TABLE OF CONTENTS

CHAPTERS	TITLE	PAGE NO
1	**ABSTRACT**	04
2	**INTRODUCTION** 2.1 OVERVIEW 2.2 PROBLEM DEFINITION 2.3 HARDWARE AND SOFTWARE USED	06
3	**COMPONENTS** 3.1 ESP32 CAM 3.2 PIR SENSOR 3.3 FIDI 232 USB TO SERIAL INTERFACE BOARD	09
4	**SYSTEM ANALYSIS** 4.1 EXISTING SYSTEM 4.2 PROPOSED SYSTEM	13
5	**SYSTEM SPECIFICATION** 5.1 HARDWARE SPECIFICATION 5.2 SOFTWARE SPECIFICATION	15
6	**SYSTEM WORKING / DESIGN** 6.1 BLOCK DIAGRAM 6.2 CIRCUIT DIAGRAM 6.3 ESP32 CAM WITH PIR SENSOR ON BREADBOARD	17
7	**SOFTWARE CODING**	20
8	**ADVANTAGE**	28
9	**APPLICATION**	30
10	**FUTURE ENHANSMENT / SCOPE**	32
11	**CONCLUSION & REFERENCES**	34

LIST OF FIGURES

FIGURE NO	NAME OF FIGURES	PAGE NO
1	ESP32 CAM	09
2	PIR SENSORS	10
3	FTDI 232 USB to Serial interface board	11
4	Architecture of Motion Detector	17
5	Circuit for Programming ESP32 CAM	18
6	ESP32 CAM with PIR Sensor on Breadboard	18
7	Project Setup	27
8	Project Output	27

CHAPTER 1

ABSTRACT

Motion detection is a crucial technology used in various fields, including security, automation, and computer vision. It involves the identification of movement within a given space, typically using sensors or cameras.

This project aims to design and implement a motion detector system using state-of-the-art technology. The core objective is to detect motion in real-time and trigger immediate responses.

It presents a motion detection system built around the ESP32-CAM development board. The system employs a Passive Infrared (PIR) sensor to detect motion within its proximity. When motion is detected, an LED indicator is activated, and further actions, such as capturing images using the ESP32-CAM's camera module, can be triggered. The system's core functionality is achieved by interfacing the PIR sensor with the ESP32-CAM board. When motion is sensed, the ESP32-CAM responds by illuminating an indicator LED and executing custom-defined actions, making it suitable for various applications, including home security, surveillance, or automation. This abstract provides a high-level overview of the motion detection system's key components and its capability to detect and respond to motion, making it a versatile tool for IOT and DIY project.

Key features of this project include adjustable sensitivity settings to minimize false alarms, integration with IOT platforms for remote monitoring and control, and a response mechanism that ranges from sending alerts to activating smart devices. The system's versatility allows it to be applied to a variety of scenarios, including home security, surveillance, and energy-efficient automation.

The ESP32-CAM is in deep sleep mode with external wake up enabled. When motion is detected, the PIR motion sensor sends a signal to wake up the ESP32.The ESP32- CAM takes a photo and saves it on the micro-SD card. Insert a formatted micro SD card and power your circuit – you can use a portable charger, after uploading code and building the circuit. Then click the reset (RST) button to get it to operate again. It activates the flash, snaps a photo, and saves it to the micro SD card when it senses motion. It goes back to deep sleep mo de until a new signal from the PIR motion sensor is received. After that, connect the micro SD card to your computer to view the photographs. We can easily design this motion sensor with a photo capture circuit using ESP 32 - CAM, PIR sensor, and some basic electronics components. And we can easily supply this circuit with a 5V DC charger.

CHAPTER 2

INTRODUCTION

2.1 OVERVIEW

A security framework that can have the option to recognize and screen the zone and respond viably to security danger is in incredible need. Because of the expanding number in wrongdoing and theft, the requirement for a proficient security framework is fundamental. There are now loads of security frameworks in the realm of innovation as of now, in the market for both indoor and outdoor applications, for example, Ultrasonic identifiers, CCTV, microwave indicators, photoelectric finders, infrared locators and so forth. Anyway, the greater part of these frameworks of being costly in the market, or they require increasingly electrical force development, more memory space of usage of the account framework and complex circuitry, and so on. Accordingly, an answer for conquer these issues could be by utilizing a sensor of minimal effort which can identify the interlopers, and other security astonishingly inside the sensor's discovery run and creates and yield. This yield is can likewise be utilized to additionally flagging and actuating other security gadgets like caution framework, helping framework and other comparable security danger gadgets. Which means this framework can spare force utilizations on the grounds that these segments get activated just when there are gate crashers and security dangers in the sensor's discovery run. A Passive Infrared radiation motion sensor is security-based system that saves the power consumption and the memory space of the recording system. Passive Infrared Radiation (PIR) sensor detects the change in infrared radiation of warm-blooded moving object in its detection range. The use of motion detectors goes back to ancient societies that developed agriculture and motion detection of people and things can be traced back to the early decades of the 20th century, with many of the same principles still in use today. The objective of this work is to develop a simplified version of a PIR sensor which can be installed on campus and houses for lightening systems, shops and malls for security systems and other major applications and places all over the globe.

