SMART SECURITY CAM

A PROJECT REPORT

Submitted by

PRIYANKA T (732416104040)

SANDHIYA K (732416104043)

VAISHNAVI S (732416104051)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SASURIE COLLEGE OF ENGINEERING, TIRUPUR

ANNA UNIVERSITY::CHENNAI 600 025

MARCH 2020

ANNA UNIVERSITY : CHENNAI:600 025

BONAFIDE CERTIFICATE

Certified that this project report **"SMART SECURITY CAM"** is the bonafide work of **"PRIYANKA T (732416104040), SANDHIYA K (732416104043), VAISHNAVI S (732416104051"),** Who have carried out this project under my supervision.

SIGNATURE	**SIGNATURE**
Mr. SRIKANTH J **B. Tech, M.E.,(Ph. D).,** **HEAD OF DEPARTMENT**	**Mr. SRIKANTH J** **B. Tech, M.E.,(Ph. D).,** **SUPERVISOR**
ASSOCIATE PROFESSOR	**ASSOCIATE PROFESSOR**
Department Of Computer science and Engineering,	Department Of Computer science and Engineering,
Sasurie College of Engineering,	Sasurie College of Engineering,
Tirupur-638056.	Tirupur-638056.

Submitted for the project viva voice held on .....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude to our beloved Chairman **Shri A.M. KANDASWAMI B.Sc**., Sasurie Institutions, for providing the facilities during the course of our study in the college.

We wish to express our sense of gratitude to our beloved **CEO Dr.VIVEKANANDHAN ,** Sasurie Institution, for providing valuable support and encouragement towards us.

We express our sincere gratitude to our beloved Principal **Dr.T.S.SIVA KUMAR , M.E., Ph.D.,** Sasurie College of Engineering who gave the opportunity to frame the project to the full satisfaction.

The gratitude passes on to **Mr.SRIKANTH J.,B.Tech.,M.E.,(Ph.D).,** Head of the Department of Computer science and Engineering for his valuable help and suggestion impaired to us.

In this regard especially we wish to express our gratitude and sincere thanks to our supervisor **Mr.SRIKANTH J.,B.Tech.,M.E.,(Ph.D).,** for extending his inspiration and advice in completing this project successfully.

The acknowledgement would be incomplete without a word of thanks to all our **STAFF MEMBERS, PARENTS** and **FRIENDS** for their continued and sincere help throughout the project.

ABSTRACT

A variety of motion detection algorithms for surveillance systems are developed. But most of the systems do not absolutely detect the moving object because it causes some darkness and it requires large memory to store the video. We are developing motion detection system that will be helpful for detecting the moving object without present of shadow. By using Human Motion Detection system banks safe will be more secured as it will send alerts regarding burglary happening. Moreover it will save memory and memory wastage would be avoided.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DFD - Data Flow Diagram

ER - Entity Relationship

ESP - Extra Sensory Perceptions

USB - Universal Serial Bus

TTL - Transistor Transistor Logic

USART - Universal Serial Asynchornous Receiver Transmitter

CAM - Camera

SMTP - Simple Mail Transfer Protocol

PIR - Passive Infrared Sensor

SMS - Short Message Service

CCTV - Closed Circuit Television

IDE - Integrated Development Environment

SD - Secured Digital

CHAPTER 1

INTRODUCTION

1.1. Introduction to IOT :

IOT is a network in which all physical objects are connected to the Internet through network devices or routes and exchange data. IOT allows objects to be controlled remotely across existing network infrastructure . IOT is a very good and intelligent technique which reduces human effort as well as easy access to physical devices . This technique also has autonomous control feature by which any device can control without any human interaction. “Things” in the IOT sense , is the mixture of hardware , software, data , and services .“Things” can refer to a wide variety of devices such as DNA analysis devices for environmental monitoring , electric clamps in coastal waters , arduino chips in home automation and many other .

These devices gather useful data with the help of various existing technologies and share that data between other devices . Examples include home automation system which uses WiFi or bluetooth for exchange data between various devices of home.

The thought of Internet of Things first became popular in 1999.British entrepreneur Kevin Ashton first used the term Internet of Things in 1999 while working at Auto-ID labs . Besides that near field communication , barcode scanners , QR code scanners and digital watermarking are the various devices which are working on IoT in the present scenario.

Four main components of IOT:

1. Low-power embedded systems:

Less battery consumption, high performance are the inverse factors play a significant role during the design of electronic systems.

2. Cloud computing :

Data collected through IoT devices is massive and this data has to be stored on reliable storage server. This is where cloud computing comes into play. The data is processed and learned, giving

more room for us to discover where things like electrical faults/errors are within the system.

3. Availability of big data :

We know that IoT relies heavily on sensors, especially real-time. As these electronic devices spread throughout every field, their usage is going to trigger a massive flux of big data.

4. Networking connection :

In order to communicate, internet connectivity is a must where each physical object is represented by an IP address. However, there are only a limited number of addresses available according to the IP naming. Due to the growing number of devices, this naming system will not be feasible anymore. Therefore, researchers are looking for another alternative naming system to represent each physical object.

Three layers of IOT architecture:

(i)The **perception layer** is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.

(ii)The **network layer** is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.

(iii)The **application layer** is responsible for delivering application specific services to the user. It defines various applications in which the Internet of Things can be deployed, for example, smart homes, smart cities, and smart health.

Evolution of IOT :

The rapid evolution of IOT is not just about maneuvering devices through mobile. The evolving IOT landscape also gave birth to revolutionary concepts like ‘Smart City’, ‘Connected Vehicle,’ and ‘Industrial IOT’. From public places to workplace environments to transport vehicles, IOT and connected gadgets are continuing to open new opportunities for automation and smart communication with machine. For any IOT app development company, these new trends hold the key opportunities to outperform the competition and grow. With this tremendous pace of evolution through connected gadgets, sensors and environments, some recent trends are worth our explanation now. Here we are going to explain some of the key trends that are going to shape the digital landscape in the time to come.

Sensors :

A **sensor** is a device, module, machine, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently computer process. A sensor is always used with other electronics. It is a device that converts signals from one energy domain to electrical domain. The definition of the Sensor can be understood if we take an example in to consideration.

1.2. Objectives :

Objective of the project:

The Objective of the proposed system is to make safety for user money, documents, etc. It provides security for user belongings and it helps to improve security by sending notifications via mail and SMS. By using this user can find the theft easier.

