorized access. However, the authors acknowledge that factors such as lighting intensity and facial obstructions

* 1. **Title: Real Time Motion Detection and Surveillance using ESP32 CAM and Telegram Bot" Author: A. B. Balaji, M. Mehta**

**Year:2021**

**Publication: International Journal of Scientific & Engineering Research (IJSER)**

CAM, a low-cost microcontroller with built-in Wi-Fi and camera functionalities, to detect motion in real-time. Upon detecting motion, the system captures an image and sends it directly to the user's Telegram account, providing instant alerts and the ability to monitor premises remotely.

The integration with Telegram is facilitated through the creation of a bot, which interacts with the ESP32- CAM to relay notifications and images. This approach not only ensures timely alerts but also allows users to engage with the surveillance system from anywhere, enhancing the flexibility and responsiveness of home security measures.

The implementation involves programming the ESP32-CAM using the Arduino IDE, setting up the Telegram bot, and configuring the system to detect motion using onboard sensors. The authors provide detailed insights into the hardware setup, software configuration, and the communication protocol between the ESP32-CAM and Telegram.

Overall, the paper demonstrates a practical application of IoT and messaging platforms to create an accessible and real-time surveillance system. By utilizing readily available components and platforms, the proposed solution offers a scalable and user-friendly alternative to conventional security systems.

* 1. **Title: Implementation of Intelligent Surveillance System Using IoT and PIR Sensor" Authors: R. B. Patil, A. B. Gaikwad**

**Year: 2020**

**Publication: International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)**

presents a comprehensive approach to enhancing security through the integration of Internet of Things (IoT) technologies with Passive Infrared (PIR) sensors. The authors aim to develop a cost-effective, efficient, and intelligent surveillance system capable of real-time motion detection and alert generation.

The proposed system utilizes PIR sensors to detect motion by sensing changes in infrared radiation, which typically occur when a human or animal moves within the sensor's range. Upon detecting motion, the system triggers a microcontroller to capture images or videos of the event, which are then transmitted to the user through IoT- enabled communication channels. This real-time alert mechanism ensures prompt notification of potential security breaches, allowing for immediate response.

The authors emphasize the system's scalability and adaptability, highlighting its potential applications in various settings such as homes, offices, and public spaces. By leveraging IoT technologies, the system can be remotely monitored and controlled, providing users with flexibility and convenience. Additionally, the integration of cloud services facilitates data storage and analysis, enabling users to review past events and identify patterns that may inform future security measures.

In their implementation, Patil and Gaikwad focus on optimizing the system's performance by selecting appropriate hardware components and developing efficient software algorithms. They address challenges such as minimizing false positives, ensuring reliable communication between devices, and maintaining system stability under varying environmental conditions. The authors also discuss the importance of user-friendly interfaces, which allow users to easily configure and manage the surveillance system according to their specific needs.

Overall, this paper contributes to the field of smart surveillance by demonstrating

* 1. **FaceDetection and Recognition Using OpenCV Authors: Harmeet Kaur, Arshdeep Singh**

**Year: 2019**

**Publication: International Research Journal of Engineering and Technology (IRJET)**

Presents a comprehensive study on implementing face detection and recognition systems utilizing the OpenCV library. The authors aim to develop a cost-effective and efficient solution for real-time face recognition applications, leveraging the capabilities of OpenCV and associated algorithms.

The study begins by discussing the significance of face detection and recognition in various domains, including security systems, user authentication, and human-computer interaction. The authors emphasize the advantages of using OpenCV, an open-source computer vision library, which offers a wide range of functionalities for image processing and machine learning tasks.

For face detection, the paper explores the use of Haar Cascade Classifiers, which are trained using positive and negative images to detect facial features. The authors detail the process of training these classifiers and implementing them in real-time applications. Additionally, the study examines the use of Local Binary Patterns (LBP) for face detection, highlighting their efficiency and accuracy in varying lighting conditions.

In the realm of face recognition, the paper delves into the implementation of algorithms such as Eigenfaces and Fisherfaces. These techniques involve dimensionality reduction methods like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to extract essential features from facial images. The authors provide insights into the mathematical foundations of these algorithms and their practical applications using OpenCV.

* 1. **Title: Development of Smart Security System using Raspberry Pi and PIR Sensor Authors: Dr. M. L. Valarmathi,**

**G. Akshaya Year: 2018**

**Publication: International Journal of Computer Applications**

Presents a comprehensive approach to enhancing home security through the integration of Internet of Things (IoT) technologies. The authors aim to develop a cost- effective, efficient, and intelligent surveillance system capable of real-time motion detection and alert generation.

The proposed system utilizes the Raspberry Pi, a low-cost, credit-card-sized computer, as the central processing unit. A Passive Infrared (PIR) sensor is employed to detect motion by sensing changes in infrared radiation, which typically occur when a human or animal moves within the sensor's range. Upon detecting

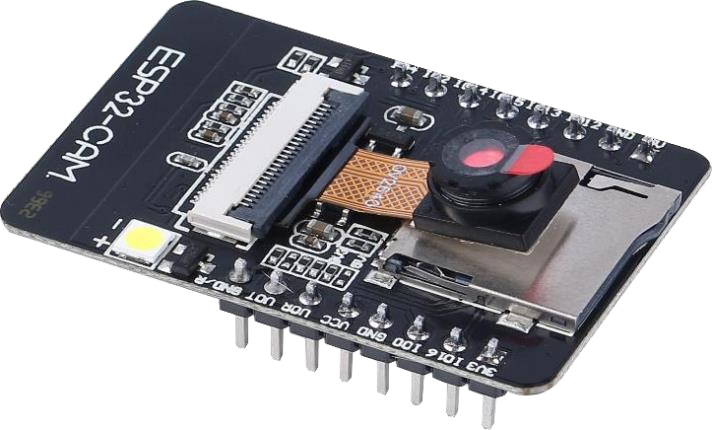
motion, the system triggers the Raspberry Pi to capture images or videos of the event, which are then transmitted to the user through IoT-enabled communication channels. This real-time alert mechanism ensures prompt notification of potential security breaches, allowing for immediate response.

The authors emphasize the system's scalability and adaptability, highlighting its potential applications in various settings such as homes, offices, and public spaces. By leveraging IoT technologies, the system can be remotely monitored and controlled, providing users with flexibility and convenience. Additionally, the integration of cloud services facilitates data storage and analysis, enabling users to review past events and identify patterns that may inform future security measures.

In their implementation, Valarmathi and Akshaya focus on optimizing the system's performance by selecting appropriate hardware components and developing efficient software algorithms. They address challenges such as minimizing false positives, ensuring reliable communication between devices, and maintaining system stability under varying environmental conditions. The authors also discuss the importance of user-friendly interfaces, which allow users to easily configure and manage the surveillance system according to their specific needs.

## CHAPTER 3 :- COMPONENT REQUIRED

### ESP32 CAM Module

****

**Figure 3.1 ESP32 CAM Module**

The ESP32-CAM is a compact and cost-effective development board that integrates the powerful ESP32 microcontroller with a built-in OV2640 camera. This module is widely used in modern IoT, surveillance, and artificial intelligence-based applications. It combines wireless communication capabilities with image capture features, making it highly suitable for real-time security systems and automation projects.

At the heart of the ESP32-CAM is the ESP32-D0WDQ6 chip, which is based on a dual-core Xtensa® 32-bit LX6 processor running at a clock speed of up to 240 MHz. It features 520 KB of internal SRAM and is equipped with an additional 4 MB of PSRAM to handle memory-intensive operations such as image buffering and face recognition. The module also includes 4 MB of flash memory for storing program code and system files.

