**SMART HOME SECURITY SYSTEM**

A report submitted by-

1. MANASJYOTI DAS

2. ARINDAM HAZARIKA

3 .MANISH KAKATI

4. SAMUDRA DAIMARY

5. VINAYAK NATH

*Guided by*

**DR. BARNALI GOSWAMI**

**Assistant Professor Department of Electrical Engineering**



**Department of Electrical Engineering**

ASSAM ENGINEERING COLLEGE, JALUKBARI

**ACKNOWLEDGEMENT**

With sincere gratitude, we express our earnest thanks to Dr. Barnali Goswami for her constant involvement, energetic efforts, and proficient guidance throughout the progress of the project. Without her wise counsel and encouragement, it would have been impossible for us to complete this project work in proper time and form. Their guidance and counsel kept us motivated throughout the entire duration of our project work.

Finally, we would also love to thank our fellow friends, from other microprocessor and microcontroller groups with whom we had numerous discussions on the domain, and with whom we faced numerous challenges together in learning, and understanding the working of microcontrollers and microprocessors.

**TABLE OF CONTENTS**

**Title Page no**

1. **Abstract 3**
2. **List of Figures 4**
3. **Introduction 5-6**
4. **Hardware Description 7**

**4.1 Circuit Diagram 8-9**

* 1. **Components Used 10**

1. **Software used**
2. **Proposed methodology**
3. **Conclusion**
4. **Reference**
5. **Appendix**

1. **ABSTRACT**

In this work, the design and implementation of a microcontroller-based home security system with GSM technology have been presented and analysed. One microcontroller with other peripheral devices which include Radar Sensor, PIR sensor, LDR sensor, Buzzer, and Global System for Mobile Communication (GSM) Module is responsible for the reliable operation of the proposed security system. In addition, a mobile phone is interfaced with a microcontroller through wifi connectivity to control the system. At last, the results of the practical circuit show the proper functions and also verify the reliable security within a reasonable cost.

This security system based on ESP32 camera module detects any kind of suspicious activity inside the place. As such the sensors fitted along with it detects the activity thereby activating signals which are sent towards the microcontroller which again activates an alarming signal along with switching on and off the light so that the user becomes aware of the activity inside the room. Also with the creation of some third party applications we can also get the pictures of the suspicious activity.

1. **List of Figures**

**Title Page no.**

Fig 4.1.1 – Pinout Diagram of ESP32 CAM

Fig 4.1.2 – Circuit diagram

Fig 4.2.1 - ESP 32 camera module

Fig 4.2.2. 1CH Relay Module

Fig 4.2.3- RCWL Radar Sensor

Fig 4.2.4- PIR sensor

Fig 4.2.5- LDR Sensor

Fig 4.2.6- TTL module

Fig 4.2.7– Buzzer

Fig 4.2.8- Bread Board

Fig 4.2.9- AC Bulb

Fig 4.2.10- Servo motor

1. **INTRODUCTION**

Throughout the history of mankind, the importance of security has always been a great matter of concern. Home automation refers to a domestic environment that improves the quality of the resident's life by facilitating a flexible, comfortable, healthy, and safe environment. With the rapid development of computers and networks, Internet-based home security has advanced a lot in residential areas. Home automation is a process for improving the quality of resident's life by facilitating a flexible, comfortable, and secure environment. The home security system is the most prominent feature of home automation. Traditional techniques of alarm-based security have gained much popularity in past decades. An integrated security system is a centralized platform combining various security parameters such as a wireless alarm system, access control devices, and video surveillance. Being integrated can allow the information to be expedited and facilitate a better response across all systems. For high-quality home automation, the selection of sensors is very important. A control system is good if the sensors used to measure the desired variables can transmit the measured values.

Automatic anti-theft techniques are important applications used in almost all modern gadgets and smart homes. The system for integrated security monitoring automatically is achieved by using ESP32 camera module based microcontroller system. ESP32 due to its increased popularity finds its varied range of applications. Motion detecting sensor PIR, the light-detecting LDR sensor, a buzzer, and ESP32 camera module with the use of some relay is the hardware used to interface with the computer, and then any kind of suspicious activity is detected in the room.

The microcontroller-based home security system has become so important that it acts as a benchmark for testing and simulation of particular sensors for detection and monitoring of various situations happening around the house. Various types of projects like minor projects as well as major projects are carried out on suitable hardware and software platforms.

