TITLE: MOTION DETECTOR USING ESP32 CAM

by

NAME: S. Patrick Raja REGISTER NUMBER: 20BCE1058
NAME: Shashvath Radhakrishnan REGISTER NUMBER: 20BCE1081
NAME: Riyasshri JS REGISTER NUMBER: 20BCE1836

A project report submitted to

Dr. Muthulakshmi S

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

MICROPROCESSOR AND INTERFACING (CSE2006)

in

B. Tech. COMPUTER SCIENCE AND ENGINEERING

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur – Kelambakkam Road

Chennai – 600127

APRIL 2022

BONAFIDE CERTIFICATE

Certified that this project report entitled "**MOTION DETECTOR USING ESP32 CAM"** is a bonafide work of **NAME: S. PATRICK RAJA REG.NO: 20BCE1058, NAME: SHASHVATH RADHAKRISHNAN REG.NO: 20BCE1081 AND NAME: RIYASSHRI JS REG.NO: 20BCE1836** who carried out the Project work under my supervision and guidance for **CSE2006-MICROPROCESSOR AND INTERFACING**

Dr. Muthulakshmi S

Associate Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

• Motion alert system is a very useful product to avoid intruders from stealing something.

• This project uses PIR sensor to detect heat energy from the surrounding and signals ESP32 cam when it sense motion, then ESP32 cam will send a image to the telegram bot using the wi-fi connection to which it is connected.

• The ESP32 cam is programmed using TTL programmer and using Arduino IDE.

• Essential libraries are downloaded in the Arduino IDE.

• The project will work in dark also but we will not get satisfied images.

• The ESP32 and smartphone don’t need to be connected in the same network.

• The ESP32 will interact with the Telegram bot to send messages to your telegram account. Whenever motion is detected, you’ll receive a notification in your smartphone (as long as you have access to the internet)

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Muthulakshmi S,** Associate Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

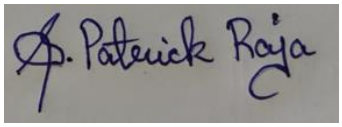
We are extremely grateful to **Dr. Sivasubramanian. A,** Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

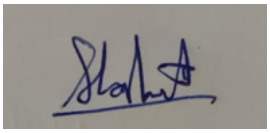
We express our thanks to our Head of the Department **Dr. Vetrivelan. P** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

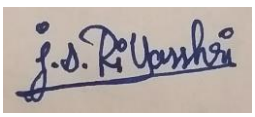
NAME: PATRICK RAJA **NAME: SHASVATH RADHAKRISHNAN**





NAME WITH SIGNATURE **NAME WITH SIGNATURE**

NAME: RIYASSHRI J S



NAME WITH SIGNATURE

TABLE OF CONTENTS

SERIAL NO.		TITLE	PAGE NO.
		ABSTRACT	
		ACKNOWLEDGEMENT	
1		INTRODUCTION	
	1.1	OBJECTIVES AND GOALS	
	1.2	APPLICATIONS	
	1.3	FEATURES	
2		DESIGN	
	2.1	BLOCK DIAGRAM	
	2.2	HARDWARE ANALYSIS	
	2.3	(SNAPSHOTS-PROJECT , TEAM, RESULTS)	
3	3.1	SOFTWARE –CODING AND ANALYSIS (SNAPSHOTS OF CODING AND RESULTS)	
4		CONCLUSION AND FUTURE WORK	
	4.1	RESULT, CONCLUSION AND INFERENCE	
	4.2	FUTURE WORK COST	
5		REFERENCES	
6		PHOTO GRAPH OF THE PROJECT ALONG WITH THE TEAM MEMBERS	

1. INTRODUCTION

1.1 OBJECTIVES AND GOALS:

To send an image to the telegram bot whenever motion is detected. To find any intruder trying to access any valuable things. This can be set such that no one would notice. The PIR sensor sense the motion and the ESP32 send an image which is dark but the owner can be alerted when it sends image.