CHAPTER 2
SYSTEM ANALYSIS

2.1 Existing System :

2.1.1.Literature Review :

Nowadays CCTVs are installed at many places like banks safe. But the CCTV cameras continuously record the situations. Hence there is an unnecessary memory wastage if there is nothing happening in front of the camera. Also the CCTV system does not provide alerts of burglary happening at particular time. So there is a need of a system which will record the situation only if there is some movement happening in front of the camera and send alerts to the manager as well as the police.

2.1.2.Drawbacks :

CCTV camera has a limited vision of a particular area, even though it can move up and down, it can't not cover the whole area. But you can add a number of cameras to make it up. 8Ch, 16CH, 32CH and more.

1. **It can be a costly affair :**

While dummy cameras may not be expensive, the real ones costs hundreds, even thousands of dollars depending on the features and the number of cameras and monitoring systems you buy. Getting them installed and their maintenance means added costs. If you're thinking of installing them yourself, lay that idea to rest unless you have good knowledge of wiring systems or you may end up damaging the cameras.

2. **Can't stop theft :**

Cameras enable users to record footage for later viewing, and to help nab criminals, and receive justice from the law. They cannot, however, stop a crime when it is in progress. They do not alert neighbors or the police like an alarm system would.

This means that you will incur losses even as you run to the court, make insurance claims and reorder

stolen inventory, which may no longer make you feel absolutely safe and even cause you to lose faith in them.

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 Software Requirements:

Software is a set of instructions that are used to command any system to perform any operation. Software has the advantage to make decisions and to deliver sensible results and is useful in handling complex situations.

There are two types of computer software as follows ,

1.System Software :

System software includes the programs that are dedicated to managing the computer itself, such as the operating system, file management utilities, and disk operating system.

2. Application Software :

Application software is specific to the task it is designed for and can be as simple as a calculator application or as complex as a word processing application.

Operating System : Windows 10 and any windows version

IDE : Arduino IDE version **1.8.10**

Language : Arduino specific C or Embedded C , PHP

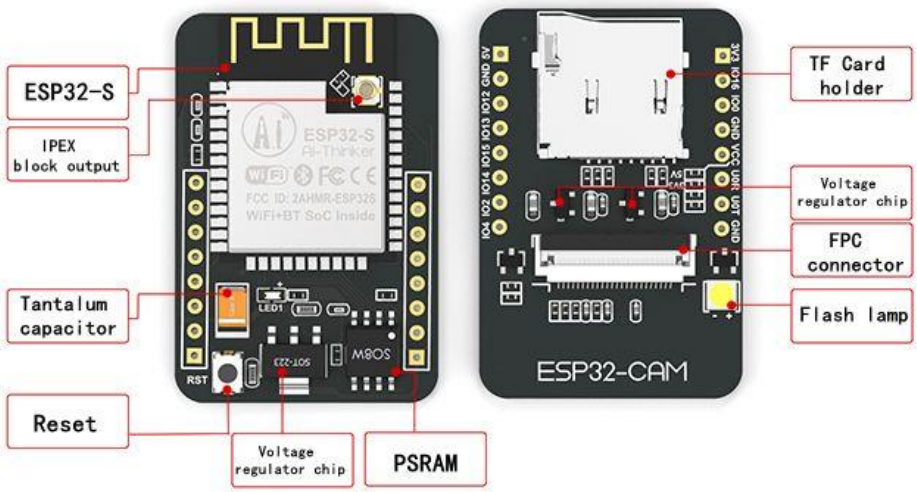
Protocol : SMTP

3.2. Hardware Requirements:

Hardware is the physical components of the computer like microprocessor, hard disks, RAM, and motherboard. Hardware devices are the executors of the commands provided by software applications.

1. ESP32 CAM :

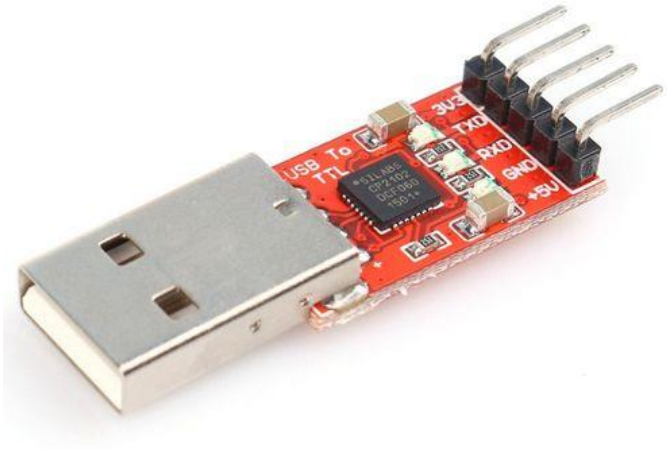
The ESP 32-CAM is a very small camera module with ESP32- S chip. Besides the OV2640 camera, and several GPIO to connect peripherals, it also features a microSD card slot that can be useful to store files to serve to clients.



3.2.1. ESP32 Camera

2. CH340 Board:

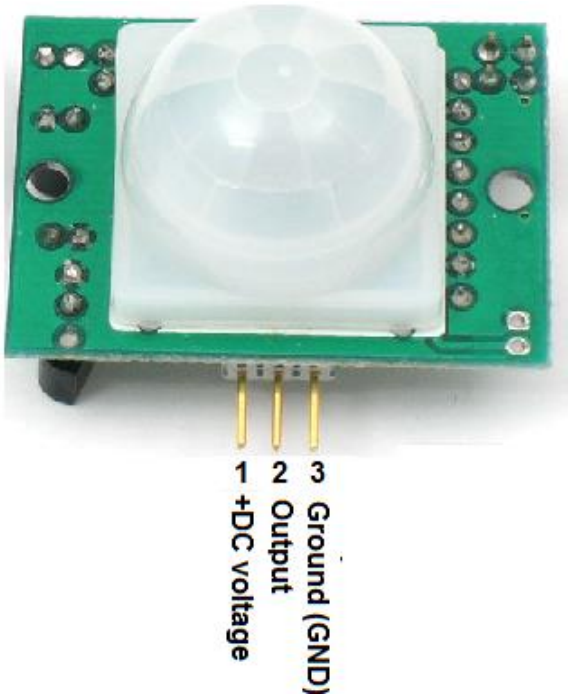
CH340 is a USB bus conversion chip, it can realize USB to UART interface or USB to printer interface of computer or upgrade the common serial device to USB bus directly. It is full speed USB device interface, compatible with USB. It supports 5V and 3.3V power supply even 3V.