The board supports Wi-Fi (802.11 b/g/n) and Bluetooth 4.2, including BLE (Bluetooth Low Energy), which enables wireless data transfer and device connectivity without additional hardware. This feature is especially useful in applications where remote monitoring or control is necessary. It allows the ESP32-CAM to act as a wireless surveillance device or to stream data over the network in real time.

Despite its small size, the ESP32-CAM offers several General Purpose Input/Output (GPIO) pins. However, due to the internal use of some pins for the camera interface, only a limited number of GPIOs are available for external use. These pins can be configured for various functionalities, including digital I/O, PWM, I2C, or analog sensing. Additionally, the board provides a microSD card slot that supports cards up to 4 GB, enabling onboard image storage or data logging functionality.

The ESP32-CAM does not come with a built-in USB interface, which means it cannot be directly connected to a computer for programming like some other development boards. Instead, an external USB-to-Serial adapter (FTDI programmer) is required to upload code via UART. The module uses the U0R and U0T pins for serial communication, and GPIO0 must be held LOW during the flashing process to enable programming mode. Once the code is uploaded, the GPIO0 pin can be set HIGH again for normal operation.

The board operates at 3.3V DC, and it is essential to ensure the input voltage is regulated to avoid damaging the device. During active operation, especially when Wi- Fi is enabled and the camera is capturing images, the current consumption can rise to around 180 mA. Therefore, a stable power supply is crucial for reliable performance.

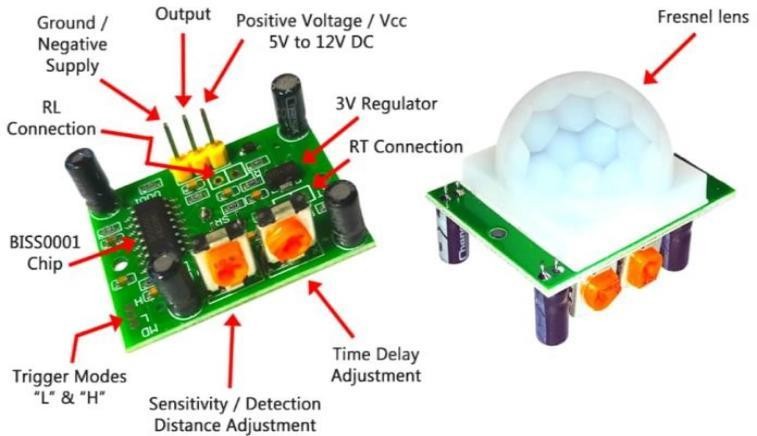
The ESP32-CAM is a versatile and powerful module used in a wide range of applications. In security and surveillance systems, it can be used to monitor motion, recognize faces, or detect intruders. In smart home automation, it can be integrated into doorbell systems or entrance monitoring units. The module is also used in IoT- based projects where visual feedback is required. With support for Over-the-Air (OTA) updates, it allows users to remotely modify and deploy firmware without physically accessing the board.

Overall, the ESP32-CAM is a robust and multifunctional module that balances processing power, camera integration, and wireless communication. It is especially beneficial in student projects and professional prototypes involving vision-based automation or intelligent alert systems. Its affordability and rich feature set make it a preferred choice for embedded developers and IoT enthusiasts alike.

### PIR Sensor

A Passive Infrared (PIR) sensor is a type of electronic sensor that measures infrared (IR) light radiating from objects in its field of view. It is called "passive" because it does not emit any infrared radiation itself; instead, it detects the infrared radiation emitted naturally by surrounding objects. Every object with a temperature above absolute zero emits some level of infrared radiation. The amount of radiation increases with temperature, and warm-blooded animals, including humans, emit infrared radiation in a wavelength range that PIR sensors are specifically designed to detect.

The core component of a PIR sensor is a pyroelectric sensor, typically made from a crystalline material that generates an electrical signal when exposed to infrared radiation. The sensor is sensitive to changes in IR radiation, not the absolute level. Therefore, it detects motion based on the difference in IR radiation levels as



### Figure 3.2 PIR Sensor

a person, animal, or warm object enters or moves across its detection area. To improve sensitivity and focus, a plastic lens known as a Fresnel lens is placed over the sensor. This lens concentrates the infrared radiation onto the pyroelectric element, enhancing its ability to detect motion even at greater distances and wider angles.

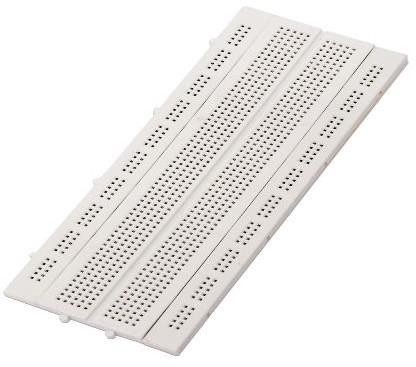
The sensor is idle, both slots detect the same amount of infrared radiation from the surrounding environment. However, when a warm body moves across the sensor’s field of view, it causes a sudden change in the infrared energy levels received by each slot. This differential change is interpreted as motion. The sensor then sends a high signal (usually 3.3V or 5V) to the output pin for a specific time, indicating that motion has been detected. After this period, if no further motion is detected, the output returns to low.

PIR sensors generally operate at voltages between 4.5V and 20V DC and consume very little current, typically less than 60 microamperes when idle, making them ideal for battery-powered and low-power applications. The typical detection range of a standard PIR sensor is around 6 to 12 meters, with a detection angle of about 110 to 120 degrees. These specifications can vary slightly depending on the sensor model and manufacturer.

Most PIR modules used in projects also include additional components like an amplifier, comparator, and sometimes potentiometers that allow users to adjust the sensitivity and the duration of the output signal. Sensitivity determines how far the sensor can detect motion, while the delay time defines how long the output remains high after motion is detected. These adjustments make the PIR sensor versatile and adaptable to a variety of use cases.

Despite its many advantages, the PIR sensor does have limitations. It cannot detect stationary objects because it only senses changes in infrared radiation. Additionally, it may give false positives in environments with rapidly changing temperatures, strong sunlight, or wind blowing warm objects like curtains or pets. To overcome such issues, PIR sensors are often combined with other components or software-based filtering.

* 1. **Bread Board**

****

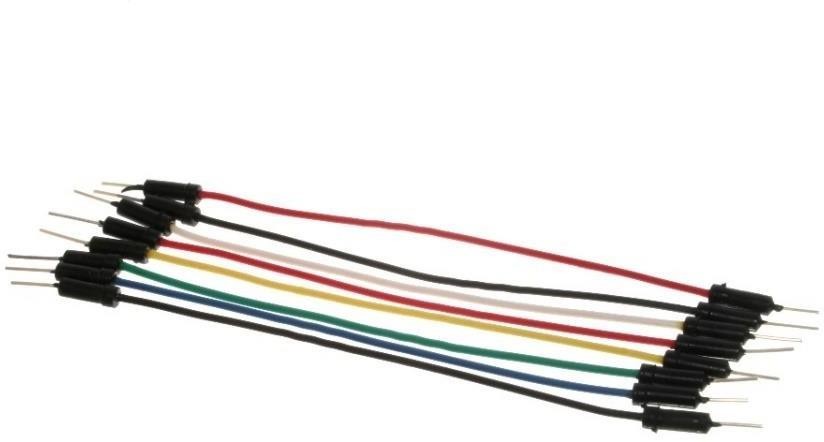
**Figure 3.3 Bread Board**

A breadboard is an essential prototyping tool used in electronics to design and test circuits without the need for soldering. It provides a quick and reusable platform for connecting electronic components such as resistors,

capacitors, LEDs, sensors, and microcontrollers. Named after the literal wooden breadboards that early engineers used to build circuits on, the modern breadboard is made of plastic and contains a grid of holes connected internally by metal strips that allow components to be inserted and interconnected easily. This makes it particularly valuable for beginners, students, and hobbyists who want to experiment and learn circuit design without permanent connections.