1.2 APPLICATION:

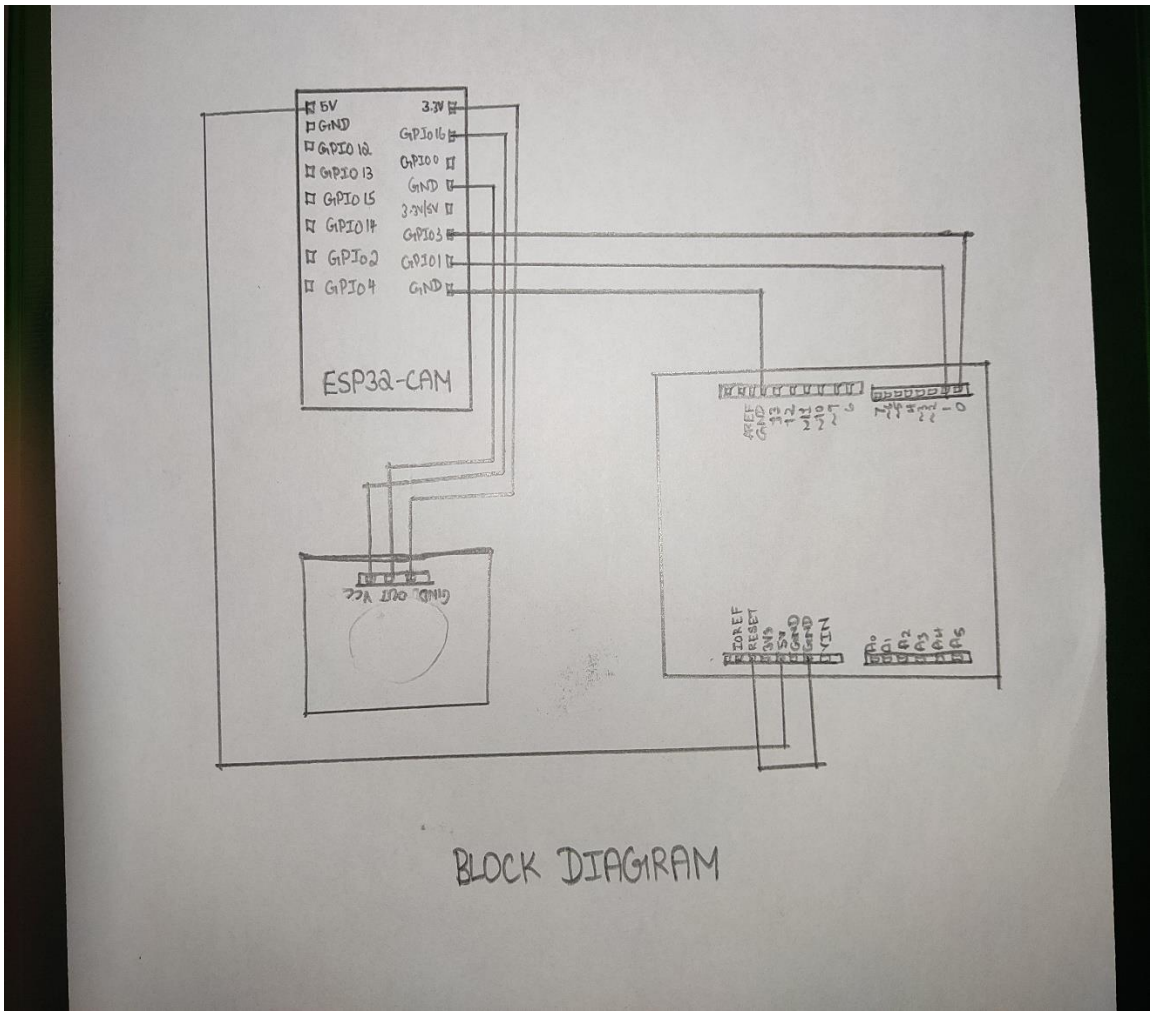
The main application of this project is finding any intruder who tries to access or steal valuable items. This can be also used to keep track on the animals living in forest.

1.3 FEATURES:

- The project send pics directly to the owners telegram which makes it easy to access.
- The project uses ESP32 cam which has the wifi module that can send information through internet all over the world.
- The project uses a Arduino UNO board which is used for uploading code. In the program the ssid and password of the wifi is given which helps it to connect to the wifi.

2. DESIGN

2.1 BLOCK DIAGRAM:



2.2 HARDWARE ANALYSIS:

HARDWARE REQUIREMENTS:

The hardware that are required for our projects are

- Arduino UNO
- PIR sensor
- ESP32 cam
- Jumper wires
- USB cable to connect Arduino UNO to Laptop
- A Mobile Phone with Telegram installed

ARDUINO UNO:

Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. Arduino UNO features AVR microcontroller Atmega328, 6 analogue input pins, and 14 digital I/O pins out of which 6 are used as PWM output. This board contains a USB interface i.e. USB cable is used to connect the board with the computer and Arduino IDE (Integrated Development Environment) software is used to program the board. The unit comes with 32KB flash memory that is used to store the number of instructions while the SRAM is 2KB and EEPROM is 1KB. The operating voltage of the unit is 5V which projects the microcontroller on the board and its associated circuitry operates at 5V while the input voltage ranges between 6V to 20V and the recommended input voltage ranges from 7V to 12V.

ARDUINO UNO

PIR SENSOR:

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

PIR SENSOR

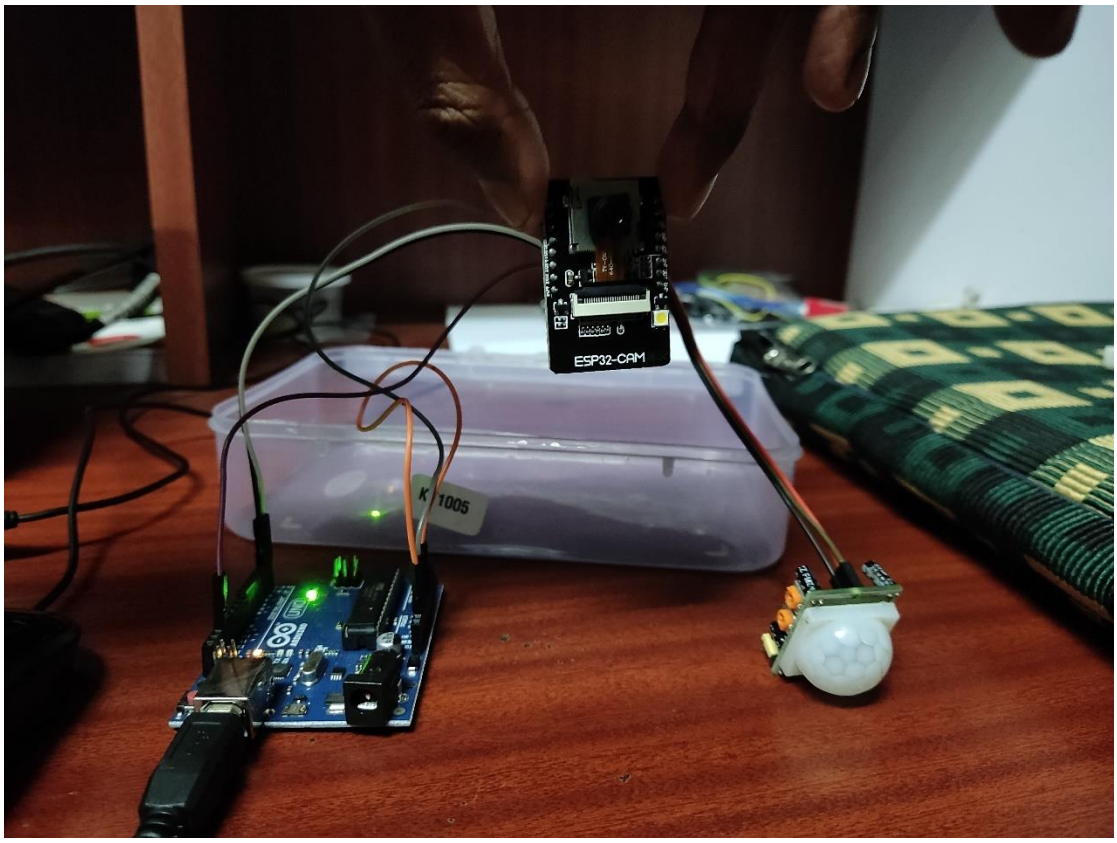
ESP32 CAM:

The ESP32-CAM is a full-featured microcontroller that also has an integrated video camera microSD card socket. It's a powerful yet inexpensive microcontroller from Espressif and A-Thinker with advanced features like Bluetooth, WIFI and multipurpose GPIO ports. It's inexpensive and easy to use, and is perfect for IoT devices requiring a camera with advanced functions like image tracking and recognition.

ESP32 CAM

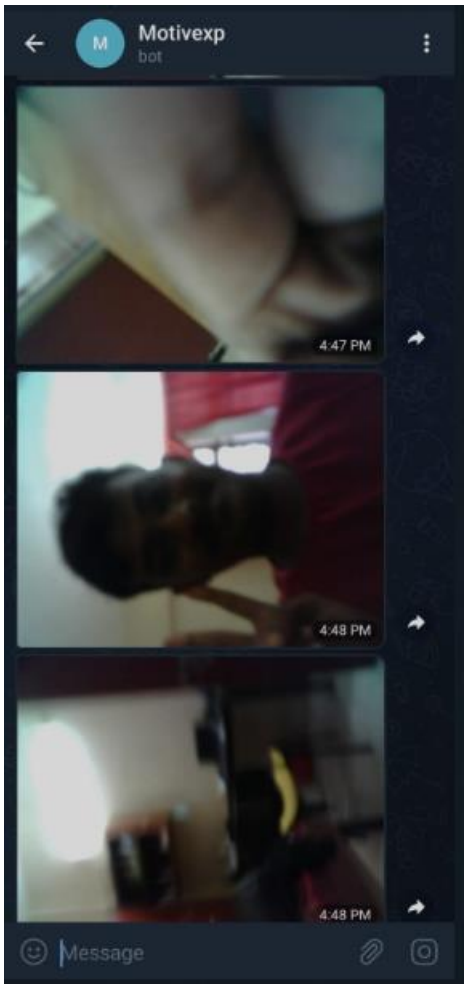
2.3 (SNAPSHOTS-PROJECT, TEAM, RESULTS)

PROJECT SNAPSHOT:



RESULTS:

PICTURES SENT TO TELEGRAM BOT NAMED MOTIVEXP AFTER MOTION IS DETECTED:



3. SOFTWARE ANALYSIS

3.1 SOFTWARE –CODING AND ANALYSIS

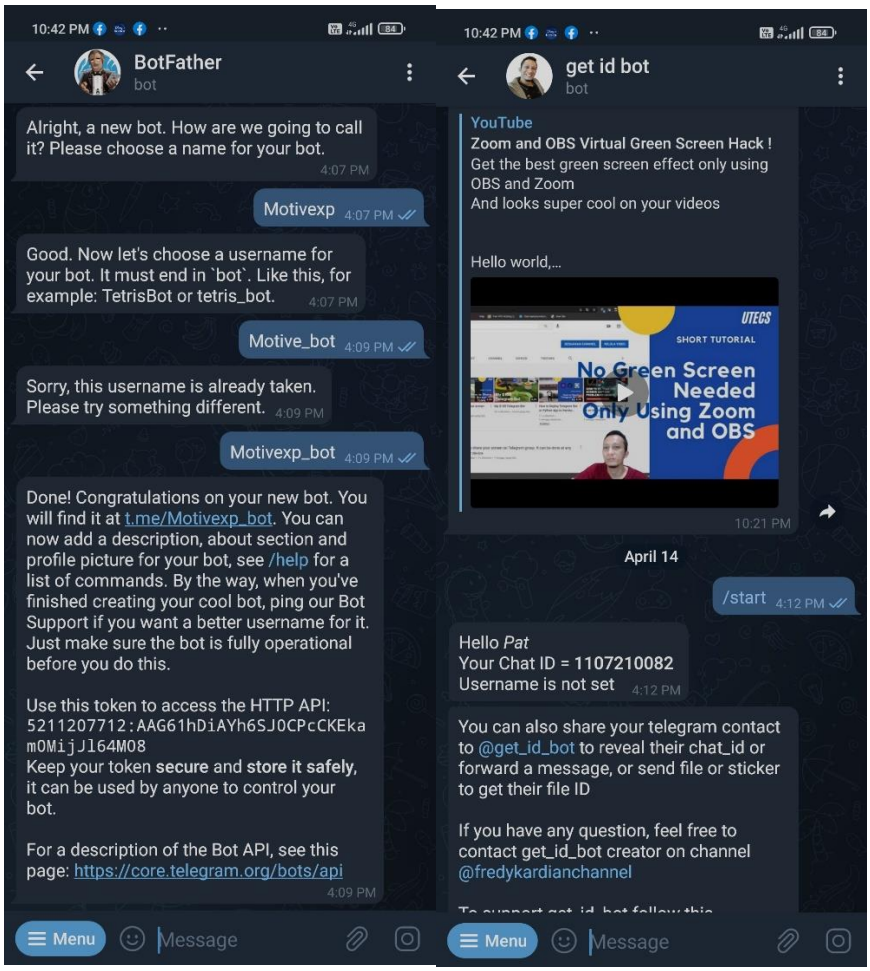
SOFTWARE REQUIREMENTS:

The software requirements for our projects are

- Arduino IDE
- Telegram app installed in a mobile phone

CREATING TELEGRAM BOT:

This is a feature available in Telegram not in Whatsapp. Using which we can create bot and get a chat id and token for the bot which is used in the program to connect to the bot.



CODE FOR ACCESSING THE ESP32 FEATURES:

```
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//            Ensure ESP32 Wrover Module or other board with PSRAM is selected
//            Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
```

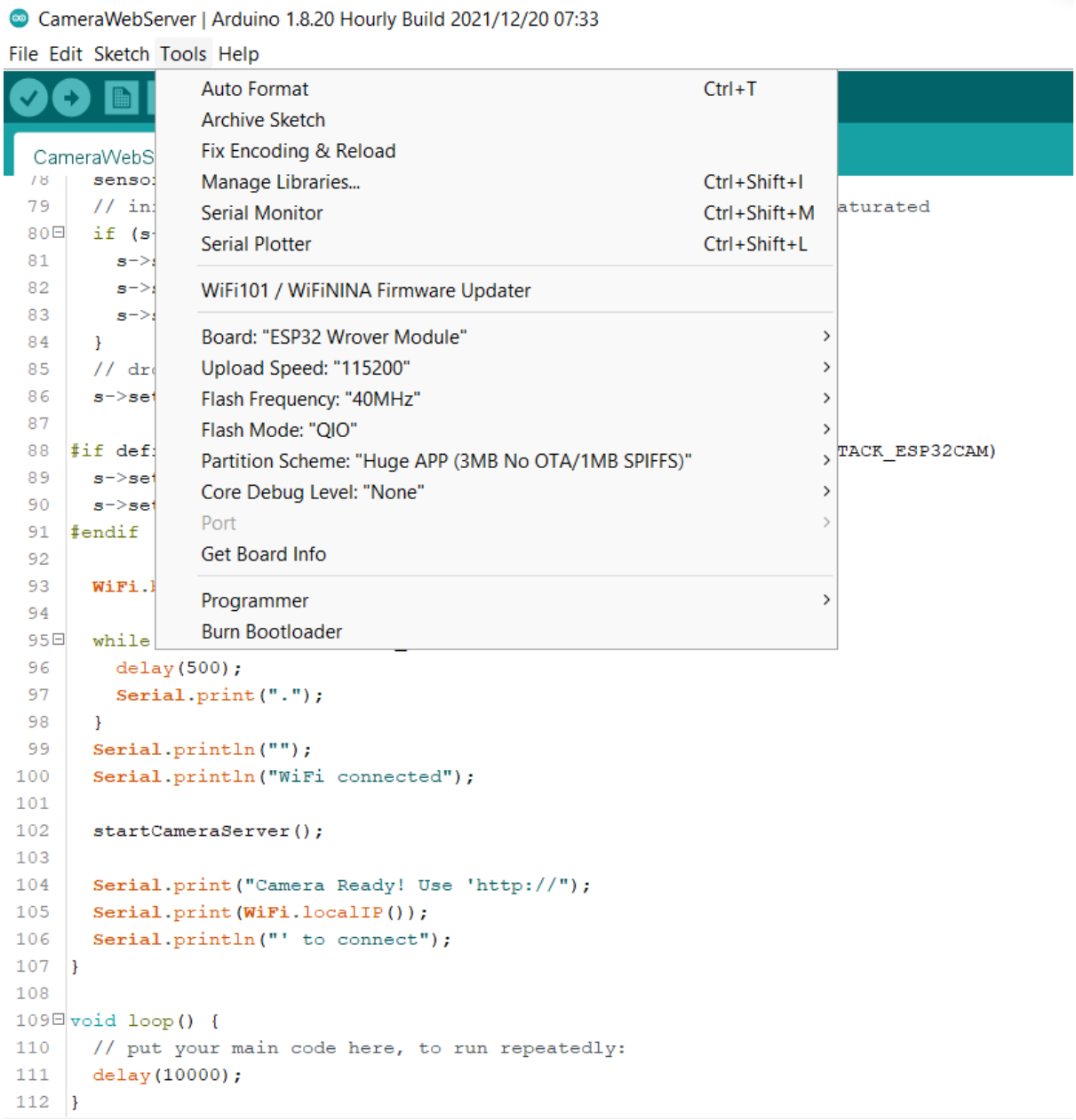
```
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "POCO X3 Pro";
const char* password = "qwerty666";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  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 PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                      for larger pre-allocated frame buffer.
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
```

```
    config.fb_count = 1;
  }

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}

void loop() {
```