3.2.2. CH340 Board

3. PIR Motion Detection Sensor:

Passive Infrared Sensor is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR based motion detectors.PIR sensors are commonly used in security alarms and automatic lighting applications. It is also called as Passive Infrared Detector. A PIR sensor can detect changes in the amount of infrared radiation impinging upon it, which varies depending the temperature and surface.



3.2.3. Passive Infrared Sensor

4. Jumper wires :

A **jump wire** (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Jumper wires typically come in three versions:

3.2.4. Male-To-Male

3.2.5. Female-To-Female

3.2.6. Male-To-Female or Female-To-Male

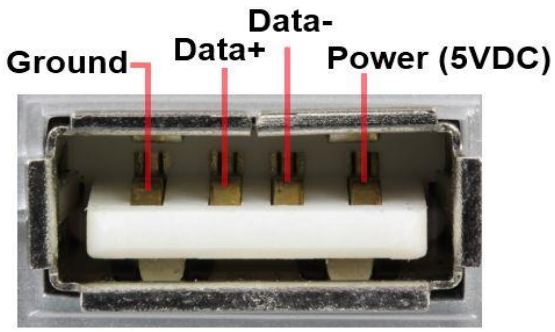
The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you'll need.

5. SD Card :

Secure Digital, officially abbreviated as **SD**, is a proprietary memory card format developed by the SD Card Association (SDA) for use in portable devices.
The standard was introduced in August 1999 by joint efforts between SanDisk, Panasonic(Matsushita Electric) and Toshiba as an improvement over MultiMediaCard (MMC), and has become the industry standard. The three companies formed SD-3C, LLC, a company that licenses and enforces intellectual property rights associated with SD memory cards and SD host and ancillary products.

6. 5v Power Supply :

In order to provide direct power current to the smart security camera 5v power supply is useful.



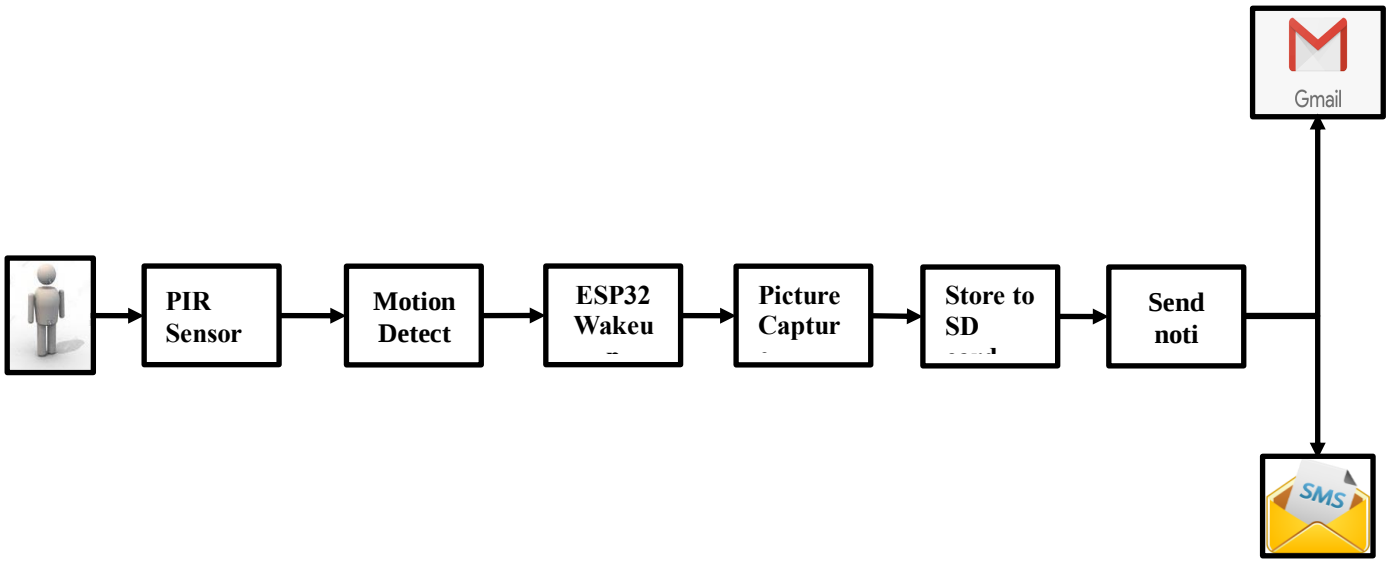
3.2.7. 5V Power Supply

CHAPTER 4
SYSTEM DESIGN

4.1. Proposed System:

This smart security camera captures image when human being, animal or any other living system try to enter restricted area. After capturing the image this system send the image via email to the authorized user. If that authorized user not enable the network for that case this system send SMS notification as well. PIR sensor detects motion by the heat energy emitted by human being, animals, or any other living system. PIR sensor passes alert signal to ESP32 Cam in order to wakeup.ESP32 Cam capture image and automatically connect to WiFi. Then ESP32 Cam send captured image via email. Then send normal text SMS to the authorized persons.

ARCHITECTURE DIAGRAM OF PROPOSED SYSTEM :