2.1. PROBLEM DEFINITION

There has been drastic increase in theft and burglary, robbing of houses, stores, shops and banks etc. This research focuses on how to configure a simple home security framework using a PIR sensor (Passive Infra -Red). PIR motion sensor used detects the motion of the person entering the house. There are now loads of security frameworks in the realm of innovation as of now, in the market for both indoor and outdoor applications, for example, Ultrasonic identifiers, CCTV, microwave indicators, photoelectric finders, infrared locators and so forth. But all of these being costly in the market and they require increasingly electrical force development, more memory space and complex circuitry. Accordingly, an answer for conquer these issues could be by utilizing a sensor of minimal effort which can identify the interlopers.

2.2. HARDWARE AND SOFTWARE USED

Hardware used:

1) ESP32-CAM (AI Thinker),
2) PIR Motion Sensor,
3) FTDI 232 USB to Serial Interface board,
4) 5-volt DC supply (Battery),
5) BC547 NPN Transistor,
6) RESISTORS (220ohm, 1k, 10k),
7) LED 5-mm,
8) Bread Board,
9) Jumper Wires

Software used:

1) Arduino IDE.

CHAPTER 3

COMPONENTS

3.1. ESP32 CAM

ESP32-CAM is a low-cost ESP32-based development board with onboard camera, small in size. It is an ideal solution for IOT application, prototypes constructions and DIY projects. The board integrates Wi-Fi, traditional Bluetooth and low power BLE, with 2 high performance 32-bit LX6 CPUs. It adopts 7-stage pipeline architecture, on-chip sensor, Hall sensor, temperature sensor and so on, and its main frequency adjustment ranges from 80MHz to 240MHz. Fully compliant with Wi-Fi 802.11 and Bluetooth 4.2 standards, it can be used as a master mode to build an independent network controller, or as a slave to other host MCUs to add networking capabilities to existing devices. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IOT applications.

Fig-1

3.2. PIR SENSOR

All living objects, whose body temperature is more than 0 degree C, emit the heat in form of infrared radiation through their body, also called as thermal radiations. This Radiated energy is invisible to human eye. These Signals can be detected by using PIR sensor which is specially designed for such purpose. In Passive Infrared (PIR) Sensor, passive word indicates PIR Sensor does not generate or radiate any energy for detection purposes. PIR Sensors don't detect or measure "HEAT"; they detect the infrared radiation emitted or reflected from objects. They are small, inexpensive, low power and easy to use. They are commonly found at home, medical, factories etc. areas.

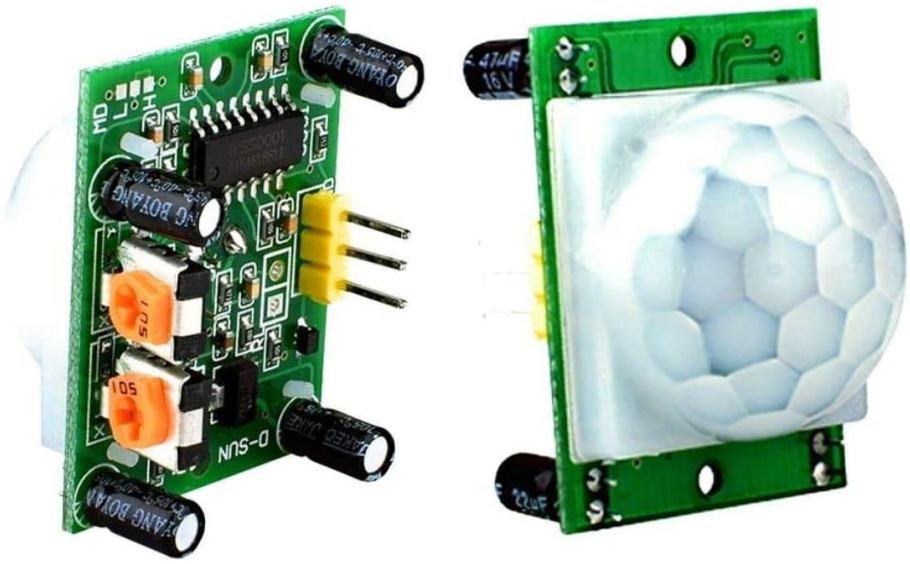

Fig-2

3.3. FTDI 232 USB to Serial interface board

This cable is used to transmit and receive data between computer and external devices such as micro controllers, Arduino, development modules (Bluetooth, GPS, GSM, etc.). Most importantly, FTDI cable is used to connect RS232 standard based devices to Personal computers and laptops.