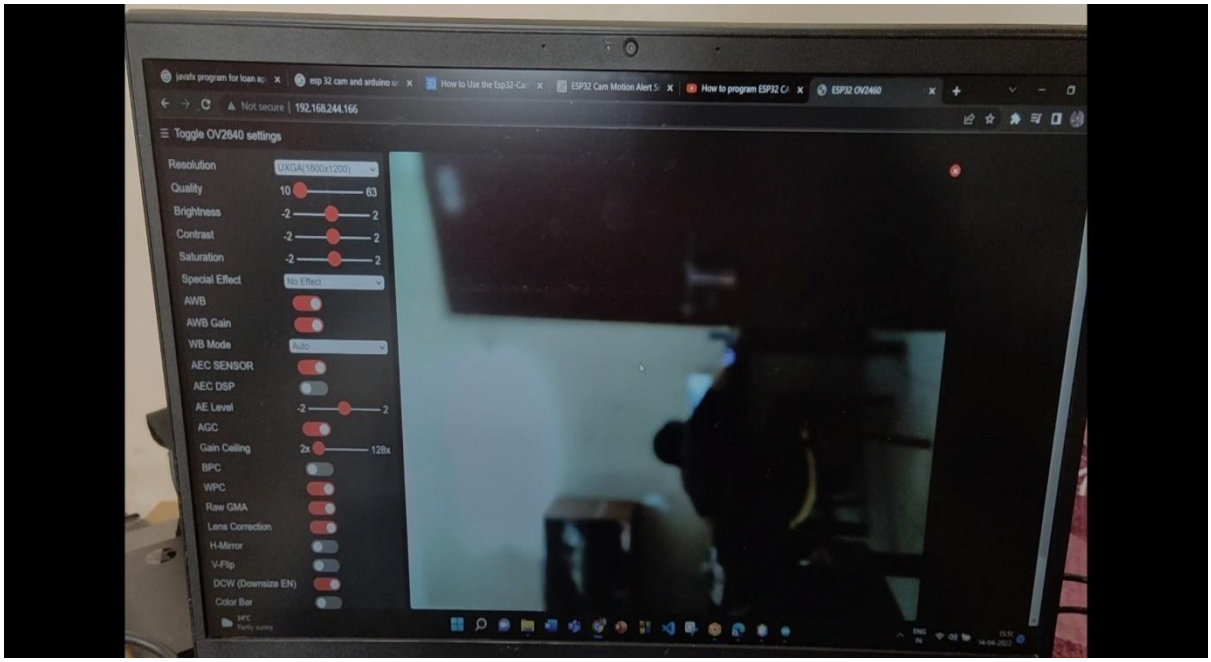
```
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

This code creates a port address which can be opened in our browser from where we can access the features of the ESP32 CAM

TOOLS SETTINGS WILL UPLOADING CODE:



PICTURE OF STIMULATION:



After setting up the features of the ESP32 CAM the real project code is uploaded.

PROJECT CODE:

```
const char* ssid     = "POCO X3 Pro";   //WIFI SSID
const char* password = "qwerty666";   //WIFI password
String token = "5211207712:AAG61hDiAYh6SJ0CPcCKEkam0MijJl64M08";
String chat_id = "1107210082";

#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"

//CAMERA_MODEL_AI_THINKER
#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

int gpioPIR = 13;   //PIR Motion Sensor

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

  Serial.begin(115200);
  delay(10);
  WiFi.mode(WIFI_STA);
  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  long int StartTime=millis();
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    if ((StartTime+10000) < millis()) break;
  }

  Serial.println("");
  Serial.println("STAIP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("");

  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Reset");

    ledcAttachPin(4, 3);
    ledcSetup(3, 5000, 8);
    ledcWrite(3,10);
    delay(200);
    ledcWrite(3,0);
    delay(200);
    ledcDetachPin(3);
    delay(1000);
    ESP.restart();
  }
  else
  {
```

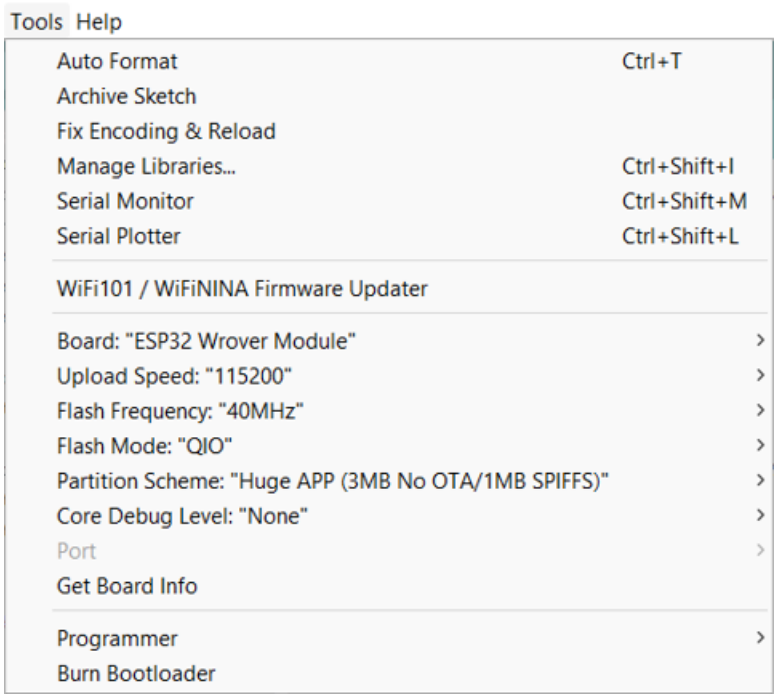
```
    ledcAttachPin(4, 3);
    ledcSetup(3, 5000, 8);
    for (int i=0;i<5;i++) {
      ledcWrite(3,10);
      delay(200);
      ledcWrite(3,0);
      delay(200);
    }
    ledcDetachPin(3);
  }