ESP32 camera module takes the help of some third party applications available in the telegram software known as ‘ BOTS ‘. These bots can control the camera module as well as all the sensors and also ensuring rotation of the camera for proper coverage of the area.

1. **Hardware Description**
   1. Circuit Diagram

The ESP32-CAM is a full-featured microcontroller that also has an integrated video camera and microSD card socket.  It’s inexpensive and easy to use, and is perfect for IoT devices requiring a camera with advanced functions like image tracking and recognition. The figure below shows the pinout of the ESP32 CAM.

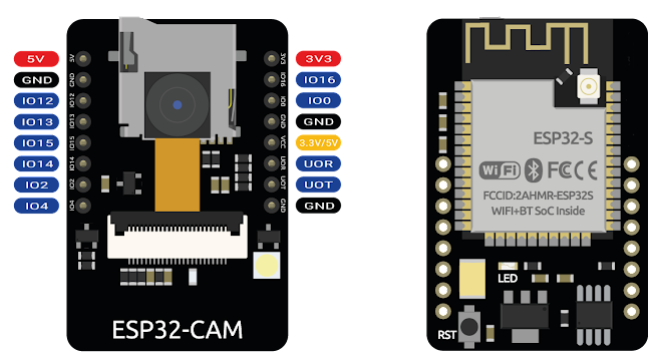


Fig 4.1.1 Pinout Diagram of ESP32 CAM

The pins are used for input and output as follows:

5V - to 5v supply

GND - to Ground

IO12 - to Radar Digital output

IO13 to - to PIR out

IO15 - to LDR out

IO14 and IO2 - to 2ch Relay

IO4 - to Buzzer

IO16 - to Servo in

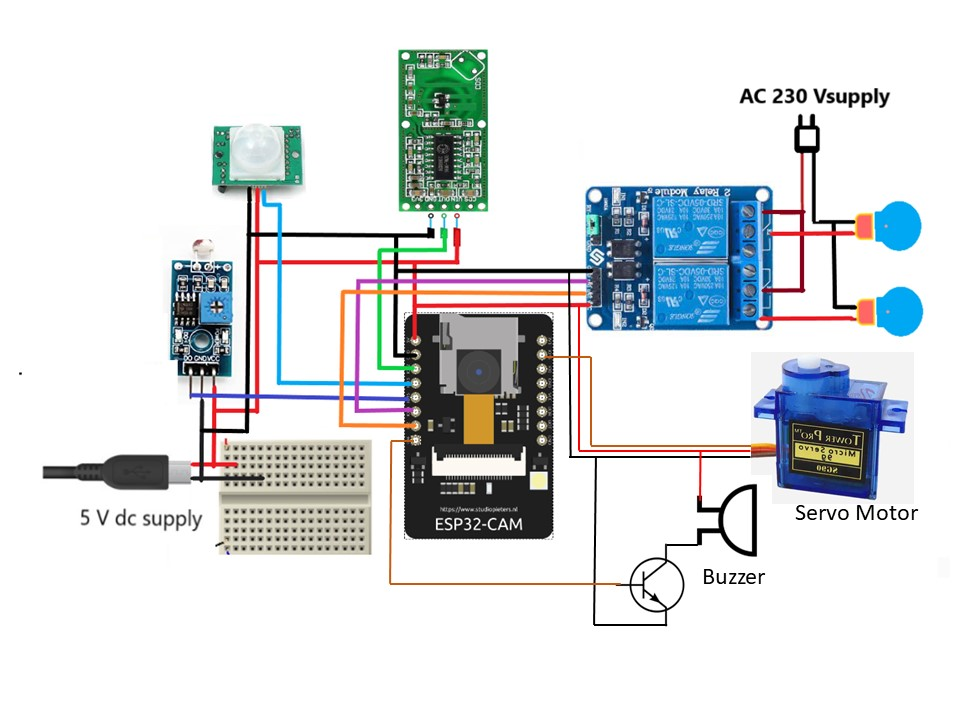


Fig 4.1.2 – Circuit diagram

* 1. **COMPONENTS USED**
* **ESP 32 Camera**

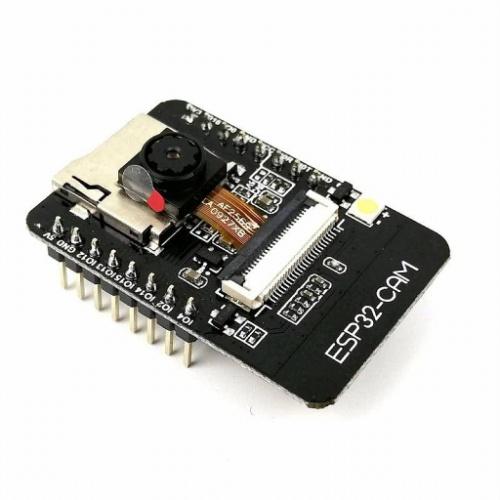


Fig4.2.1 - ESP 32 camera module

Using the ESP 32 camera module, the need for a microcontroller was eliminated and hence the cost of making was reduced. The ESP32-CAM provides an inexpensive way to build more advanced home automation projects that feature video, taking photos, and face recognition. The ESP32-CAM is a full-featured microcontroller that also has an integrated video camera and micro SD card socket.  It’s inexpensive and easy to use, and is perfect for IoT devices requiring a camera with advanced functions like image tracking and recognition.

* 1CH RELAY MODULE



Fig 4.2.2 - 1CH Relay Module

The Single Channel Relay Module is a convenient board that can be used to control high voltage, and a high current load such as motor, solenoid valves, lamps, and AC load. It is designed to interface with a microcontroller such as Arduino, PIC, etc.

* Radar Sensor



Fig1.4.2.3 - RCWL Radar Sensor

Radar sensors are conversion devices that transform microwave echo signals into electrical signals. They use wireless sensing technology to detect motion by figuring out the object's position, shape, motion characteristics, and motion trajectory. One of the biggest advantages radar sensors have over other sensors is their detection of motion and velocity.

* PIR sensor

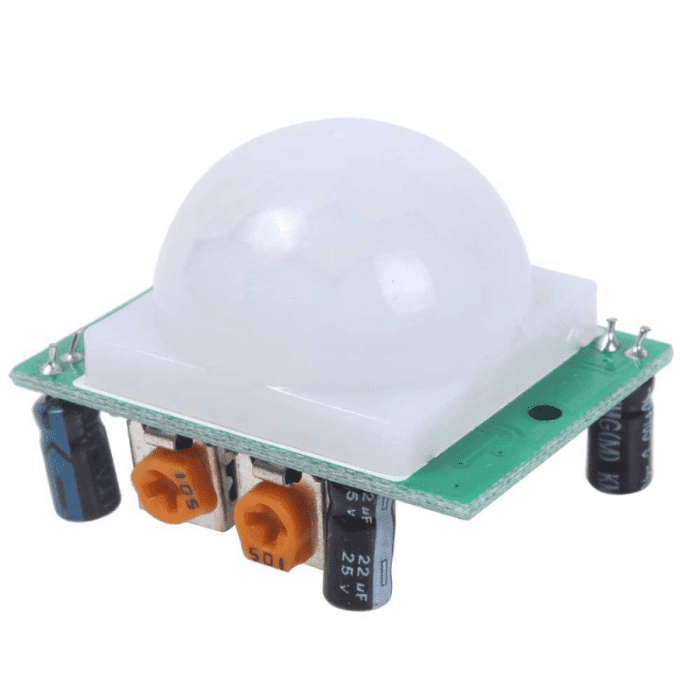


Fig 4.2.4 - PIR sensor

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensor range. They are small, inexpensive, low-power, easy to use, and don't wear out. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors. PIRs are made of a pyroelectric sensor that can detect levels of infrared radiation. Everything emits some low-level radiation, and the hotter something is, the more radiation is emitted.

* LDR Sensor



Fig4.2.5 - LDR Sensor

Photo resistors are light-sensitive resistors whose resistance decreases as the intensity of light they are exposed to increases. Photo resistors, also known as light-dependent resistors (LDR), are light-sensitive devices most often used to indicate the presence or absence of light or to measure light intensity. In the dark, their resistance is very high, sometimes up to 1 MΩ, but when the LDR sensor is exposed to light, the resistance drops dramatically, even down to a few ohms, depending on the light intensity. LDRs have a sensitivity that varies with the wavelength of the light applied and are nonlinear devices.