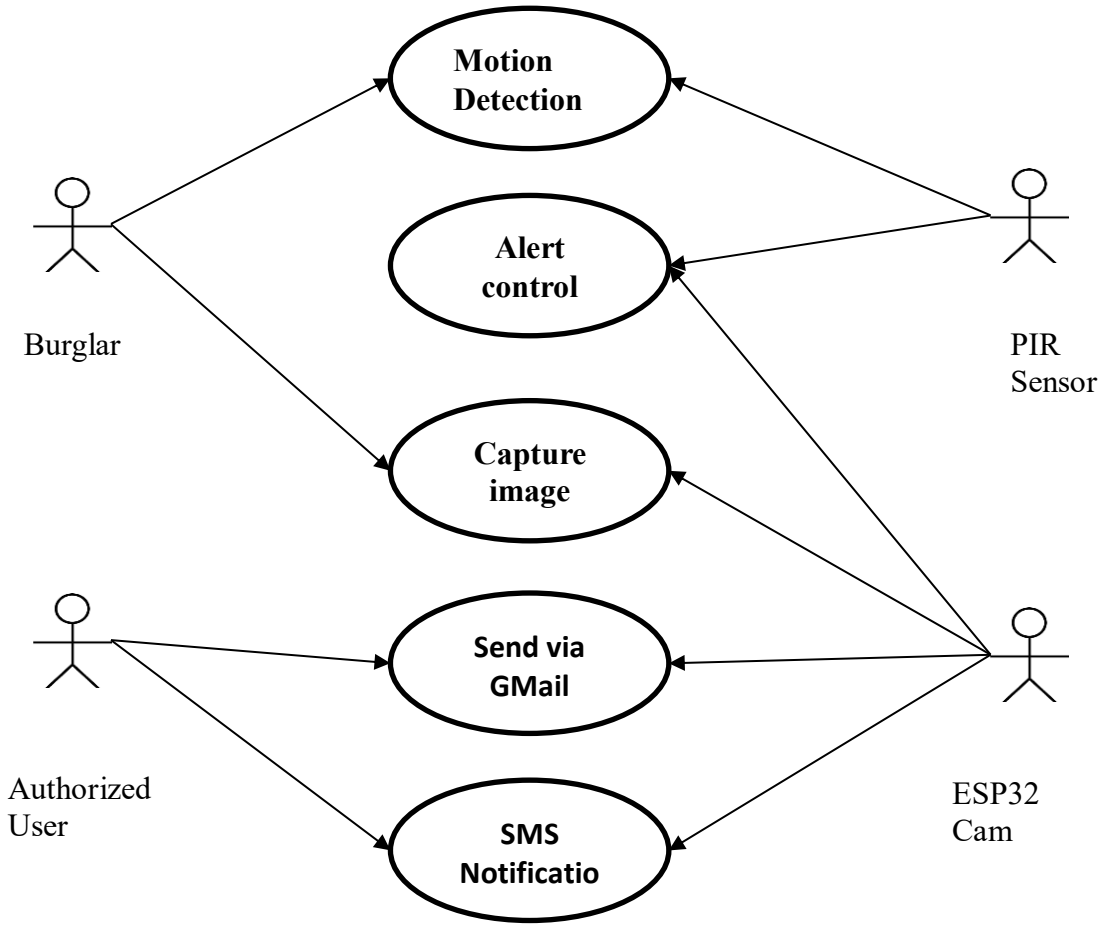
4.1.1. Architecture Diagram

4.2. Overview Of The Project:

1. When person enter into the PIR placed room.
2. PIR sense passive infrared radiation of human and send that signal to corresponding circuit.
3. ESP32 Cam wake up from sleep mode when GPIO pin13 goes low.
4. Take picture.
5. Save to sd card.
6. Login to sender mail id.
7. Attach taken image and compose mail.
8. Send to recipient.
9. Send SMS.
10. ESP32 go to sleep mode.

4.3. Use Case Diagram:

Use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into actors and use cases. Actors represent roles that can are played by users of the system. Those users can be humans, other computers, pieces of hardware, or even other software systems. The only criterion is that they must be external to the part of the system being partitioned into use cases. They must supply stimuli to that part of the system, and the must receive outputs from it.

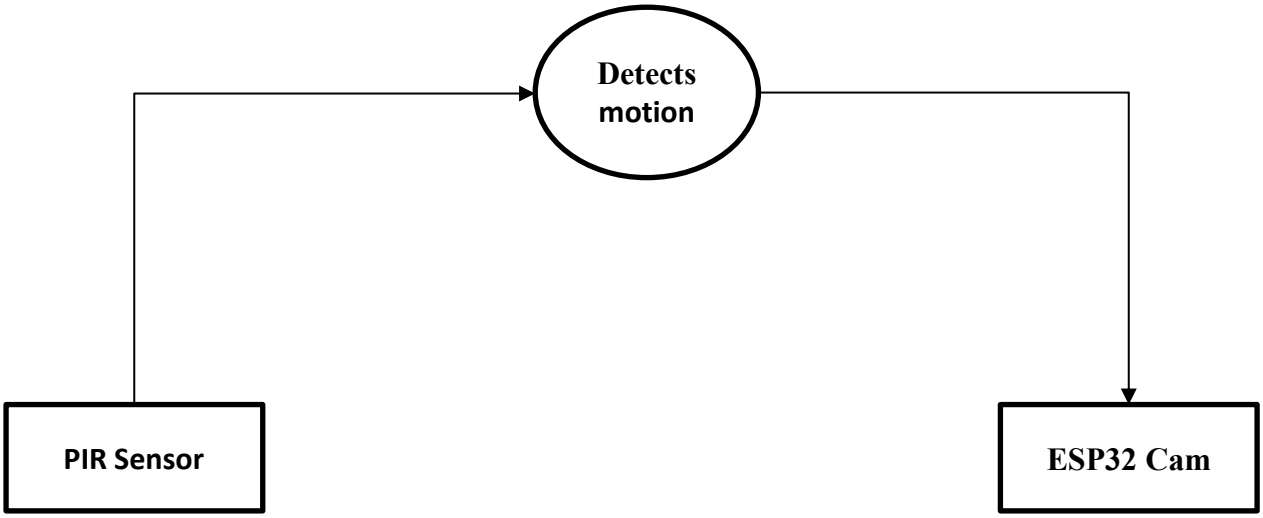


4.3.1. Use Case Diagram

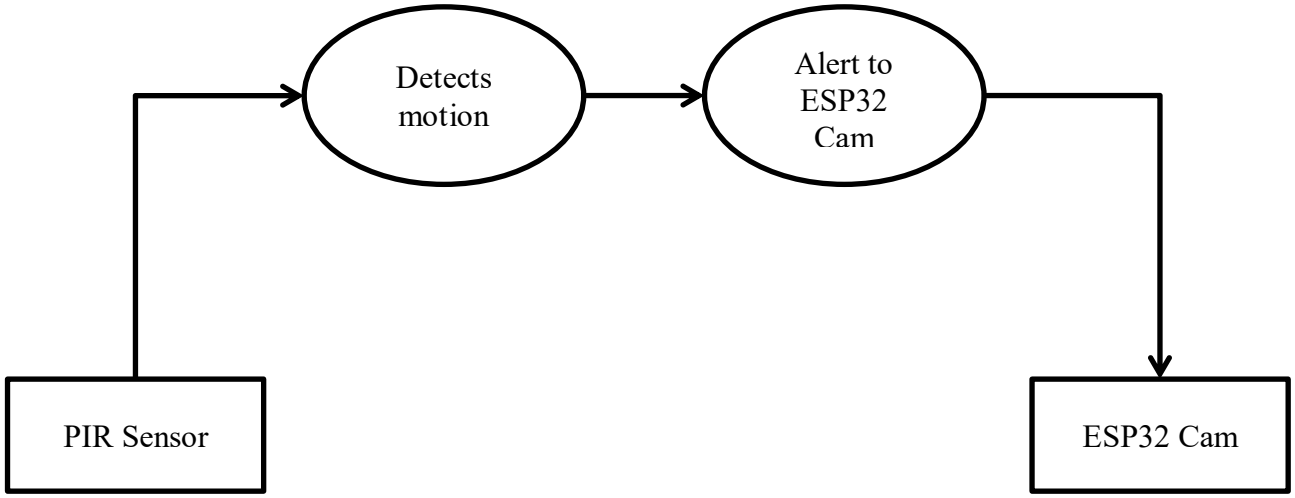
4.4. Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel.