The structure of a breadboard typically consists of two main areas: the power rails and the terminal strips. Power rails run horizontally along the top and bottom edges of the board, marked with red and blue or black lines, and are used to distribute power across the board—commonly for +5V and ground. The terminal strip in the center is the main working area where components are placed. It is divided into vertical columns of five interconnected holes, with a central groove that separates the two sides. This groove is often used for integrated circuits (ICs), allowing each pin of the IC to connect to a different column of holes.

* 1. **Connection Wires**

****

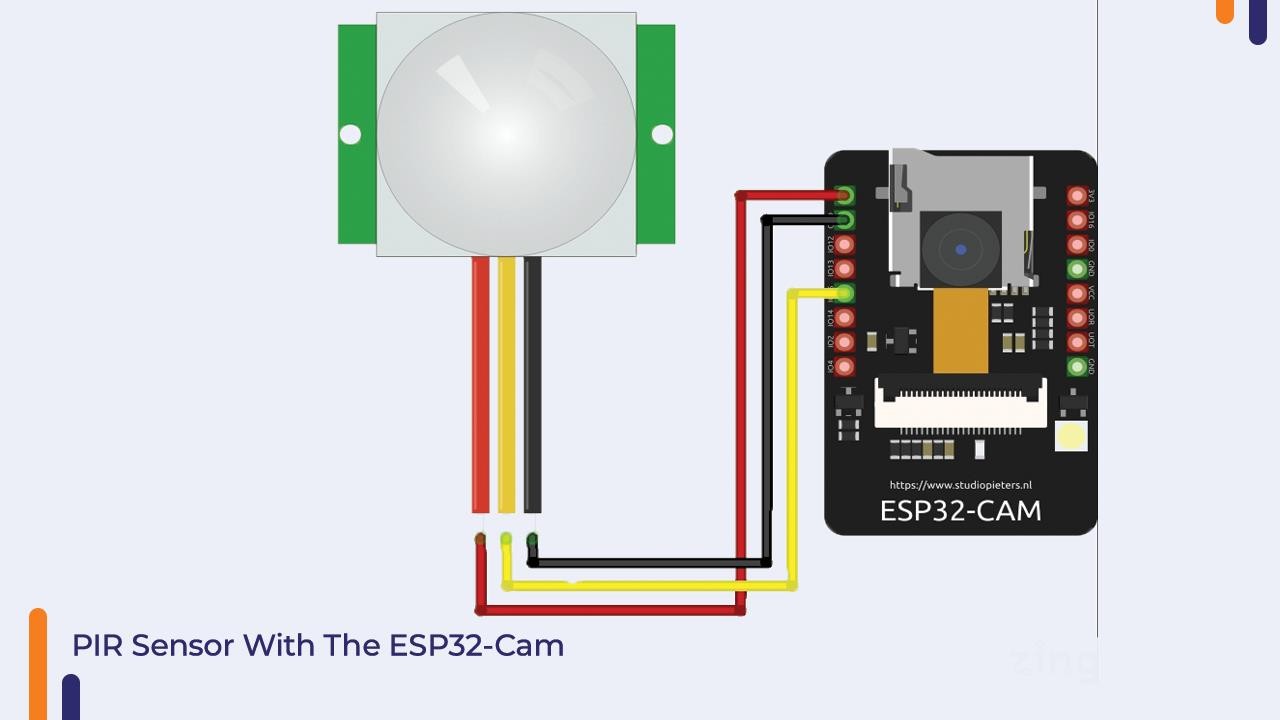
**Figure 3.4 Connecting Wires**

The primary role of connecting wires is to facilitate the transfer of electrical power and signals between different components of an electrical circuit. Without connecting wires, it would not be possible to form complete circuits and allow electrical devices to function. These wires act as conduits, enabling electricity to flow from a power source to a load, or from one component to another, thereby allowing the system to operate efficiently. They are an essential part of any electrical setup, whether in household wiring, electronic devices, or complex industrial systems. Connecting wires are critical in ensuring that power is transmitted reliably and consistently throughout an electrical circuit. These wires allow electricity to flow smoothly from the source to the destination, without significant losses. The quality of the wire material and its insulation ensures minimal power loss, thereby improving the efficiency of the system. If wires were not properly connected, power delivery could be disrupted, causing malfunction or failure of electrical devices. High-quality connecting wires ensure that electricity flows with minimal resistance, enhancing overall system performance.