  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;  //0-63 lower number means higher quality
    config.fb_count = 2;
  }
else
{
    config.frame_size = FRAMESIZE_QQVGA;
    config.jpeg_quality = 12;  //0-63 lower number means higher quality
    config.fb_count = 1;
  }

  // camera init
  esp_err_t err = esp_camera_init(&config);
```

```
  if (err != ESP_OK)
{
    Serial.printf("Camera init failed with errors 0x%x", err);
    delay(1000);
    ESP.restart();
  }

sensor_t * s = esp_camera_sensor_get();
 s->set_framesize(s, FRAMESIZE_XGA);

}

void loop()
{

  pinMode(gpioPIR, INPUT_PULLUP);
  int v = digitalRead(gpioPIR);
  Serial.println(v);
  if (v==1)
  {
    alerts2Telegram(token, chat_id);
    delay(10000);
  }
  delay(1000);

}

String alerts2Telegram(String token, String chat_id)
{
  const char* myDomain = "api.telegram.org";
  String getAll="", 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";
  }


WiFiClientSecure client_tcp;

  if (client_tcp.connect(myDomain, 443))
{
    Serial.println("Connected to " + String(myDomain));
```

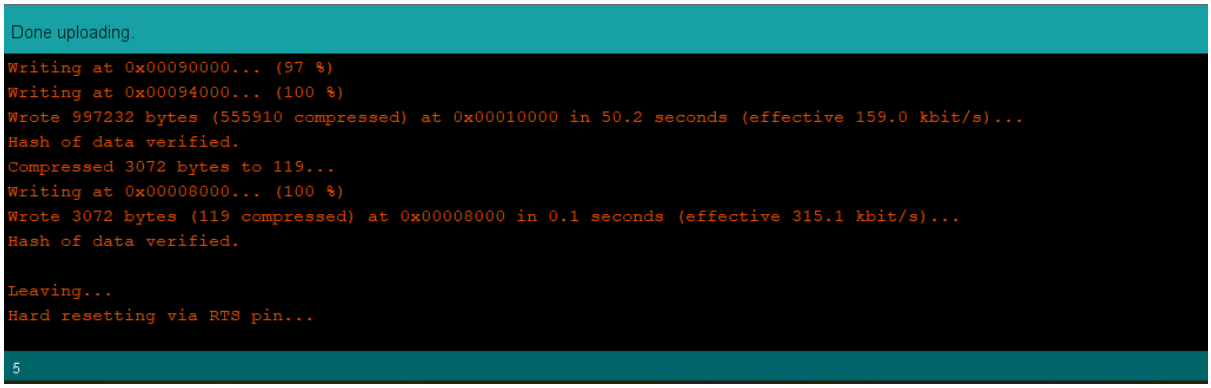
```
    String head = "--India\r\nContent-Disposition: form-data;
name=\"chat_id\"; \r\n\r\n" + chat_id + "\r\n--India\r\nContent-Disposition:
form-data; name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type:
image/jpeg\r\n\r\n";
    String tail = "\r\n--India--\r\n";

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

    client_tcp.println("POST /bot"+token+"/sendPhoto HTTP/1.1");
    client_tcp.println("Host: " + String(myDomain));
    client_tcp.println("Content-Length: " + String(totalLen));
    client_tcp.println("Content-Type: multipart/form-data; boundary=India");
    client_tcp.println();
    client_tcp.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)
{
        client_tcp.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen%1024>0)
{
        size_t remainder = fbLen%1024;
        client_tcp.write(fbBuf, remainder);
      }
    }

    client_tcp.print(tail);

    esp_camera_fb_return(fb);

    int waitTime = 10000;   // timeout 10 seconds
    long startTime = millis();
    boolean state = false;

    while ((startTime + waitTime) > millis())
    {
      Serial.print(".");
      delay(100);
```