* **TTL module**



Fig 4.2.6 - TTL module

TTL stands for transistor-transistor logic. TTL module is a great little tool for embedded systems that require a serial connection to a computer. The board can simply attach to a USB bus and will appear as a standard COM port. The TTL converter module is a signal conditioning module which converts an analog signal into a TTL pulse. It is typically used to convert the signal from a blade passage into a TTL pulse. TTL or Transistor-Transistor Logic is a standard method of making logical circuits using transistors. Digital logic circuits are the main basic part of any electronic devices such as computer, laptops, smartphones, etc. A computer CPU has many complex logical circuits inside it. So many transistors are connected to each other in the TTL method to make the logical circuits.

* Buzzer



Fig4.2.7 - Buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical or electromechanical. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke. In this project the buzzer has been connected to the esp camera module. Based on the data received from the sensors, it will give an output of the object detected to the LCD and buzzer. The LCD will show an alert message and the buzzer will make an alarming sound for a few seconds which will confirm to the host that there is someone or a moving object in that area.

* Bread Board

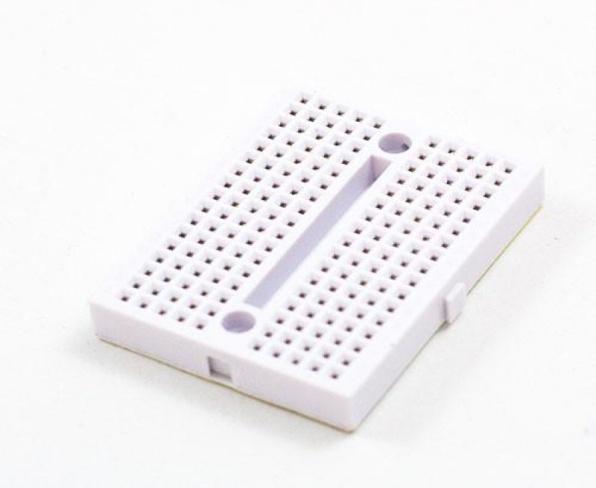


Fig4.2.8 - Bread Board

A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. Unlike a perboard or stripboard , breadboards do not require soldering or destruction of tracks and are hence reusable. For this reason, breadboards are also popular with students and in technological education.

* AC light bulb



Fig 4.2.9 - AC Bulb

The change in direction of the current flow happens so fast that the light bulb does not have a chance to stop glowing. The lightbulb does not care if it is using DC or AC current.

* Servo Motor



Fig 4.2.10 - Servo motor

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor.

* 1. **Software used**

The software used for programming the Arduino Uno in this project is Arduino IDE1. The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

* 1. **Proposed Methodology**

The system works mainly based on two approaches, wired communication and wireless communication. The wireless communication is carried out through wifi-internet connection.

The controller that we used is an ESP32 CAM. The ESP32-CAM is a small size, low power consumption camera module based on ESP32. It comes with an OV2640 camera and provides onboard TF card slot. The Wifi connectivity is also done through this board.

The components such as the PIR Sensor, RCWL Radar Sensor, LDR sensor, 2 Channel relays, Buzzer, AC light bulb, Servo Motor, all are connected as shown in the circuit diagram. Though the circuit shown is a simplified one, actually it is very conjested to do the wirings on a bread board. For this reason, we made a PCB (Printed Circuit Board) for reducing wiring mess and using less space. Now according to the PCB design, the components were connected and soldiered together.

The system does 3 types of detection- direct motion in front of the sensor, motion beyond a solid object, and light intensity.

Motion sensing is done by the PIR sensor and the RADAR sensor. Now if we place the PIR/Radar sensor on places where possible threats may pass by, it will detect and send a signal to the controller. The LDR sensor is given to add a automatic light activation during night hours. the AC light bulb is controlled through a relay where the LDR sends an input when the resistance value goes beyond a threshold value.

Smartphone connectivity is the key point of the system, which is achieved by the app called Telegram. Telegram has feature by which we can create AI bots.

How do Telegram bots work?

Most Telegram chat bots are based on a conversational flow.