# 

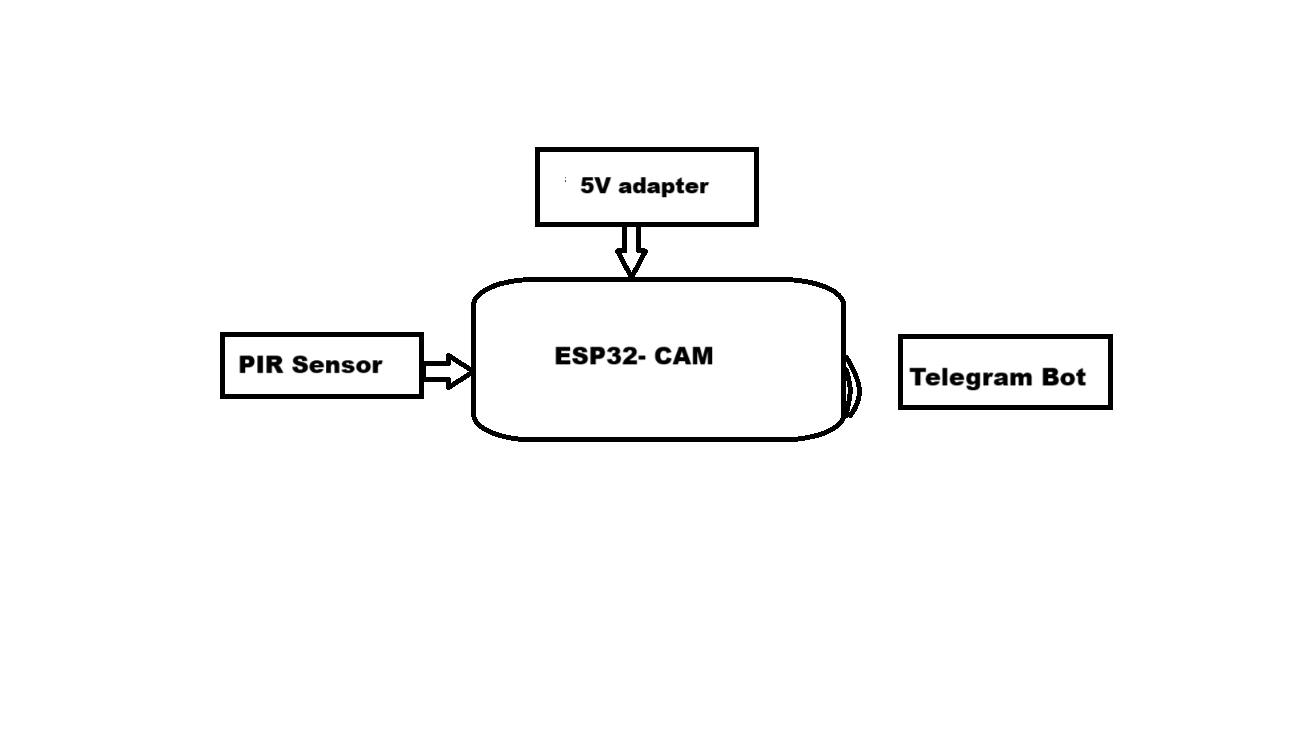
# CHAPTER 4 :- WORKING

### 4.1 Circuit Diagram

****

**Figure 4 .1 Circuit Diagram**

The image illustrates a basic motion detection setup using a PIR (Passive Infrared) sensor and an ESP32 CAM Module development board. This configuration is commonly used in smart home and IoT applications, particularly for detecting human movement. The PIR sensor operates by detecting infrared radiation emitted by warm objects like humans. It has three pins: VCC (power), OUT (signal), and GND (ground). In the diagram, the PIR sensor’s VCC pin is connected to the 3.3V output of the ESP32 CAM Module through a red wire, the GND pin is connected to the ground (GND) of the ESP32 CAM Module via a black wire, and the OUT pin is connected to digital pin D2 (GPIO4) using a yellow wire. This allows the PIR sensor to communicate with the microcontroller, sending a HIGH signal when motion is detected.

****

**Figure 4 .2 Block Diagram**

## CHAPTER 5 :- ADVANTAGE & DISADVANTAGE

### 5.1 Advantages

* Cost**-**effective**:**

Uses affordable ESP32-CAM and PIR sensor; no need for expensive DVRs or cloud subscriptions; free alerting via Telegram.

* Intelligent monitoring**:**

Combines motion detection with face recognition to minimize false alarms by identifying known individuals.

* Energy**-**efficient**:**

Event-driven operation with low-power mode; reduces energy consumption compared to continuous CCTV recording.

* Real-time alerting**:**

Instant image-based notifications via Telegram, supporting secure and encrypted data transmission.

* Flexible and scalable**:**

Modular design allows easy expansion with more sensors or integration with cloud services and mobile apps.

* Easy installation and portability:

Compact hardware with minimal wiring; suitable for temporary setups, rental spaces, and remote locations.

* Enhanced privacy**:**

Local face recognition without sending biometric data to external servers; no continuous video storage reduces misuse risks.

* Educational value**:**

Covers concepts in IoT, embedded systems, computer vision, wireless communication, and cybersecurity, ideal for students and developers.

### Disadvantages

* Limited processing power:

ESP32-CAM can only handle basic face recognition; may cause false positives/negatives, especially in poor lighting.

* Low image quality and poor low-light performance:

Basic OV2640 sensor limits clarity, distance coverage, and effectiveness at night or in dim areas.

* Limited PIR sensor range and detection field:

Detects motion only within 7–10 meters and about 110–120°; slow or direct movements may go undetected.

* Dependence on stable Wi-Fi:

Requires strong internet connectivity for sending alerts; weak networks can delay or disrupt notifications.

* No live video or audio streaming:

Captures only still images, restricting real-time monitoring and interactive response.

* Power supply issues:

High current needed during transmission; unstable or low-capacity power sources may cause device resets or failures.

* Small face recognition database:

Can store only about 7–10 faces; not ideal for environments with many users or multiple devices.

* No tamper detection or physical security:

Vulnerable to physical damage or tampering without alert mechanisms.

## CHAPTER 6 :- APPLICATION

### Application

This project finds relevance in a wide range of domains, thanks to its compact size, real-time functionality, and intelligent face recognition capabilities. The following application areas represent practical scenarios where this technology can be deployed effectively.

* + 1. **Residential Security**

One of the most essential applications of this system is in home and apartment security. It can be placed at entrances, balconies, staircases, and garage areas to detect human movement and identify whether the individual is a recognized household member or an unknown intruder.

Alerts the homeowner in real-time through Telegram, even when they are not at home. Differentiates between family members and strangers using onboard face recognition. Saves storage and power by avoiding continuous video recording

Can be mounted in discreet locations without requiring complex installation.

* + 1. **Retail Shops and Small Businesses**

Shops, showrooms, and small-scale businesses often operate without round-the-clock staff or security guards. This system offers a low-cost surveillance alternative that is easy to set up.

Detects motion after hours and sends intruder alerts instantly.

Helps store owners monitor front counters, storage rooms, and entry points. Prevents unauthorized entry and reduces potential theft.

Useful in establishments where installing expensive CCTV systems is not feasible.

* + 1. **Offices and Workplaces**

Modern offices, especially startups or small enterprises, can benefit from an intelligent monitoring system that respects employee privacy and avoids false alarms can be installed at office entrances, server rooms, or manager cabins. Alerts are sent only when unfamiliar individuals are detected helps in tracking visitors during non-working hours can log image-based attendance when integrated with recognition modules.

* + 1. **Educational Institutions**

Schools, colleges, and coaching centers often face challenges in controlling ccess to buildings and monitoring common areas

Detects unknown individuals near classrooms, exam halls, or staff rooms.

Ensures student safety by monitoring restricted zones like laboratories orserver rooms. Helps in identifying unauthorized movement in hostels and dormitories.

Enhances security without increasing staff workload.

* + 1. **Farmhouses and Remote Monitoring**

Rural homes, farmhouses, and isolated properties often lack surveillance due to absence of infrastructure. System operates even with basic internet and can be powered using batteries or solar panels.

Helps prevent intrusion, theft, or unauthorized access in remote locations.

Sends instant photo alerts, enabling the owner to respond quickly even from far distances. Ideal for seasonal or low-occupancy properties where traditional security is not practical.

* + 1. **Healthcare Facilities and Elderly Monitoring**

Hospitals, clinics, and elder care homes require quiet, non-intrusive surveillance to maintain patient safety. Detects unrecognized visitors in patient wards or critical access zones.

Ensures safety of elderly or vulnerable individuals in assisted living setups. Can be used to monitor entry to medical storage rooms or private care spaces. Notifies staff remotely without disrupting the facility's calm environment.

* + 1. **Event Surveillance and Temporary Installations**

Events such as trade shows, exhibitions, and cultural festivals often need fast- deployable surveillance solutions.

Portable system that can be powered via power banks.

Helps monitor entry points or stall areas without setting up permanent CCTV systems. Offers temporary security for events where installation time and resources are limited. Useful for outdoor gatherings or makeshift setups.