```
      while (client_tcp.available())
      {
          char c = client_tcp.read();
          if (c == '\n')
          {
            if (getAll.length()==0) state=true;
            getAll = "";
          }
          else if (c != '\r')
            getAll += String(c);
          if (state==true) getBody += String(c);
          startTime = millis();
       }
       if (getBody.length()>0) break;
    }
    client_tcp.stop();
    Serial.println(getBody);
  }
  else {
    getBody = "Connection to telegram failed.";
    Serial.println("Connection to telegram failed.");
  }

  return getBody;
}
```

TOOLS SETTING WHILE UPLOADING:

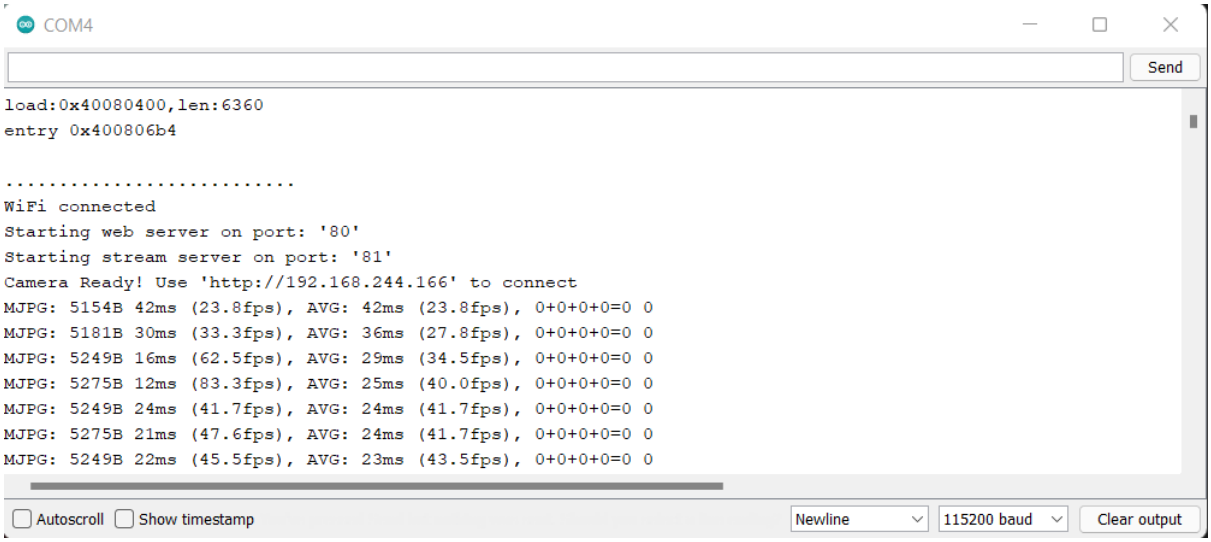


UPLOADING CODE:

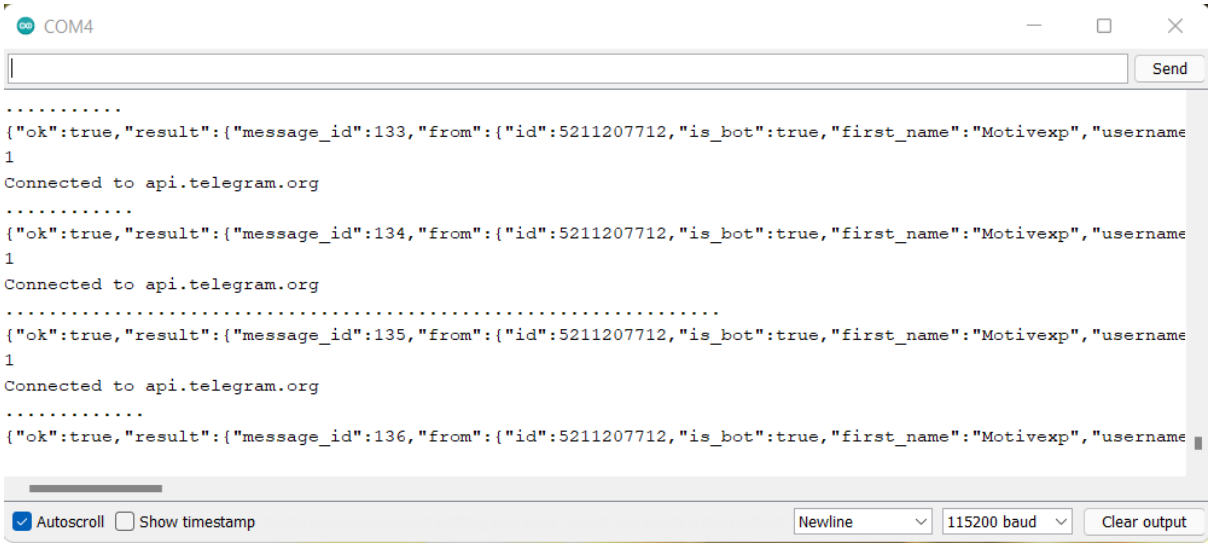


SERIAL MONITOR OUTPUTS:

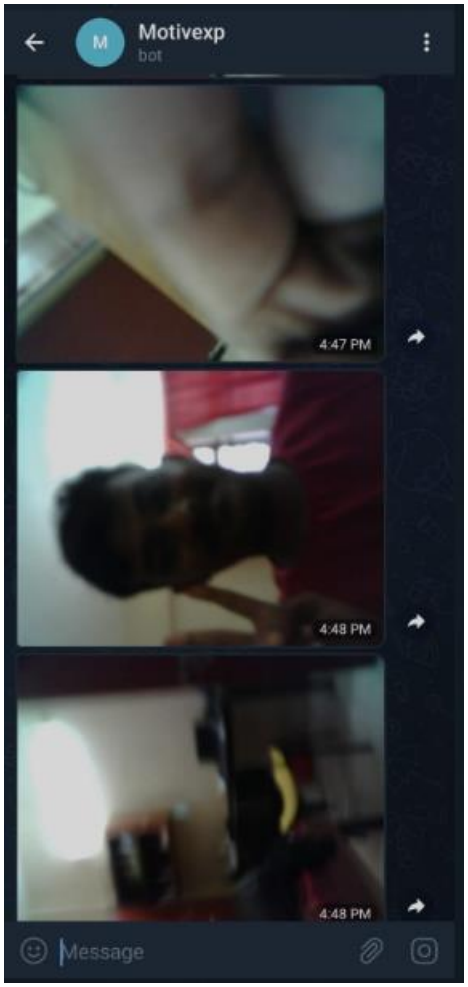
WIFI CONNECTION:



CONNECTING TO TELEGRAM AND SENDING PICTURES TO MOTIVEXP TELEGRAM BOT:



PICTURES SENT TO TELEGRAM BOT:



FUTURE WORK AND COST:

Future extendibility

Among many possible future applications of this project, these two applications will possibly be the most relevant in today's world

Increasing the scale of our current project will find its space in industries and office spaces, the increased range and precision of the sensor will be perfect for commercial building security. The bigger scale can also be a vital addition for wildlife monitoring, the cameras can capture feedback only if movement is triggered and a sensor with increased rage will definitely be needful to cover vast forest areas

The next possible extension would be a fully connected smart home security system, it'll consist of multiple sensors throughout the home centrally connected to ensure reduction in redundancy.

Currently the project costs about Rs 2000 to assemble further extension and improvement of the project may cost 5000 for each component to monitor larger area like forests more number of sensors and components are required.