And a conversational flow is created by three core elements:

Trigger: Tells the chat bot when to start a conversation

Action: Tells the chat bot what to do (for example: send a message)

Condition: Performs checks based on if this, then that logic

The process to create a BOT -

We go the telegram app, there we have to search for "Botfather" then start a chat. Then we have to click on Start at the bottom. So, now we need to say “/newbot” to create a chatbot.

After that we will get a new unique Chat ID and Token no, which will be specific for a user. The ID and Token no is then used in the code for communication through the device.

The microcontroller has Wifi connectivity, which through code we can connect it to internet by providing the Wifi ID and password.

Now the signal from the motion sensors give two outputs- the buzzer goes off and the camera module takes a picture at that instant. Then the photo is sent to the user which has the unique chat ID. We can store multiple chat IDs so that different users can control the device if required.

We can control all the sensors through the BOT, such as turning ON/OFF a particular sensor, Rotation of the servo, clicking pictures manually, turning ON/OFF all the sensors etc.

**1.5 Conclusion**

'Smart Home Security System' detects motion and sends an alarm through a buzzer. PIR motion sensors along with a Radar sensor have been used to detect an object beyond a solid surface, interfaced with the ESP32 module to sense the motion. The LDR sensor which detects the light intensity of the outdoor environment has been adjusted with the system and it turns on an AC light bulb through a relay. The camera fitted to the system clicks a picture of the detected objects and sending it to the user through telegram. The system was tested and we were able to get the desired results. The conclusion arrived from this project that this system will help in minimizing threats of burglary and theft and thus provide complete security for the house hold. This project was replaced with our Arduino Uno microcontroller as ESP32 is way too convenient to use. Also economically it is cheaper and all the sensors are interfaced properly.

* 1. **Reference**

1. GONG Shang-fu, YIN Xiao-quing. “ Solution of Home Security based on ARM and ZIGBEE”, International Symposium on Computer, Consumer and Control 2016.
2. Zhaoqing Peng, Takumi Kato, Hideyuki Takahashi, Tetsuo Kinoshita.

“Intelligent Home Security System Using Agent based IOT Devices”.

1. P. Satya Ravi Teja, A. Sai Srikar, V. Kushal, K. Srinivasan.

“Photosensitive Security System for Theft Detection and Control

using GSM technology”.

1. Suresh S, J Bhavya, S. Sakshi, K. Varun and G. Debarshi. “Home monitoring and Security System”
2. Rozita Teymourzadeh, Salah Addin Ahmed, Kokwaichen, Mok Vee Hoong. “Smart GSM based home automation”. IEEE conference, system and control
3. Vaibhav Sharma, Chirag Fatnani, Pranjalkatara, Vishnu Shankar. “Advanced Low Cost Security System using Sensors, Arduino and GSM Communication Module”.
   1. **Appendix**

**Code used till now**

#include <ArduinoJson.h>

#include <EEPROM.h>

#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>

//"Suman-JioFiber-4G";

//static const char\* WIFI\_PASS = "SDsuman06#";

#include <ESP32Servo.h>

const char\* ssid = "WIFI\_221";//"Suman-JioFiber-4G";//"realme 6"; //WIFI SSID WIFI\_221//

const char\* password = "221\_hostel61";//"SDsuman06#";//"donotaskme"; //WIFI password221\_hostel6"221\_hostel6";//

//String token = "5457437715:AAGjlo3VKmrLI8Guv9D4roZfDAZ1uYnbyiM";//

//String token= "5868109345:AAFJMlKpi7S4lDWhBxHC2FqcEF\_FUWedvWA";//

//String token = "6117248989:AAF-5vlZU1YUjWAX0CqpTrlNINwsUxtmwog";//

String token = "5457437715:AAFMyoy0ubgH3RsgyrND6sKNODinpXoCniI";

String chat\_id = "1058093244"; //;//

//String chat\_id="1635819206";

//String chat\_id="1116751308";

String Cht\_ID = "";

//#define Cht\_ID "1635819206"

String Cht\_ID2 ="";

String Cht\_ID3 ="";

/\*\*/

WiFiClientSecure clientTCP;

UniversalTelegramBot bot(token, clientTCP);

Servo myservo;

#define CAMERA\_MODEL\_AI\_THINKER // Has PSRAM

//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

//pins

#define PIR 13

#define LDR 15

#define RADAR 14

#define REL1 3

#define REL2 1

#define ServoPin 12

#define buzzer 2

#define Read\_PIR digitalRead(PIR)