GROUND: Connect to the ground pin of the device to which you want to connect with the computer.

CTS: Clear to Send = It control input and is used to clear the send request of Data.

VCC: Connect it with V cc.

TxD: Transmit Asynchronous Data = It is an output pin and used to transmit output data asynchronously.

RxD: Receive Asynchronous Data = It is an input pin and used to receive input data asynchronously.

RTS: This is a control output pin and is used to make a request for sending the data.

Standard interface layout, compatible with a variety of Arduinos such as the Pro Mini Original FTDI FT232 chip, stable performance USB power has current protection, using 500MA self-restore fuse
RXD/TXD transceiver communication indicator.
With power, sending, receiving indicator, working status LED indicators
Mini USB Port Connection, Support 3.3V, 5V.

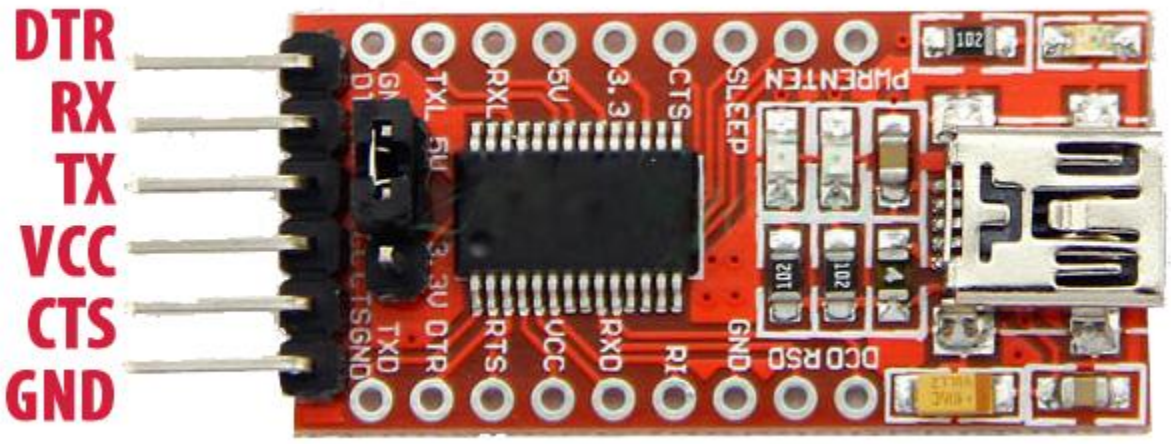

Fig-3

CHAPTER 4

SYSTEM ANALYSIS

4.1. EXECUTED SYSTEM

The executed system for a motion detector is designed to detect movement and trigger responses in real-time. Here's a general overview of such a system.

a. **Hardware Components:**
   - PIR Sensor: A Passive Infrared (PIR) sensor is used to detect changes in heat or infrared radiation caused by motion. It's a key component for motion detection.
   - Microcontroller: A microcontroller like an Arduino, Raspberry Pi, or ESP32 processes the sensor data and controls the system's response.
   - Output Device: These may include LEDs, alarms, or actuators that respond to detected motion.

b. **Software:**
   - Motion Detector Algorithm: The microcontroller uses a motion detection algorithm to process data from the PIR sensor and determine if motion is present.
   - Response Logic: When motion is detected, the system activates output devices, such as turning on lights, sounding an alarm, or triggering a camera to capture images.
   - Data Logging and Alerting: Some systems log motion events and send alerts to users via email, SMS, or mobile apps.

c. **Applications:**
   - Executed systems are often used in home security, lighting automation, and basic surveillance systems.

4.2. PROPOSED SYSTEM

A proposed system for a motion detector can be designed to enhance existing capabilities or address specific needs. Here's an outline of a proposed system.

1. **Enhanced Detection:**
   - Consider using advanced sensors or multiple sensor types to improve motion detection accuracy, especially in challenging environments.
2. **Integration With IOT:**
   - Integrate the system with the Internet of Things (IoT) to enable remote monitoring and control through smartphone apps or web interfaces.
3. **Machine Learning:**
   - Incorporate machine learning algorithms for more intelligent motion analysis. This can help distinguish between different types of motion and reduce false alarms.
4. **Cloud Connectivity:**
   - Use cloud services to store and analyze motion data, allowing for historical tracking and advanced analytics.

CHAPTER 5

SYSTEM SPECIFICATION

A motion detector typically combines both Hardware and Software components. Here an overview of the Hardware and Software Specifications for the motion detector:

5.1. HARDWARE SPECIFICATIONS

- **PIR Sensors (Passive Infrared Sensors):** PIR sensors are a common hardware component in motion detectors. They detect changes in infrared radiation, typically emitted by warm objects, including humans and animals.