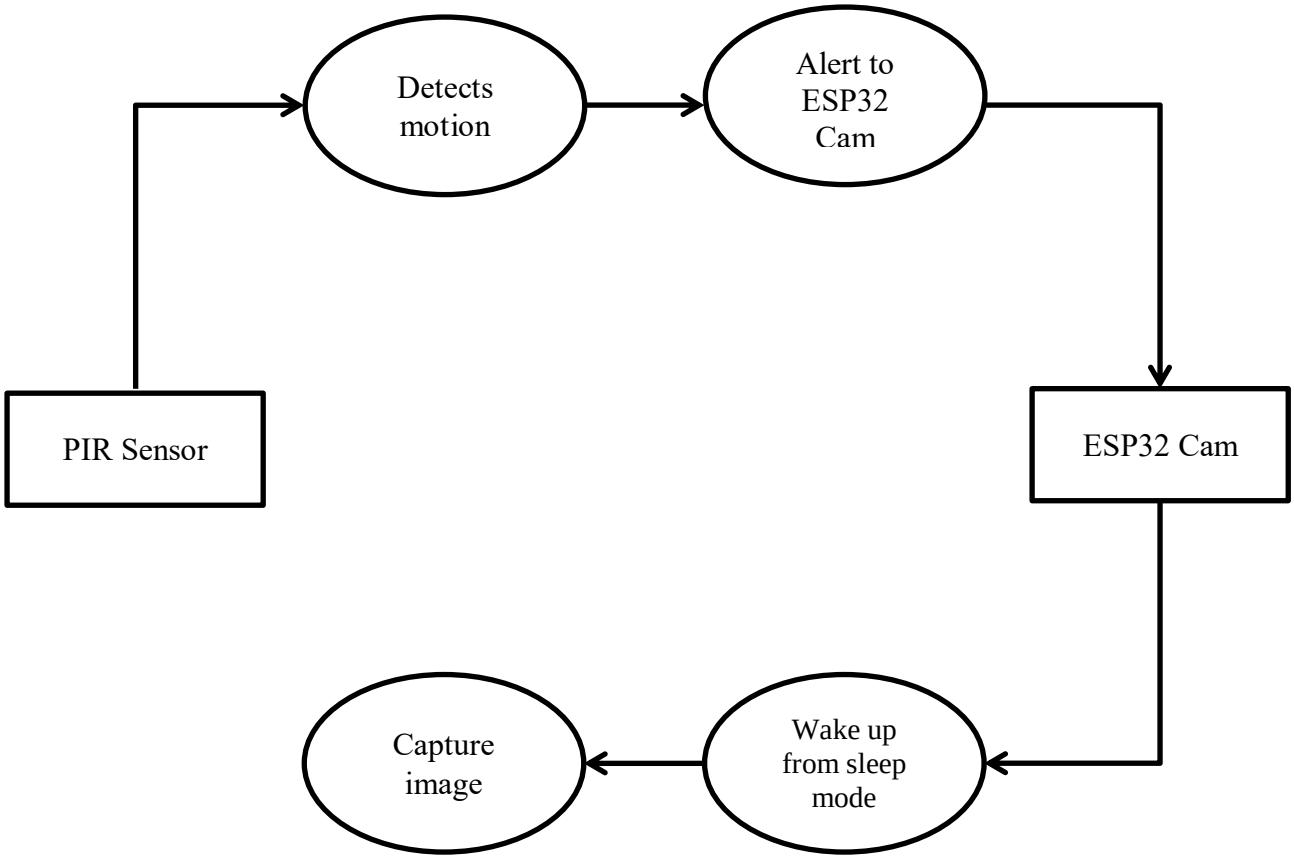
4.4.1. Level 0 DFD :