#define Read\_LDR digitalRead(LDR)

#define Read\_RADAR digitalRead(RADAR)

#define REL1\_ON digitalWrite(REL1,1)

#define REL1\_OFF digitalWrite(REL1,0)

#define REL2\_ON digitalWrite(REL2,1)

#define REL2\_OFF digitalWrite(REL2,0)

int pir1 = 0,L1=0,L2=0,RDR=0, count=0,count1=0, count2=0;

int botRequestDelay = 1000;

unsigned long lastTimeBotRan;

#define Waiting\_period 3000 //milisec

#define Update\_period 2000 //milisec

#define Ldr\_threshold 700

#define Ldr\_TurnoffRange 600

bool Relay1\_status=false,Relay2\_status=false,temp1=false,temp2=false,temp3=false,servos=false;

uint32\_t privious\_time;

uint32\_t privious\_time1;

#define FLASH\_LED\_PIN 4

bool flashState = LOW,sendPhoto = false, motionDetected = false;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void myservo (int Pin\_No, int angle){

int Time\_pos= map(angle, 0,180,550,2450);int minval=2450-Time\_pos;

digitalWrite(Pin\_No,1); delayMicroseconds(Time\_pos);

digitalWrite(Pin\_No,0); delayMicroseconds(minval);

}\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EEPROMWRITE(int addr, const String &data)

{ byte Length = data.length(); EEPROM.write(addr, Length);EEPROM.commit();

for (int i = 0; i < Length; i++){EEPROM.write(addr + 1 + i, data[i]);EEPROM.commit(); }

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

String EEPROMREAD(int addr)

{ int Length = EEPROM.read(addr); char data[Length + 1];

for (int i = 0; i < Length; i++) { data[i] = EEPROM.read(addr + 1 + i); }

data[Length] = '\0'; return String(data);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EEPROM\_READ( ){pir1 = EEPROM.read(10);RDR = EEPROM.read(11);L1 = EEPROM.read(12);L2 = EEPROM.read(13);}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EEPROM\_WRITE(int addr, int val ){EEPROM.write(addr,val);EEPROM.commit();}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void handleNewMessages(int numNewMessages){//Serial.println(" 1New message");

for (int i = 0; i < numNewMessages; i++) {// Serial.println(" New message");

String chat\_id = bot.messages[i].chat\_id; // Serial.print(" chat id "); Serial.println(chat\_id);// Cht\_ID=chat\_id;

String text = bot.messages[i].text; // Serial.print(" message is"); Serial.println(text);

EEPROM\_READ( );

String from\_name = bot.messages[i].from\_name;

if (from\_name == "") from\_name = "Guest";

if (text == "/SENSOROFF"){pir1 = 0;RDR=0 ; EEPROM\_WRITE(10,pir1);EEPROM\_WRITE(11,RDR); bot.sendMessage(chat\_id, "ALL SENSORS ARE OFF", ""); }

if (text == "/SENSORON"){pir1 = 1;RDR=1 ;EEPROM\_WRITE(10,pir1); EEPROM\_WRITE(11,RDR);bot.sendMessage(chat\_id, "ALL SENSORS ARE ON", ""); }

if (text == "/PIR1OFF") { pir1=0; EEPROM\_WRITE(10,pir1); bot.sendMessage(chat\_id, "PIR SENSOR IS OFF", ""); }

if (text == "/PIR1ON") { pir1=1; EEPROM\_WRITE(10,pir1); bot.sendMessage(chat\_id, "PIR SENSOR IS ON", ""); }

if (text == "/RadarOFF") { RDR=0; EEPROM\_WRITE(11,RDR); bot.sendMessage(chat\_id, "Radar SENSOR IS OFF", ""); }

if (text == "/RadarON") { RDR=1; EEPROM\_WRITE(11,RDR); bot.sendMessage(chat\_id, "Radar SENSOR IS ON", ""); }

if (text == "/status") {EEPROM\_READ();