- **Camera:** Advanced motion detectors may include a camera for visual verification of detected motion.

- **Control Unit:** This can be a microcontroller or a dedicated motion detector chip. It processes the sensor data and triggers an alarm or notification when motion is detected.

- **Power Supply:** Typically, motion detectors are powered by batteries or through a wired connection to the electrical grid.

- **Housing:** The sensor components are enclosed in a housing that protects them from environmental factors and tampering.

5.2. SOFTWARE SPECIFICATION

- **Signal Processing Algorithm:** The software component in a motion detector includes an algorithm that processes the sensor data. For PIR sensors, this might involve analyzing changes in the infrared signal over time.

- **Threshold Setting:** Motion detectors allow you to set sensitivity levels. The software includes parameters for adjusting the threshold at which motion is detected.

- **Timing and Delay:** You can configure how long the motion detector remains active after detecting motion (e.g., a few seconds or minutes). This is managed by software.

- **Alert System:** The software includes an alert system that triggers notifications or alarms when motion is detected. This could be an audible alarm, a notification to a connected device, or both.

- **Communication Protocol:** Advanced motion detectors may include communication protocols like Wi-Fi or Zig bee to connect to smart home systems or security networks.

- **Firmware Update:** Some motion detectors support firmware updates, allowing for improvements and bug fixes over time.

CHAPTER 6

SYSTEM WORKING / DESIGN

The sensor can detect the presence of intruders. Upon detection of IR, PIR sensor generates the output in the form of electrical signal. Although the output from the sensor is of few volts, it could be amplified to required voltage using amplifier circuit and could be used for actuating lighting system and the webcam. The lamp and webcam could be turned ON when the PIR sensor is activated and could remain OFF when the sensor is idle. This way, the energy consumed by the overall system could be minimized. Also, the cost of system could be far less than the security system available in the market. With this hypothesis, we have proposed a simple low power PIR based security system.

The system works in the following steps:

i. The software developed is kept running and checks if the cam is turned ON.
ii. When an intruder comes in the detection range of the PIR sensor, the sensor generates an output of 3.3 volts.
iii. This output is further amplified and is used for activating the relay of the lighting system and the webcam.
iv. Once the lamp and webcam are actuated with the output from the amplifier, software finds the webcam is turned ON.
v. The software starts to capture the photo by the esp32 cam.
vi. After the intruder leaves the detection range of the sensor, there is no output from the sensor. Therefore, it turns OFF the webcam. The photo captured will be saved to micro-SD card.
vii. Every time when the intruders come in the detection range of the sensor, the above steps from step 2 to step 6 repeats. PIR sensors have ranges of up to 10 meters (30 feet).