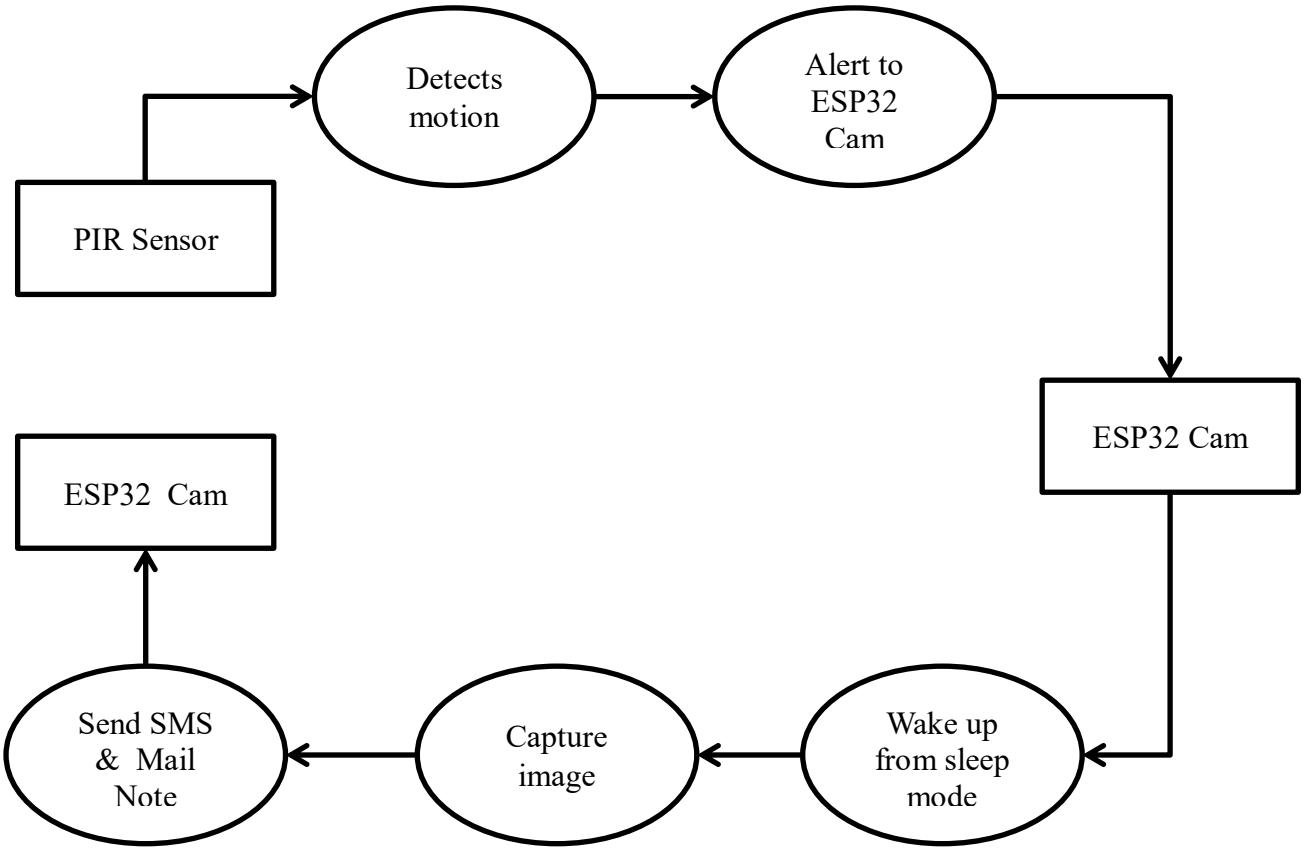
4.4.2. Level 1 DFD:



4.4.3. Level 2 DFD :

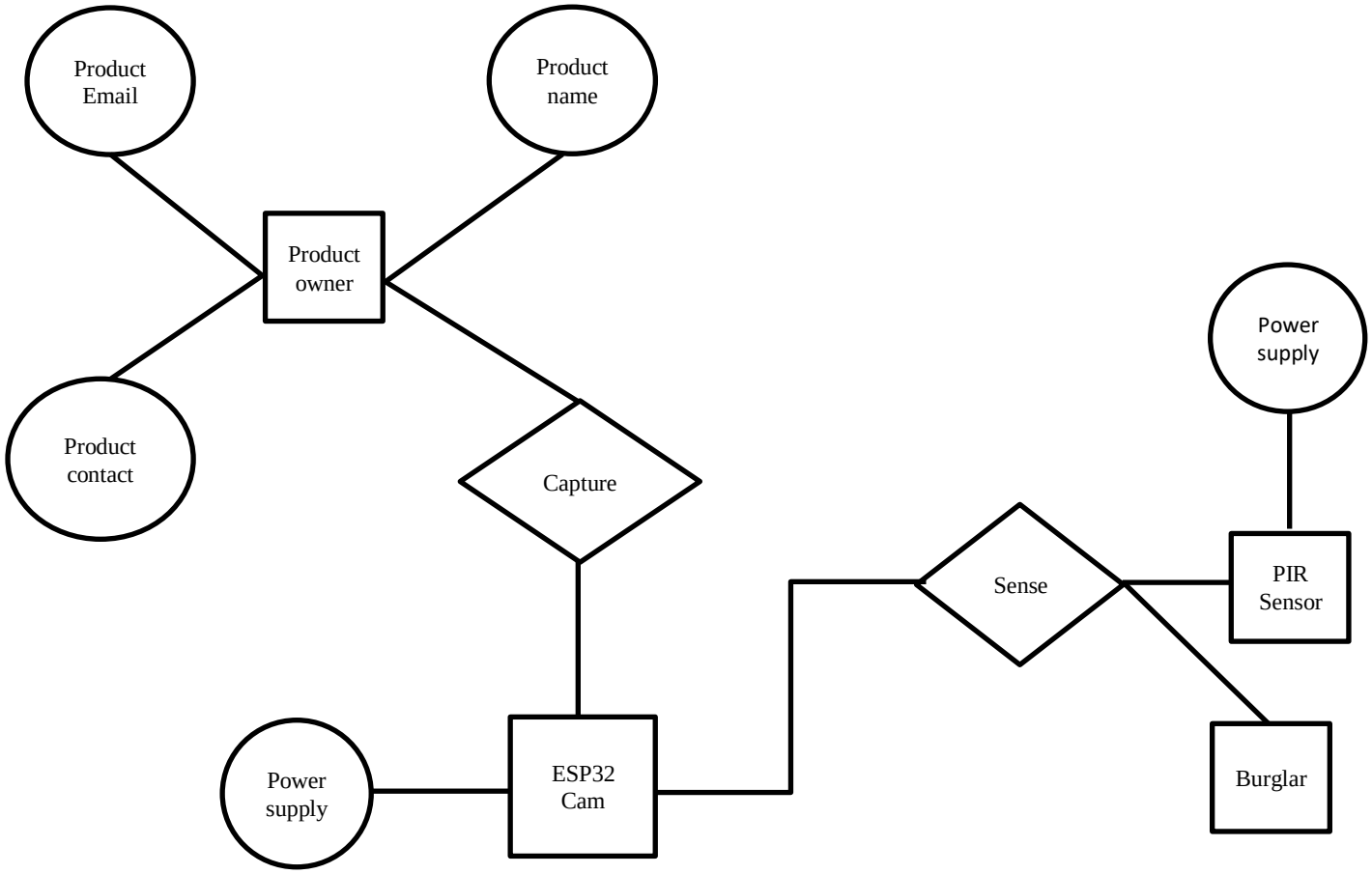


4.4.4. Level 3 DFD :



4.5. ER DIAGRAM :

An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within database or information systems. An entity is a piece of data-an object or concept about which data is stored.



4.5.1. ER Diagram

CHAPTER 5
SYSTEM IMPLEMENTATION

5.1. SOFTWARE SETUP

Software setup is the first step in the project system implementation . In Arduino IDE we have to setup some library which is provided by third party.