String text1="";

if (pir1==1){ text1="PIR SENSOR IS ON\n" ; } if (pir1==0){text1="PIR SENSOR IS OFF\n" ; }

if (RDR==1){ text1+="Radar SENSOR IS ON\n"; }if (RDR==0){text1+="Radar SENSOR IS OFF\n"; }

if (L2==1){ text1+="RELAY2 IS ON\n"; } if (L2==0) { text1+="RELAY2 IS OFF\n"; }

if (L1==1){ text1+="RELAY1 IS ON\n"; }if (L1==0){ text1+="RELAY1 IS OFF\n"; }

bot.sendMessage(chat\_id, text1, "");

}

if (text=="/SERVOOFF"){ servos=false; myservo.write(10);;bot.sendMessage(chat\_id, "SERVO IS OFF", "");}

if (text=="/SERVOON"){ servos=true;bot.sendMessage(chat\_id, "SERVO IS ON", "");}

if (text == "/ADDCHATID"){Cht\_ID=chat\_id;EEPROMWRITE(50, chat\_id);bot.sendMessage(chat\_id, "Chat ID "+chat\_id+" is added", "");}

if (text == "/RELAY2ON"){if (!Relay2\_status){REL2\_ON;Relay2\_status=true;bot.sendMessage(chat\_id, "Relay2 is on", "");EEPROM\_WRITE(13,1);}}

if (text == "/RELAY2OFF"){REL2\_OFF;Relay2\_status=false;bot.sendMessage(chat\_id, "Relay2 is off", "");EEPROM\_WRITE(13,0);}

if (text == "/photo"){bot.sendMessage(chat\_id, "Sending photo", "");sendPhotoTelegram();delay(1000);}

if (text == "/start")

{ String welcome = "TELEGRAM USER..., " + from\_name + ".\n";

welcome += "/SENSORON : to turn on all the sensors \n";

welcome += "/SENSOROFF : to turn off all the sensors\n";

welcome += "/status : to check the status of the sensors\n";

welcome += "/start : to get welcome SMS\n";

welcome += "/PIR1OFF : to turn off PIR Sensor\n";

welcome += "/PIR1ON : to turn on PIR Sensor\n";

welcome += "/RadarON : to turn on Radar Sensor\n";

welcome += "/RadarOFF : to turn off Radar Sensor\n";

welcome += "/RELAY2ON : to turn on relay2\n";

welcome += "/RELAY2OFF : to turnoff relay2\n";

welcome += "/ADDCHATID : to add new chat id\n";

welcome += "/photo : to get photo\n";

welcome += "/SERVOON : to Start the servo\n";

welcome += "/SERVOOFF : to Stop the servo\n";

bot.sendMessage(chat\_id, welcome, "Markdown");

}

}}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

String sendPhotoTelegram(){

const char\* myDomain = "api.telegram.org";

String getAll="", getBody = "";

camera\_fb\_t \* fb = NULL;

fb = esp\_camera\_fb\_get();

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

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

String head = "--India\r\nContent-Disposition: form-data; name=\"chat\_id\"; \r\n\r\n" + Cht\_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;

clientTCP.println("POST /bot"+token+"/sendPhoto HTTP/1.1");

clientTCP.println("Host: " + String(myDomain));

clientTCP.println("Content-Length: " + String(totalLen));

clientTCP.println("Content-Type: multipart/form-data; boundary=India");

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; // timeout 10 seconds

long startTime = millis(); boolean state = false;

while ((startTime + waitTime) > millis())

{ Serial.print("."); delay(100);

while (clientTCP.available()) { char c = clientTCP.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; }

clientTCP.stop();// Serial.println(getBody);

}

else { getBody = "Connection to telegram failed.";

//Serial.println("Connection to telegram failed.");

}

return getBody;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

static void IRAM\_ATTR detectsMovement(void \* arg){

if (pir1==1){ sendPhoto=true;}}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void camera\_config(){

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); // UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA

err = gpio\_isr\_handler\_add(GPIO\_NUM\_13, &detectsMovement, (void \*) 13);

if (err != ESP\_OK){ }

err = gpio\_set\_intr\_type(GPIO\_NUM\_13, GPIO\_INTR\_POSEDGE);

if (err != ESP\_OK){ }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// if PSRAM IC present, init with UXGA resolution and higher JPEG quality

// for larger pre-allocated frame buffer.

if(config.pixel\_format == PIXFORMAT\_JPEG){

if(psramFound()){

config.jpeg\_quality = 10;

config.fb\_count = 2;

config.grab\_mode = CAMERA\_GRAB\_LATEST;