6.1. BLOCK DIAGRAM

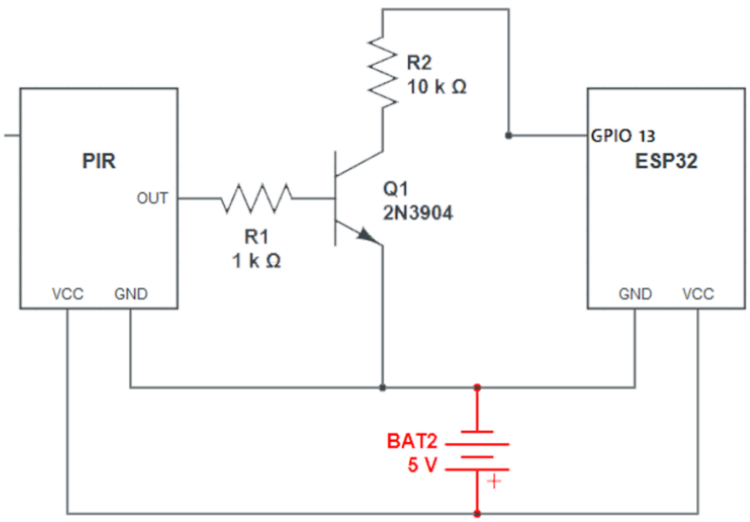

Fig-4

6.2. CIRCUIT DIAGRAM

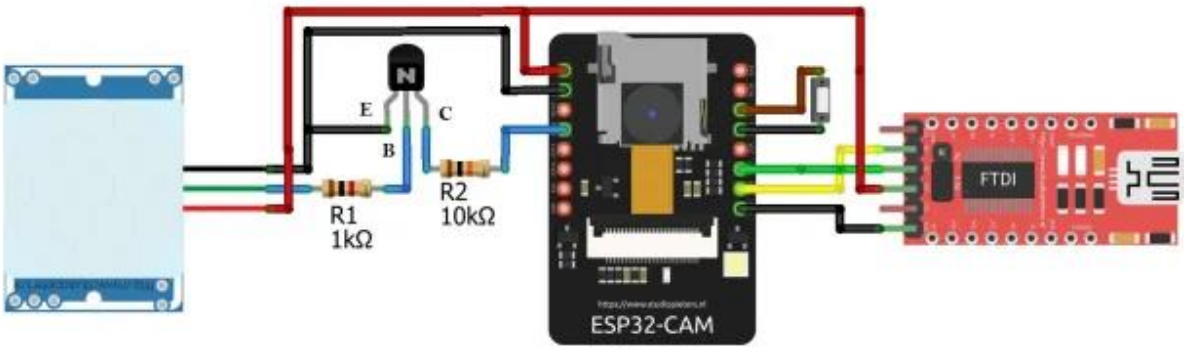

Fig-5

6.3. ESP32 CAM WITH PIR SENSOR ON BREADBOARD

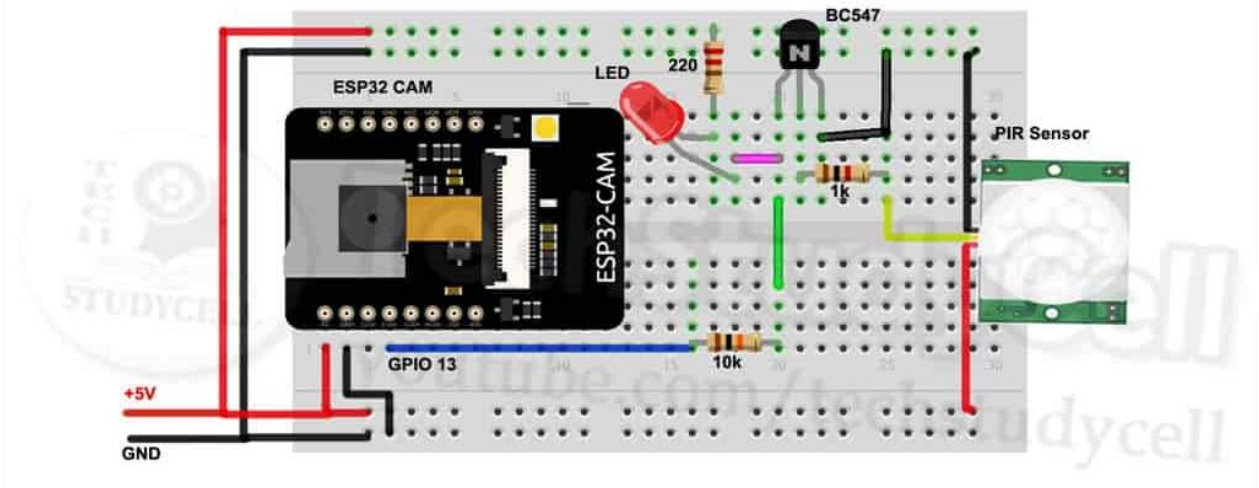

Fig-6

CHAPTER 7

SOFTWARE CODING

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

const char* ssid = "**********";
const char* password = "**********";

String BOTtoken = "**********:**********";
String CHAT_ID = "**********";

bool sendPhoto = false;

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH_LED_PIN 4
bool flashState = LOW;

int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

const int PIRsensor = 13;
const int led = 12;
int PIRstate = LOW;
int val = 0;

const int calibrationTime = 30; // 30 secs

#define PWDN_GPIO_NUM     32
#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

void configInitCamera() {
  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_UXGA;
    config.jpeg_quality = 10;  //0-63 lower number means higher quality
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;  //0-63 lower number means higher quality
    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);
    delay(1000);
    ESP.restart();
  }
```

```
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_CIF);
}

void handleNewMessages(int numNewMessages) {
  Serial.print("Handle New Messages: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID) {
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;
    if (text == "/start") {
      String welcome = "Welcome , " + from_name + "\n";
      welcome += "Use the following commands to interact with the ESP32-CAM \n";
      welcome += "/photo : takes a new photo\n";
      welcome += "/flash : toggles flash LED \n";
      bot.sendMessage(CHAT_ID, welcome, "");
    }
    if (text == "/flash") {
      flashState = !flashState;
      digitalWrite(FLASH_LED_PIN, flashState);
      Serial.println("Change flash LED state");
    }
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("New photo request");
    }
  }
}

String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";

  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
```

```
  if (!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
    ESP.restart();
    return "Camera capture failed";
  }

  Serial.println("Connect to " + String(myDomain));


  if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connection successful");