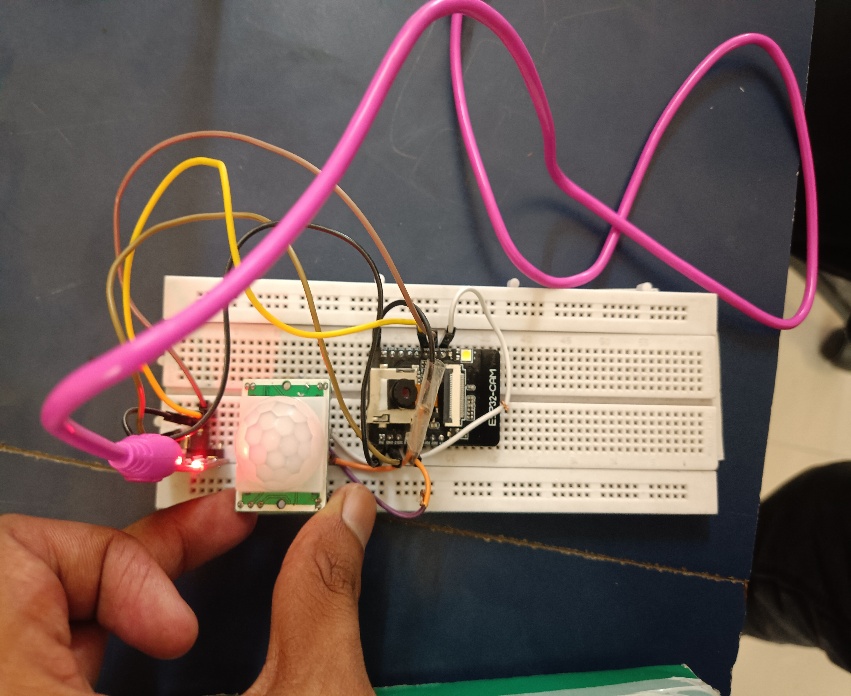
* + 1. **Attendance Logging and Access Control**

With minor modifications, the system can be adapted into an automatic attendance logger. Captures face of every person entering and timestamps the entry.

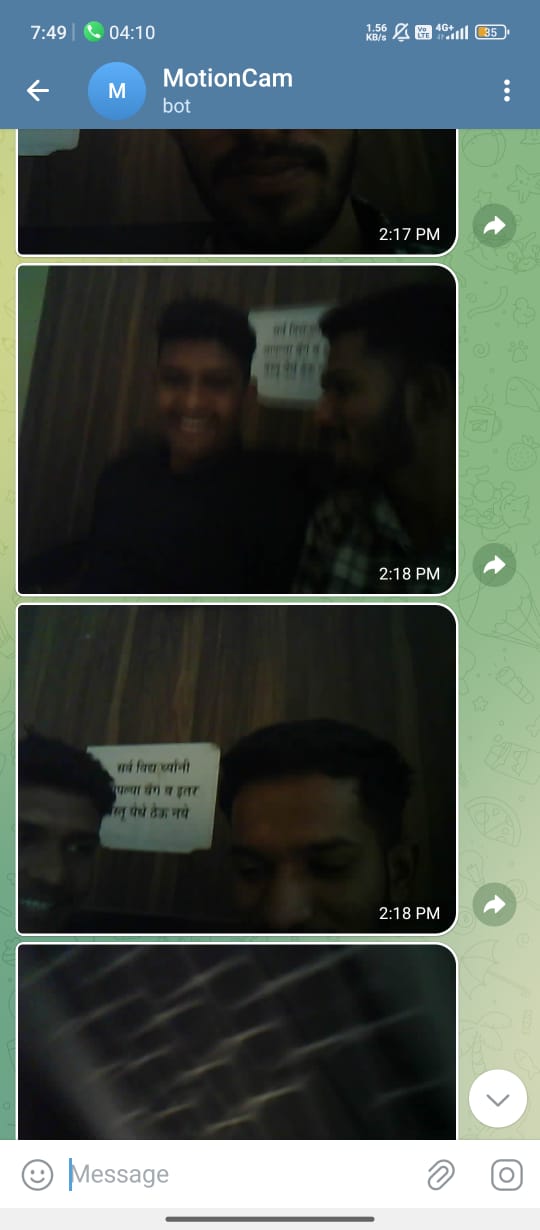
Can be stored locally or sent to cloud-based logs.

Suitable for coaching institutes, gated societies, small companies, or clubs. Enhances accountability without biometric sensors.

# CHAPTER 7 :- RESULT

****

**Figure 7.1 Hardware Output**

****

**Figure 7.2 Software Output**

## CHAPTER 8 :- FUTURE SCOPE & CONCLUSION

**8.1.1 Integration with Cloud Platforms**

Cloud integration can enable real-time uploading and centralized storage of captured images and videos. This would be useful for:

Maintaining a long-term historical log of events.

Providing access to security footage from any device, anytime. Enabling advanced analytics using cloud-based AI tools.

**8.1.2 Real-Time Video Streaming and Playback**

Adding support for real-time video streaming using RTSP or HTTP protocols would allow users to actively monitor their premises via mobile or web applications. Video playback and storage capabilities can further aid post-incident analysis.

* + 1. **Advanced AI and Machine Learning Capabilities**

Currently, the system performs basic on-device face recognition. Future developments may include: Training deep learning models for better accuracy and reduced false positives.

Incorporating object detection (e.g., identifying weapons or intrusions).

**8.1.4Edge Computing for Faster Processing**

Using more powerful edge computing boards like NVIDIA Jetson Nano or Google Coral TPU could allow real-time processing of high-resolution video streams with more complex AI models while maintaining low latency and independence from cloud services.

* + 1. **SMS, Call, and Email Notification Support**

In addition to Telegram alerts, the integration of GSM modules or third-party APIs could enable multi-channel alert systems through:

.

* + 1. **Power Optimization and Outdoor Deployment**

To increase the usability of the system in rural or remote locations, future iterations can include: Solar power support with battery backup.

Weatherproof enclosures for outdoor installation.

Power-efficient sleep modes to conserve energy during inactivity.

* + 1. **Multi-Device Mesh Networking**

Implementing a network of interconnected ESP32-CAM modules can help monitor larger areas, such as campuses, parking lots, or residential colonies. Communication between nodes could be established using protocols like ESP-NOW, MQTT, or Wi-Fi mesh.

* + 1. **Integration with IoT Home Automation Systems**

The system can be extended to trigger other IoT-enabled devices based on events. For example: Automatically turning on lights when motion is detected at night.

Locking smart doors when an unknown individual approaches.

Activating alarms or sending emergency messages to family members or security agencies.

### 8.2 Conclusion

The "Human Detection System using PIR and ESP32-CAM" successfully combines motion sensing, computer vision, and IoT communication technologies to deliver a cost-effective and intelligent home surveillance solution. The system utilizes a PIR sensor to detect motion, and upon detection, activates the ESP32-CAM to capture an image. This image undergoes on-device face recognition to identify whether the person is known or unknown. If an unknown person is detected, the system instantly sends an image-based alert via Telegram, enabling immediate user response.

This project demonstrates the practical application of embedded systems in real-world security use cases. It

effectively reduces false alarms by filtering known individuals and animals, thereby enhancing the reliability of motion-based alerts. The use of the ESP32-CAM, a compact and low-cost camera module, combined with lightweight AI processing, ensures the system remains efficient, portable, and easy to deploy.

Throughout the development process, various aspects of hardware-software integration were explored, including image processing, IoT messaging, and real-time notifications. The system was tested under different environmental conditions to validate its accuracy and performance. Results showed a high level of responsiveness and minimal delay in notification delivery .

The project not only fulfills its objectives but also opens avenues for future enhancements in security and automation. With the increasing demand for intelligent monitoring systems in both urban and rural areas, this solution has strong potential for commercial adaptation, especially in small businesses, homes, and schools. In conclusion, the "Human Detection System using PIR and ESP32-CAM" provides a solid foundation for building smarter, more secure environments and reflects the growing trend of using embedded systems and AI for enhancing everyday safety and convenience.