} else {

// Limit the frame size when PSRAM is not available

config.frame\_size = FRAMESIZE\_SVGA;

config.fb\_location = CAMERA\_FB\_IN\_DRAM;

}

} else {

// Best option for face detection/recognition

config.frame\_size = FRAMESIZE\_240X240;

#if CONFIG\_IDF\_TARGET\_ESP32S3

config.fb\_count = 2;

#endif

}

#if defined(CAMERA\_MODEL\_ESP\_EYE)

pinMode(13, INPUT\_PULLUP);

pinMode(14, INPUT\_PULLUP);

#endif

// drop down frame size for higher initial frame rate

if(config.pixel\_format == PIXFORMAT\_JPEG){

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

#if defined(CAMERA\_MODEL\_ESP32S3\_EYE)

s->set\_vflip(s, 1);

#endif

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void setup() { //Serial.begin(115200);

WRITE\_PERI\_REG(RTC\_CNTL\_BROWN\_OUT\_REG, 0);

EEPROM.begin(200);

pinMode(PIR, INPUT);

pinMode(RADAR, INPUT);

pinMode(LDR, INPUT);

pinMode(REL1, OUTPUT);

pinMode(REL2, OUTPUT);

pinMode(buzzer, OUTPUT);

pinMode(FLASH\_LED\_PIN, OUTPUT);digitalWrite(FLASH\_LED\_PIN ,0);

digitalWrite(RADAR, 0); REL1\_OFF;REL2\_OFF ; delay(10);

Cht\_ID=EEPROMREAD(50);

EEPROM\_READ( );

camera\_config();

myservo.attach(ServoPin);//pinMode(ServoPin, OUTPUT);

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)

{digitalWrite(FLASH\_LED\_PIN ,1); delay(500);digitalWrite(FLASH\_LED\_PIN ,0); if ((StartTime+10000) < millis()) break; }

digitalWrite(FLASH\_LED\_PIN ,0);

clientTCP.setCACert(TELEGRAM\_CERTIFICATE\_ROOT);

if (WiFi.status() != WL\_CONNECTED) { } //ESP.restart(); Serial.println("Reset");

bot.sendMessage(chat\_id, "hii.... device started", "");

}

void loop() {

if (millis() > lastTimeBotRan + botRequestDelay) {

int numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);//Serial.println(numNewMessages);

while (numNewMessages) { handleNewMessages(numNewMessages); //Serial.println("got response");

numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

} lastTimeBotRan = millis();

}

if (sendPhoto) { bot.sendMessage(chat\_id, "THIEF DETECTED ON PIR", ""); digitalWrite(FLASH\_LED\_PIN ,1);

sendPhotoTelegram();digitalWrite(FLASH\_LED\_PIN ,0); sendPhoto=false; delay(1000); }

if ((Read\_LDR==1)&&(Read\_PIR==1)&&(Relay1\_status==false)){ REL1\_ON; EEPROM\_WRITE(12,1);Relay1\_status=true;}

else if((Read\_LDR==1)&&(Relay1\_status==true)) { REL1\_OFF;Relay1\_status=false;EEPROM\_WRITE(12,0);

}

if (pir1==1){

if (Read\_PIR==1){temp1=true;count1++; if(count1==2){tone(buzzer, 200,1500);

bot.sendMessage(Cht\_ID,"Object DETECTED on PIR", "");sendPhoto=true; noTone(buzzer); } }

else{count1=0;temp1=false;}}else{}//Serial.println("loopd1 ");

if (RDR==1){

if ((Read\_RADAR)&&(millis() - privious\_time > Waiting\_period))

{ temp2=true;count2++;count=0;if(count2==1){ bot.sendMessage(Cht\_ID,"Object DETECTED on RADAR", "");count2++;}

privious\_time=millis();

}else{ temp2=false;count1=0;count2=0;count++;if(count==1){ count++;}}

}

if (servos==true){//int num=text.toInt();

for (int a=10;a<100;a++){ myservo.write(a); delay(80); }

for (int a=100;a>10;a--){ myservo.write(a); delay(80); }

}

if (millis() - privious\_time1 > Update\_period){

privious\_time1=millis();//Serial\_Update();

}

}