    String head = "--c010blind3ngineer\r\nContent-Disposition: form-data; name=\"chat_id\";
\r\n\r\n" + CHAT_ID + "\r\n--c010blind3ngineer\r\nContent-Disposition: form-data;
name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
    String tail = "\r\n--c010blind3ngineer--\r\n";

    uint16_t imageLen = fb->len;
    uint16_t extraLen = head.length() + tail.length();
    uint16_t totalLen = imageLen + extraLen;

    clientTCP.println("POST /bot" + BOTtoken + "/sendPhoto HTTP/1.1");
    clientTCP.println("Host: " + String(myDomain));
    clientTCP.println("Content-Length: " + String(totalLen));
    clientTCP.println("Content-Type: multipart/form-data; boundary=c010blind3ngineer");
    clientTCP.println();
    clientTCP.print(head);

    uint8_t *fbBuf = fb->buf;
    size_t fbLen = fb->len;
    for (size_t n = 0; n < fbLen; n = n + 1024) {
      if (n + 1024 < fbLen) {
        clientTCP.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen % 1024 > 0) {
        size_t remainder = fbLen % 1024;
        clientTCP.write(fbBuf, remainder);
      }
    }

    clientTCP.print(tail);

    esp_camera_fb_return(fb);
```

```
    int waitTime = 10000;
    long startTimer = millis();
    boolean state = false;

    while ((startTimer + waitTime) > millis()) {
      Serial.print(".");
      delay(100);
      while (clientTCP.available()) {
        char c = clientTCP.read();
        if (state == true) getBody += String(c);
        if (c == '\n') {
          if (getAll.length() == 0) state = true;
          getAll = "";
        }
        else if (c != '\r')
          getAll += String(c);
        startTimer = millis();
      }
      if (getBody.length() > 0) break;
    }
    clientTCP.stop();
    Serial.println(getBody);
  }
  else {
    getBody = "Connected to api.telegram.org failed.";
    Serial.println("Connected to api.telegram.org failed.");
  }
  return getBody;
}

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

  Serial.begin(115200);

  pinMode(FLASH_LED_PIN, OUTPUT);
  digitalWrite(FLASH_LED_PIN, flashState);

  pinMode(PIRsensor, INPUT);
  pinMode(led, OUTPUT);

  Serial.println("Waiting for the sensor to warm up on first boot");
  delay(calibrationTime * 1000);

  digitalWrite(led, HIGH);
  delay(500);
```

```
  digitalWrite(led, LOW);
  delay(500);
  digitalWrite(led, HIGH);
  delay(500);
  digitalWrite(led, LOW);
  delay(500);
  digitalWrite(led, HIGH);
  delay(500);
  digitalWrite(led, LOW);
  Serial.println("PIR sensor is ACTIVE");

  configInitCamera();

  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("ESP32-CAM IP Address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  val = digitalRead(PIRsensor);

  if (val == HIGH) {
    digitalWrite(led, HIGH);
    if (PIRstate == LOW) {
        Serial.println("Motion detected!");
      delay(500);
      Serial.println("Sending photo to Telegram");
      sendPhotoTelegram();
      PIRstate = HIGH;
    }
  }
  else if (sendPhoto) {
    Serial.println("Preparing photo");
    digitalWrite(FLASH_LED_PIN, HIGH);
    Serial.println("Flash state set to HIGH");
    delay(500);
```