INFERENCE:

we were able to implement Arduino and ESP32 CAM and programme them using sketch language with the help Arduino IDE.

CONCLUSION :

Thus we have successfully implemented motion detector using Arduino PIR sensor and ESP32 CAM and the results were posted.

REFERENCES

Reference journals :

Arduino controlled motion sensor, *September 2018,* https://www.instructables.com/id/Arduino-Controlled-Motion-Sensor/.

https://www.ijsdr.org/papers/IJSDR1605022.pdf

International journals :

Soleh, S. S. S. M., Som, M. M., Abd Wahab, M. H., Mustapha, A., Othman, N. A., & Saringat, M. Z. (2018, November). Arduino-based wireless motion detecting system. In *2018 IEEE Conference on Open Systems (ICOS)* (pp. 71-75). IEEE.

International Conference :

International Conference on Advances in Computing and Communication Engineering (ICACCE)

Date of Conference: 22-23 June 2018

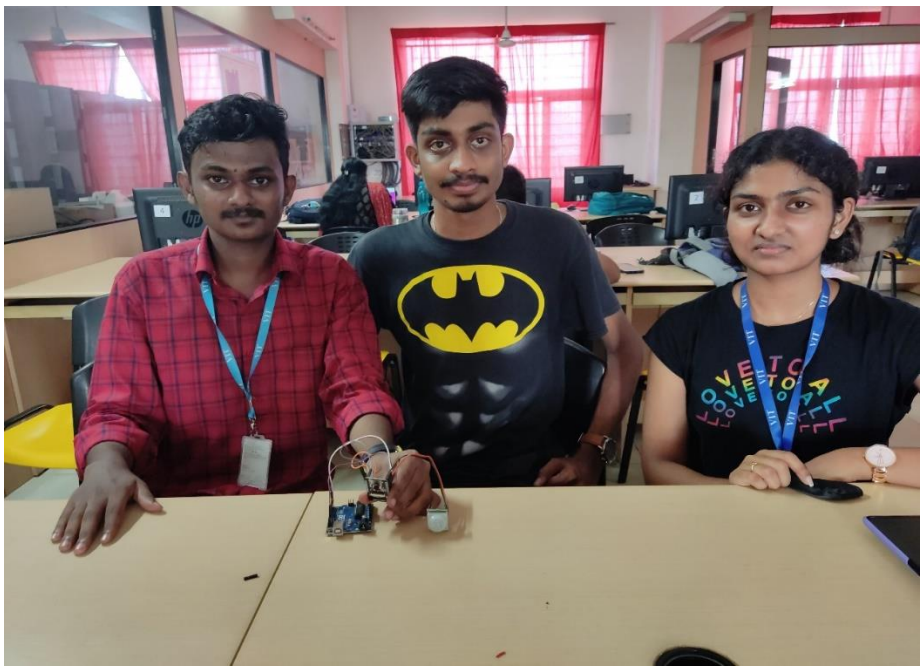
Reference sites :

https://randomnerdtutorials.com/telegram-esp32-motion-detection-arduino/

https://www.peertechzpublications.com/articles/ARA-4-106.php#:~:text=The%20PIR%20sensor%20is%20mainly,are%20exposed%20to%20InfraRed%20radiation

https://maker.pro/arduino/projects/arduino-motion-detector-using-pir-sensor

SNAPSHOT OF THE TEAM WITH THE PROJECT:



BIODATA

Name : S PATRICK RAJA

Mobile Number : 9487582209

E-mail : patrickraja.s2020@vitstudent.ac.in

Permanent Address: 18-7-11/1, Kamaraj Nagar, Surandai, Tenkasi

Name : SHASVATH RADHAKRISHNAN

Mobile Number : 6383459689

E-mail : shashvath.radhakrish2020@vitstudent.ac.in

Permanent Address: 3a Mithila, Sai brindavan apt, ramakrishnnagar 2nd main road, adambakkam Chennai 88

Name : RIYASSHRI J S

Mobile Number : 9361235894

E-mail : riyasshri.js2020@vitstudent.ac.in

Permanent Address: 52B / 28b JEEVAN BIMA NAGAR, ANNA NAGAR WEST EXTENSION CHENNAI -600101