## CHAPTER 9 :- REFERENCES

**Appendix B: Project Cost and Overall Project Images**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No | Components | Quantity | Cost |
| 1 | **PIR Sensor** | **1** | **250** |
| 2 | **ESP 32 cam** | **1** | **560** |
| 3 | **FDTI** | **1** | **150** |
| 4 | **Breadboard** | **1** | **60** |
| 5 | **Jumper Wire** | **-** | **50** |

**Appendix D: Code**

#include "esp\_camera.h" #include <WiFi.h>

#include <WiFiClientSecure.h> #include "fd\_forward.h" #include "fr\_forward.h" #include "fr\_flash.h"

#include <Base64.h>

const char\* ssid = "VYANKATESH"; const char\* password = "ramramram";

String botToken = "7900583780:AAGAMK-k3Tfh61Z8DOz-Cv78KLG-jBx1wfY"; String chatId = "1186014886";

#define PIR\_SENSOR 13

#define FACE\_ID\_SAVE\_NUMBER 7 // Number of faces to recognize #define ENROLL\_CONFIRM\_TIMES 5

// ESP32-CAM settings (AI Thinker) #define PWDN\_GPIO\_NUM -1

#define RESET\_GPIO\_NUM -1

#define XCLK\_GPIO\_NUM 0

#define SIOD\_GPIO\_NUM 26

#define SIOC\_GPIO\_NUM 27

|  |  |
| --- | --- |
| #define Y9\_GPIO\_NUM | 35 |
| #define Y8\_GPIO\_NUM | 34 |
| #define Y7\_GPIO\_NUM | 39 |
| #define Y6\_GPIO\_NUM | 36 |
| #define Y5\_GPIO\_NUM | 21 |
| #define Y4\_GPIO\_NUM | 19 |
| #define Y3\_GPIO\_NUM | 18 |
| #define Y2\_GPIO\_NUM | 5 |

#define VSYNC\_GPIO\_NUM 25

#define HREF\_GPIO\_NUM 23

#define PCLK\_GPIO\_NUM 22

// Face recognition data

static face\_id\_list id\_list = {0};

static mtmn\_config\_t mtmn\_config = {0}; static int8\_t detection\_enabled = 1;

static face\_recognition\_state\_t recognition\_state = FACE\_REC\_IDLE; static int8\_t is\_enrolling = 0;

static int8\_t do\_recognition = 1;

static face\_id\_node \*enrolled\_faces[FACE\_ID\_SAVE\_NUMBER] = {0}; void startCameraServer();

void setup() { Serial.begin(115200); pinMode(PIR\_SENSOR, INPUT);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) { delay(500); Serial.print(".");

}

Serial.println("WiFi connected");

camera\_config\_t config;

config.ledc\_channel = LEDC\_CHANNEL\_0; config.ledc\_timer = LEDC\_TIMER\_0; config.pin\_d0 = Y2\_GPIO\_NUM; config.pin\_d1 = Y3\_GPIO\_NUM; config.pin\_d2 = Y4\_GPIO\_NUM; config.pin\_d3 = Y5\_GPIO\_NUM; config.pin\_d4 = Y6\_GPIO\_NUM; config.pin\_d5 = Y7\_GPIO\_NUM; config.pin\_d6 = Y8\_GPIO\_NUM; config.pin\_d7 = Y9\_GPIO\_NUM; config.pin\_xclk = XCLK\_GPIO\_NUM; config.pin\_pclk = PCLK\_GPIO\_NUM; config.pin\_vsync = VSYNC\_GPIO\_NUM; config.pin\_href = HREF\_GPIO\_NUM; config.pin\_sscb\_sda = SIOD\_GPIO\_NUM;

config.pin\_sscb\_scl = SIOC\_GPIO\_NUM; config.pin\_pwdn = PWDN\_GPIO\_NUM; config.pin\_reset = RESET\_GPIO\_NUM; config.xclk\_freq\_hz = 20000000; config.pixel\_format = PIXFORMAT\_JPEG;

if(psramFound()) {

config.frame\_size = FRAMESIZE\_VGA; config.jpeg\_quality = 10;

config.fb\_count = 2;

} else {

config.frame\_size = FRAMESIZE\_CIF; config.jpeg\_quality = 12;

config.fb\_count = 1;

}

esp\_err\_t err = esp\_camera\_init(&config); if (err != ESP\_OK) {

Serial.printf("Camera init failed with error 0x%x", err); return;

}

// Initialize face detection mtmn\_config = mtmn\_init\_config();

face\_id\_init(&id\_list, FACE\_ID\_SAVE\_NUMBER, ENROLL\_CONFIRM\_TIMES);

// Load enrolled family faces from flash memory loadEnrolledFaces();

Serial.println("System ready. Motion detection activated.");

}

void loop() {

if (digitalRead(PIR\_SENSOR) == HIGH) { Serial.println("Motion Detected!");

// Check if the person is a family member

bool isFamilyMember = detectAndRecognizeFace();

if (!isFamilyMember) {

Serial.println("Unknown person detected. Sending photo alert!"); sendPhoto();

} else {

Serial.println("Family member recognized. No alert needed.");

}

delay(10000); // wait before checking again

}

}

bool detectAndRecognizeFace() { camera\_fb\_t \*fb = esp\_camera\_fb\_get(); if (!fb) {

Serial.println("Camera capture failed"); return false;

}

dl\_matrix3du\_t \*image\_matrix = dl\_matrix3du\_alloc(1, fb->width, fb->height, 3); if (!image\_matrix) {

Serial.println("dl\_matrix3du\_alloc failed"); esp\_camera\_fb\_return(fb);

return false;

}

bool converted = fmt2rgb888(fb->buf, fb->len, fb->format, image\_matrix->item); esp\_camera\_fb\_return(fb);

if (!converted) { Serial.println("fmt2rgb888 failed"); dl\_matrix3du\_free(image\_matrix); return false;

}

box\_array\_t \*net\_boxes = face\_detect(image\_matrix, &mtmn\_config); if (!net\_boxes) {

Serial.println("No face detected");

dl\_matrix3du\_free(image\_matrix); return false;

}

// If there's at least one face in the image if (net\_boxes->len > 0) {

dl\_matrix3du\_t \*aligned\_face = dl\_matrix3du\_alloc(1, FACE\_WIDTH, FACE\_HEIGHT, 3); if (!aligned\_face) {

Serial.println("Could not allocate aligned face matrix"); free(net\_boxes->box);

free(net\_boxes); dl\_matrix3du\_free(image\_matrix); return false;

}

if (align\_face(net\_boxes, image\_matrix, aligned\_face) == ESP\_OK) {

// Recognize the face

int matched\_id = recognize\_face(&id\_list, aligned\_face); if (matched\_id >= 0) {

Serial.printf("Face recognized as ID %d (Family member)\n", matched\_id); dl\_matrix3du\_free(aligned\_face);

free(net\_boxes->box); free(net\_boxes); dl\_matrix3du\_free(image\_matrix);

return true; // Family member recognized

} else {

Serial.println("Unknown face");

}

}

dl\_matrix3du\_free(aligned\_face);

}

free(net\_boxes->box); free(net\_boxes); dl\_matrix3du\_free(image\_matrix);

return false; // Not a family member or no face detected

}

void sendPhoto() {

camera\_fb\_t \* fb = esp\_camera\_fb\_get(); if (!fb) {

Serial.println("Camera capture failed"); return;

}

WiFiClientSecure client; client.setInsecure();

String message = "https://api.telegram.org/bot" + botToken + "/sendPhoto";

String head = "--123456\r\nContent-Disposition: form-data; name=\"chat\_id\"\r\n\r\n" + chatId