```
  sendPhotoTelegram();
  sendPhoto = false;
  digitalWrite(FLASH_LED_PIN, LOW);
  Serial.println("Flash state set to LOW");
 }
 else if (millis() > lastTimeBotRan + botRequestDelay)  {
  int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  while (numNewMessages) {
    Serial.println("got response");
    handleNewMessages(numNewMessages);
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  }
  lastTimeBotRan = millis();
 }
 else {
  digitalWrite(led, LOW);
  if (PIRstate == HIGH) {
    Serial.println("Motion ended!");
    PIRstate = LOW;
  }
 }
```

PROJECT SETUP

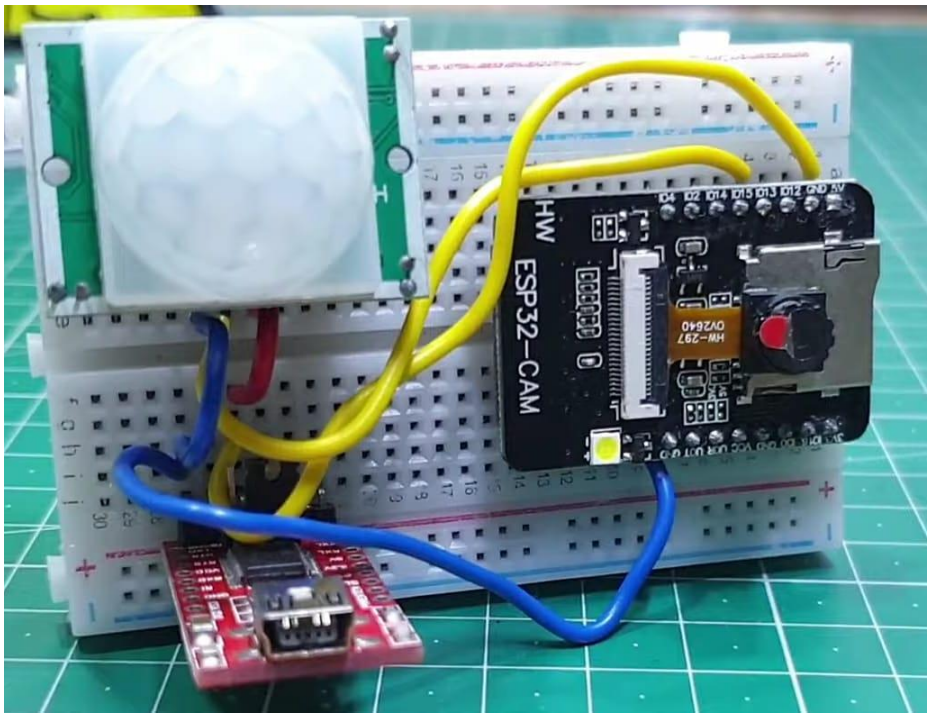

Fig-7

PROJECT OUTPUT

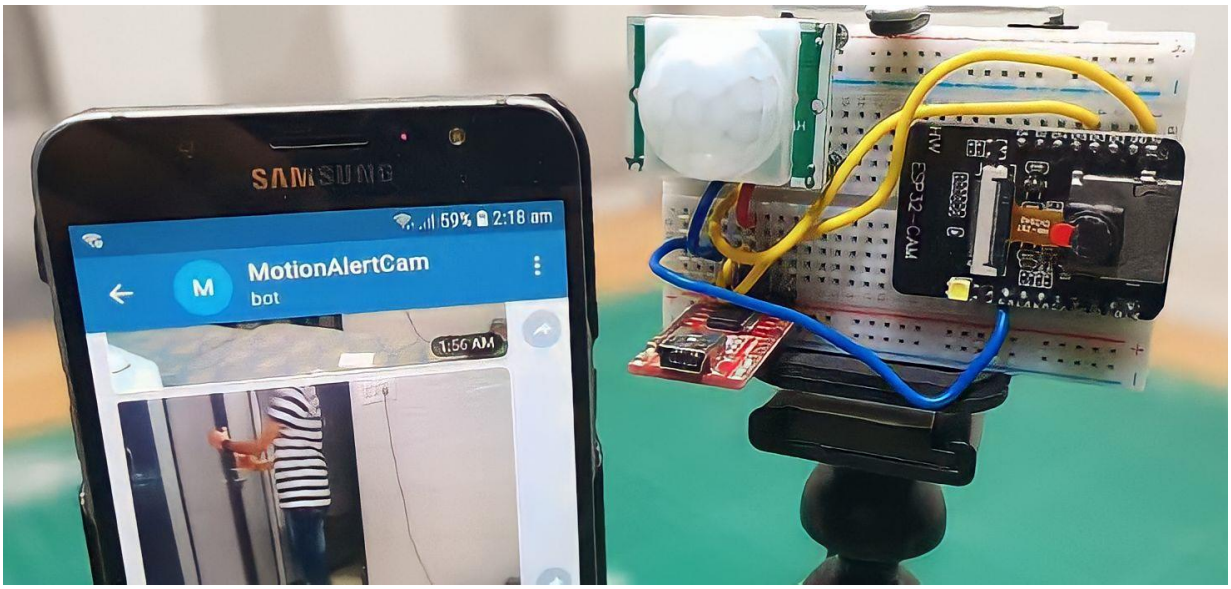

Fig-8

CHAPTER 8

ADVANTAGES

Motion detectors offer several advantages, primarily in security, energy efficiency, and automation. Here are some of the key advantages of using motion detectors:

i. **Enhanced Security:**
   - Motion detectors are commonly used in security systems to detect intruders. When motion is detected, they can trigger alarms, notifications, or activate security cameras.

ii. **Energy Efficiency:**
   - In homes and commercial buildings, motion detectors can be used to control lighting and HVAC systems. Lights and climate control systems can be automatically turned on or off when motion is detected, reducing energy consumption.

iii. **Convenience:**
   - Motion detectors enhance convenience by automating various tasks. For example, they can turn on lights when you enter a room and turn them off when you leave, reducing the need to manually control lighting.

iv. **Safety:**
   - In stairwells, hallways, and other areas, motion detectors can improve safety by ensuring that lighting is available when people enter these spaces, reducing the risk of accidents.

v. **Cost Savings:**
   - Motion-activated lighting and heating/cooling systems can lead to cost savings on energy bills by reducing unnecessary energy consumption.

vi. **Environmental Impact:**
   - Using motion detectors to control lighting and heating/cooling systems can contribute to a reduced carbon footprint and lower environmental impact by conserving energy.

vii. **Extended Device Lifespan:**
   - Motion detectors can help extend the lifespan of devices and equipment by ensuring they are only active when needed, reducing wear and tear.

viii. **Customization and Control:**
   - Many motion detectors have adjustable sensitivity and timing settings, allowing users to customize the device to suit their specific needs and preferences.

ix. **Integration with Home Automation:**
   - Motion detectors can be integrated into smart home systems, allowing users to control and monitor their homes remotely, receive alerts, and automate various tasks.

CHAPTER 9

APPLICATIONS

Motion detectors are versatile devices used in various applications to detect movement and trigger specific responses. Here are some common applications of motion detectors:

1. Home Security Systems:
- Motion detectors are a fundamental component of home security systems. They can trigger alarms, alert homeowners or monitoring services, and activate security cameras when intruders are detected.

2. Outdoor Security Lighting:
- Motion-activated outdoor lights are a popular choice for home security. They illuminate dark areas when motion is detected, deterring potential intruders and providing safety.

3. Automatic Doors:
- Many commercial buildings use motion detectors to automatically open doors when someone approaches, enhancing convenience and accessibility.

4. Smart Lighting Control:
- Motion detectors are used in smart lighting systems to automatically turn lights on and off as people enter or leave a room. This saves energy and enhances convenience.

5. Energy-Efficient HVAC Control:
- In commercial buildings, motion detectors are used to control heating, ventilation, and air conditioning (HVAC) systems, ensuring that they operate only when occupants are present, leading to energy savings.

6. Burglar Alarms:
- In addition to home security systems, motion detectors are commonly used in burglar alarms for businesses and industrial properties to detect unauthorized access.

7. CCTV Systems:
- Motion detectors trigger security cameras to start recording when motion is detected, which is crucial for surveillance and capturing evidence.

8. Automated Hand Sanitizer Dispensers:
- In public spaces, especially during health crises, motion detectors can activate hand sanitizer dispensers, promoting hygiene.

9. Automatic Faucets and Soap Dispensers:
- In restrooms, motion detectors are used to activate faucets and soap dispensers, reducing the need for touching surfaces and promoting cleanliness.

CHAPTER 10

FUTURE ENHANCEMENT / SCOPE

The field of motion detection is continuously evolving, and there are several future enhancements and trends that can be expected in motion detectors:

1. Advanced Sensing Technologies:

- Continued advancements in sensor technologies, such as improved PIR sensors, radar sensors, and LiDAR (Light Detection and Ranging), will enhance the accuracy and range of motion detection.

2. AI and Machine Learning Integration:

- Motion detectors will increasingly incorporate artificial intelligence (AI) and machine learning algorithms for more intelligent and context-aware motion analysis. This will reduce false alarms and enhance detection capabilities.

3. Facial Recognition:

- Some motion detectors may include facial recognition technology to identify individuals, making them useful for access control and personalized automation.

4. Privacy Features:

- Enhanced privacy features will allow users to control what data is captured and stored, addressing concerns related to privacy and data security.

5. Energy-Efficient Designs:

- Motion detectors will continue to be designed for improved energy efficiency, extending battery life in wireless devices and reducing power consumption in wired systems.

6. Integration with Smart Home Ecosystems:

- Enhanced compatibility and integration with smart home systems, voice assistants, and IoT devices will provide users with seamless control and automation options.

7. Multi-Modal Sensing:

- Combining multiple sensor types, such as PIR, microwave, and camera sensors, will provide a more comprehensive view of the environment, reducing false positives and enhancing security.

8. Cloud Connectivity:

- Motion detectors will increasingly connect to the cloud for remote monitoring and data storage, allowing users to access motion events and analytics from anywhere.

9. Wireless Connectivity:

- The adoption of wireless communication technologies like 5G, LoRa, and NB-IoT will enable motion detectors to be deployed in more locations without wiring constraints.

CHAPTER 11

CONCLUSION

Motion capture (Mo cap) is an effective 3D animation tool for realistically capturing human motion. In this PIR Sensor Based Security System, we have used low power, low cost PIR sensor that are easy to interface with other components. By using this system, we were able to reduce the power consumed and memory space of the system.

Currently, we have used only one webcam in our project which could only capture the area facing to it. The software developed is for the recording of the photo captured by the web cam.

REFERENCES

[1].http://ieeexplore.ieee.org/document/8256877/
[2].https://ieeexplore.ieee.org/abstract/document/9673226/
[3].https://randomnerdtutorials.com/esp32-cam-pir-motion-detector-photo-capture/
[4].https://easyelectronicsproject.com/esp32-projects/esp32-cam-pir-motion-detector with-photo-capture-1/
[5].https://www.researchgate.net/publication/347062859_Smart_Home_Monitoring_System_Using_ESP32_Microcontrollers

[6].https://youtu.be/KTRwBBLEsXg

[7].https://youtu.be/LBoM_Uoq_nA