+ "\r\n--123456\r\nContent-Disposition: form-data; name=\"photo\"; filename=\"image.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";

String tail = "\r\n--123456--\r\n";

int contentLength = head.length() + fb->len + tail.length();

client.connect("api.telegram.org", 443);

client.println("POST /bot" + botToken + "/sendPhoto HTTP/1.1"); client.println("Host: api.telegram.org");

client.println("Content-Length: " + String(contentLength)); client.println("Content-Type: multipart/form-data; boundary=123456"); client.println();

client.print(head); client.write(fb->buf, fb->len); client.print(tail);

while (client.connected()) {

String line = client.readStringUntil('\n'); if (line == "\r") break;

}

esp\_camera\_fb\_return(fb);

}

// Function to save an enrolled face to flash bool saveEnrolledFace(int face\_id) {

if (id\_list.count == 0 || face\_id >= id\_list.count) return false;

face\_id\_node \*face = id\_list.head; for (int i = 0; i < face\_id; i++) { face = face->next;

}

if (face == NULL) return false;

Serial.printf("Saving face ID %d to flash\n", face\_id);

return face\_id\_save(&id\_list, face\_id, FACE\_ID\_SAVE\_NUMBER);

}

// Function to load enrolled faces from flash void loadEnrolledFaces() {

Serial.println("Loading enrolled faces from flash memory"); face\_id\_read\_from\_flash(&id\_list);

// Count loaded faces int count = 0;

face\_id\_node \*face = id\_list.head; while (face != NULL) {

count++;

face = face->next;

}

Serial.printf("Loaded %d enrolled family member faces\n", count);

}

// Enroll new face bool enrollFace() {

camera\_fb\_t \*fb = esp\_camera\_fb\_get(); if (!fb) {

Serial.println("Camera capture failed"); return false;

}

dl\_matrix3du\_t \*image\_matrix = dl\_matrix3du\_alloc(1, fb->width, fb->height, 3);

if (!image\_matrix) { Serial.println("dl\_matrix3du\_alloc failed"); esp\_camera\_fb\_return(fb);

return false;

}

bool converted = fmt2rgb888(fb->buf, fb->len, fb->format, image\_matrix->item); esp\_camera\_fb\_return(fb);

if (!converted) { Serial.println("fmt2rgb888 failed"); dl\_matrix3du\_free(image\_matrix); return false;

}

box\_array\_t \*net\_boxes = face\_detect(image\_matrix, &mtmn\_config); if (!net\_boxes) {

Serial.println("No face detected for enrollment"); dl\_matrix3du\_free(image\_matrix);

return false;

}

// If there's at least one face in the image if (net\_boxes->len > 0) {

dl\_matrix3du\_t \*aligned\_face = dl\_matrix3du\_alloc(1, FACE\_WIDTH, FACE\_HEIGHT, 3); if (!aligned\_face) {

Serial.println("Could not allocate aligned face matrix"); free(net\_boxes->box);

free(net\_boxes); dl\_matrix3du\_free(image\_matrix); return false;

}

if (align\_face(net\_boxes, image\_matrix, aligned\_face) == ESP\_OK) { int enrolled\_id = enroll\_face(&id\_list, aligned\_face);

if (enrolled\_id >= 0) {

Serial.printf("Face enrolled with ID %d\n", enrolled\_id);

// Save to flash

if (saveEnrolledFace(enrolled\_id)) { Serial.println("Face saved to flash memory");

} else {

Serial.println("Failed to save face to flash");

}

dl\_matrix3du\_free(aligned\_face); free(net\_boxes->box); free(net\_boxes); dl\_matrix3du\_free(image\_matrix); return true;

} else {

Serial.println("Failed to enroll face");

}

}

dl\_matrix3du\_free(aligned\_face);

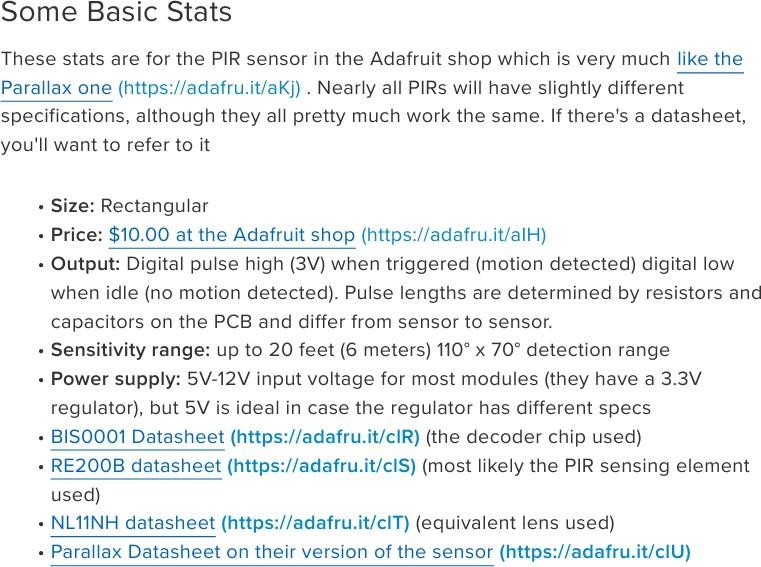
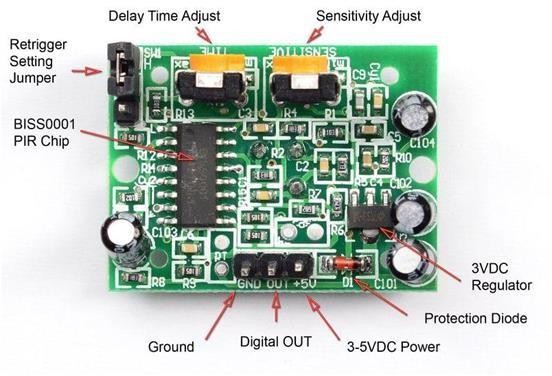
}

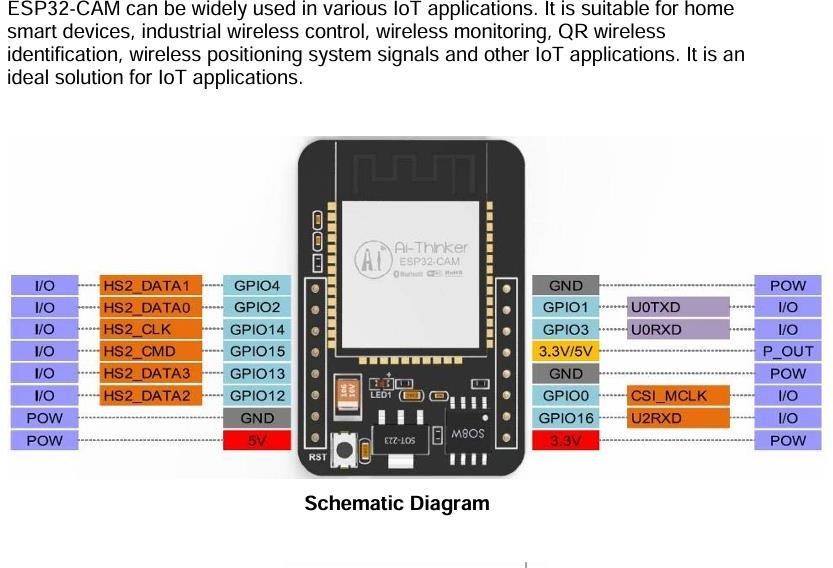
free(net\_boxes->box); free(net\_boxes); dl\_matrix3du\_free(image\_matrix); return false;

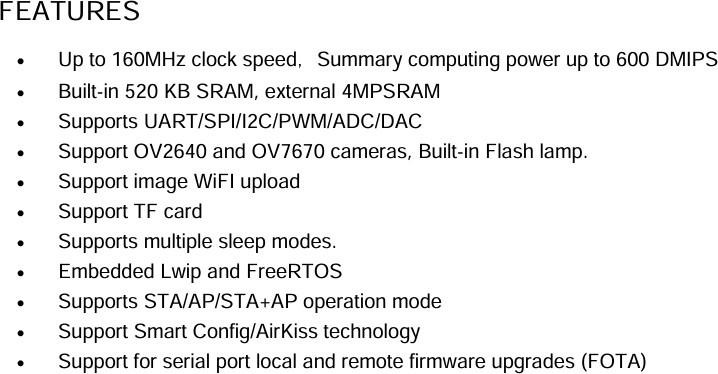
}

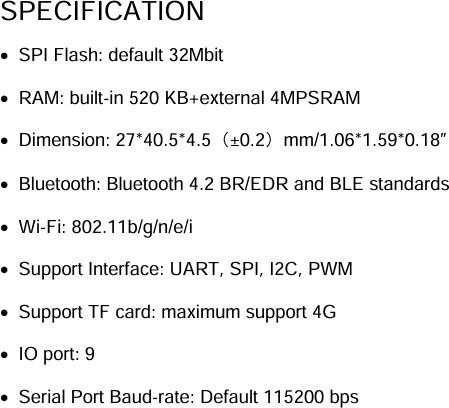
# Appendix E: Data sheet

**PIR Sensor**

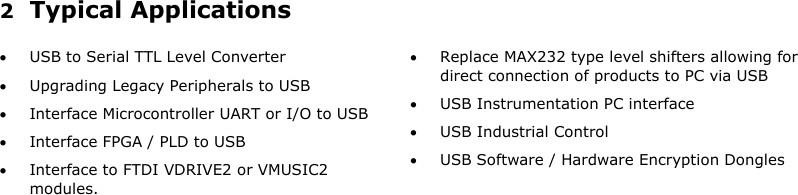
****

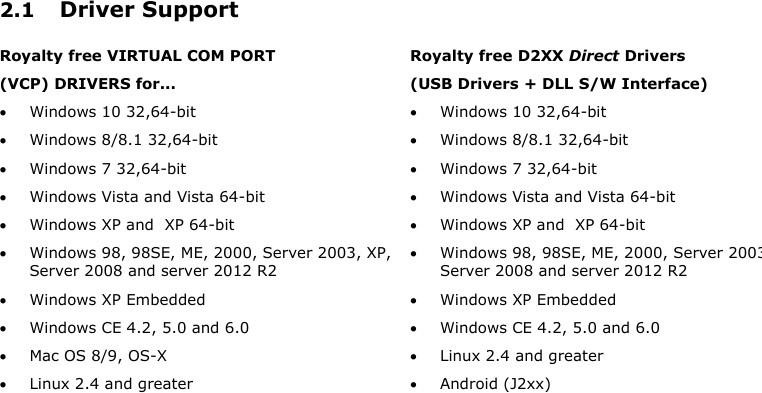


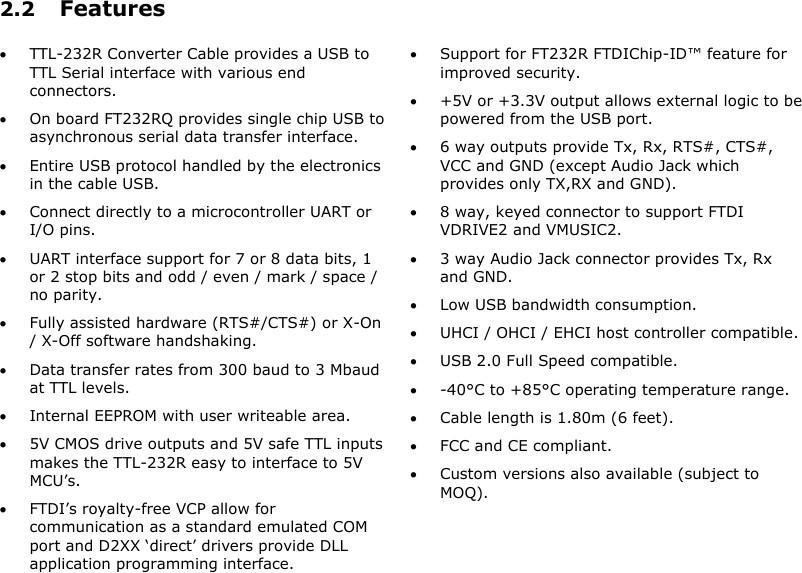
****



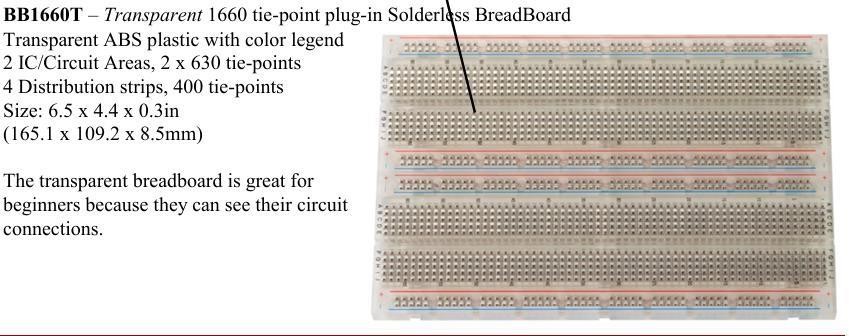
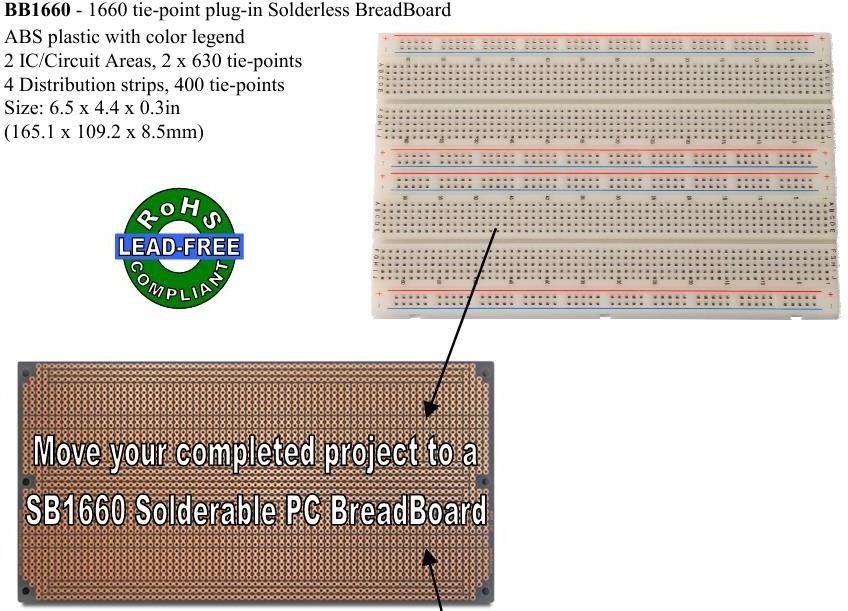
## FDTI

****

****



**Breadboard**

****