Link to download libraries for ESP32 CAM:

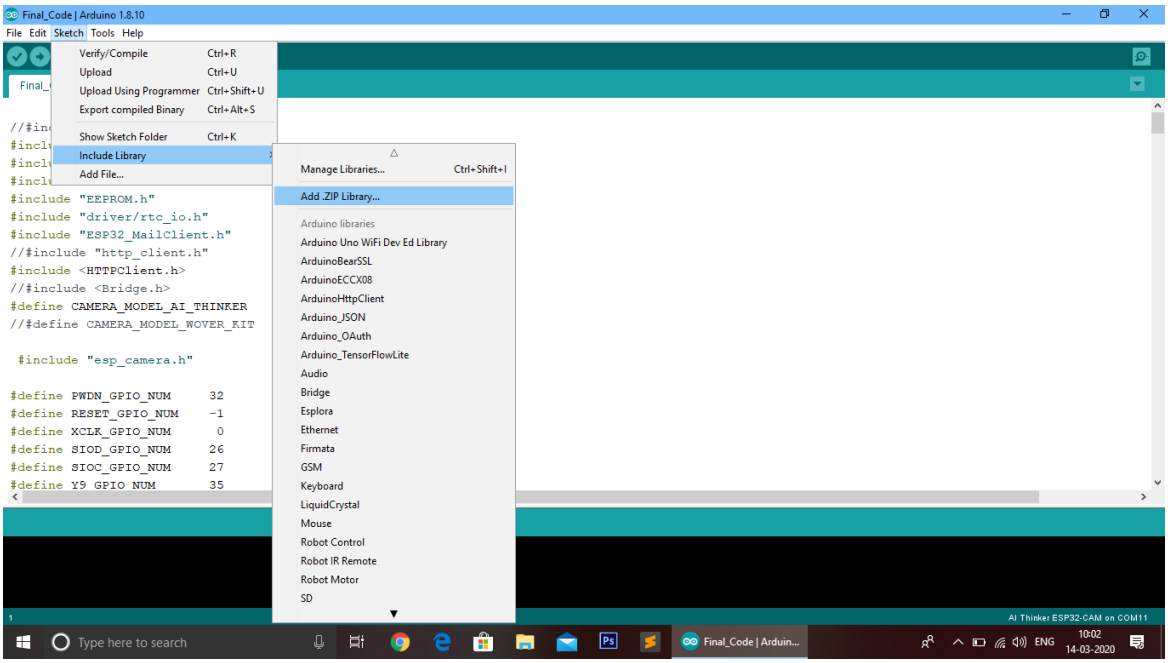
espressif/esp32-camera – GitHub

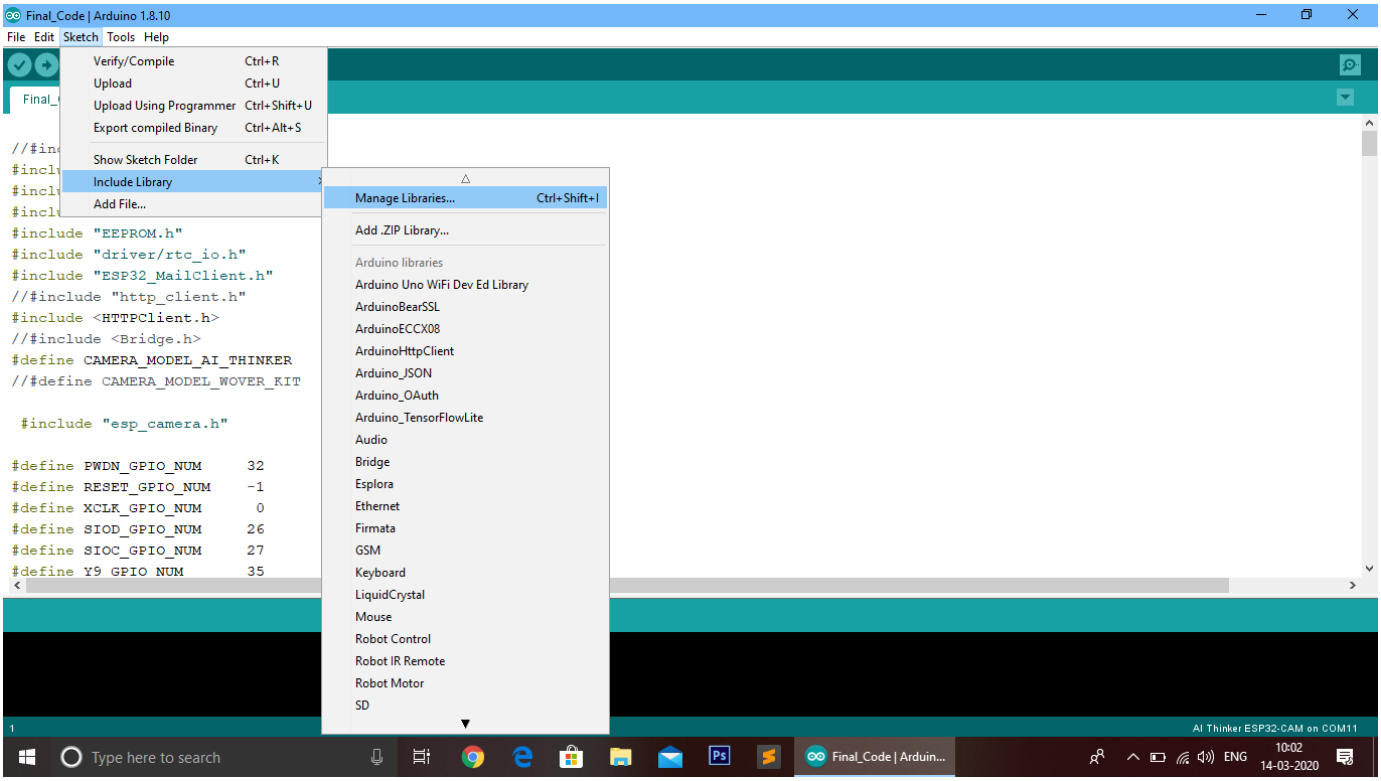
5.1.1. Library Setup:

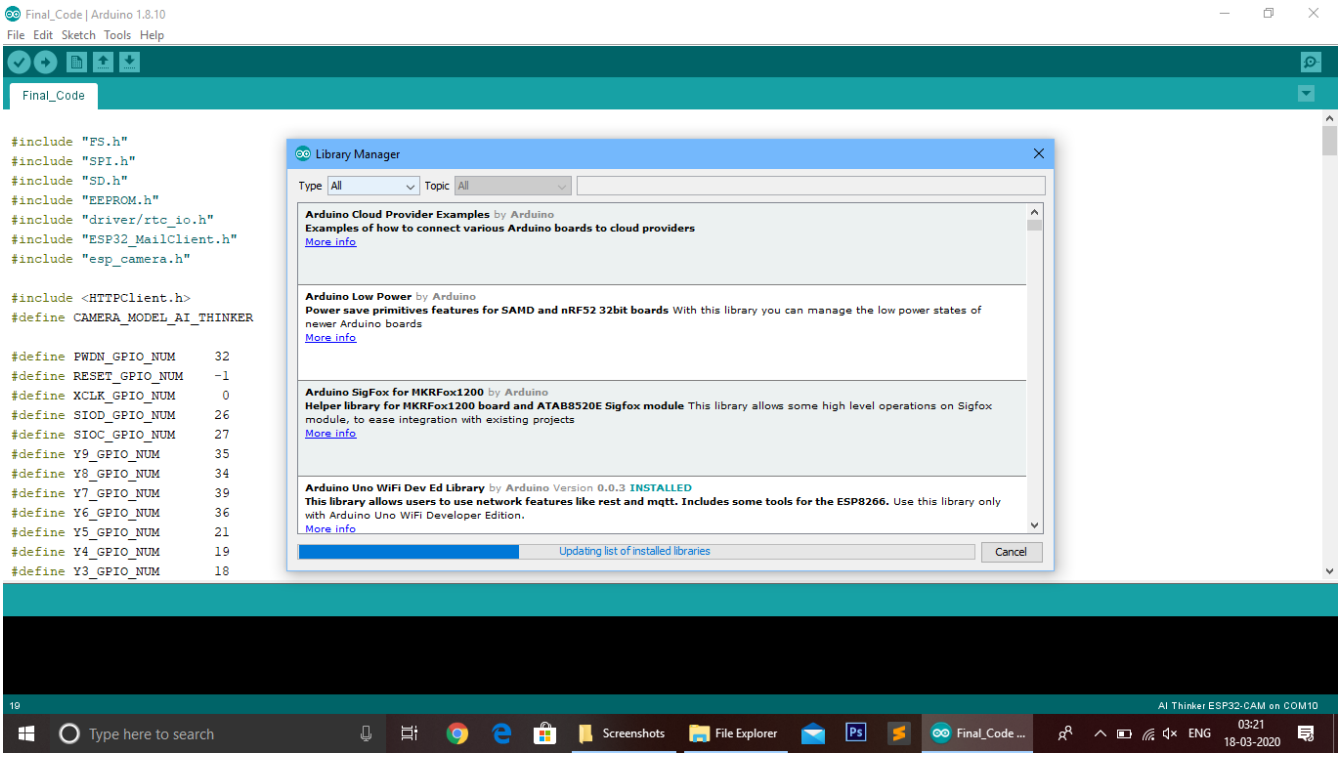
Add zip library:

In the **Arduino** IDE, navigate to Sketch > Include **Library** > Add .ZIP **Library**. At the top of the drop down list, select the option to "Add .ZIP **Library**". You will be prompted to select the **library** you would like to add. Navigate to the .zip file's location and open it.

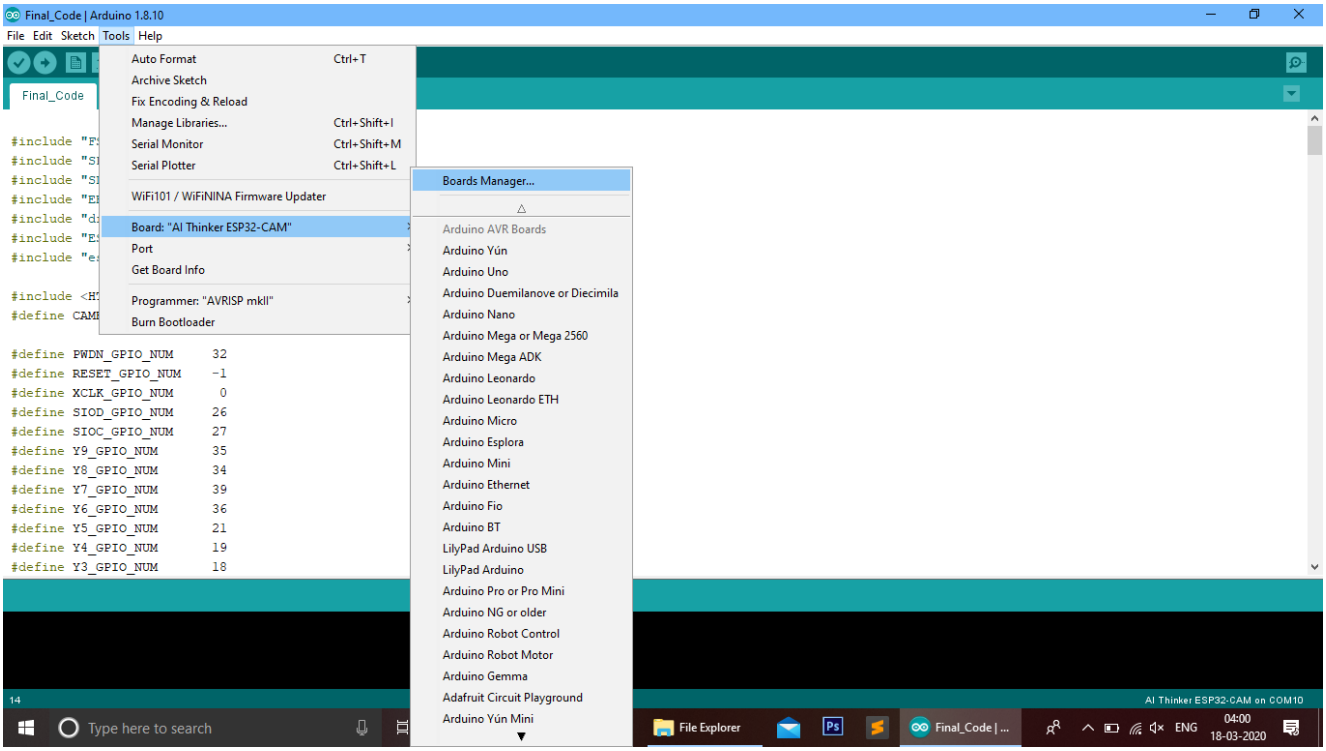
Include Library:

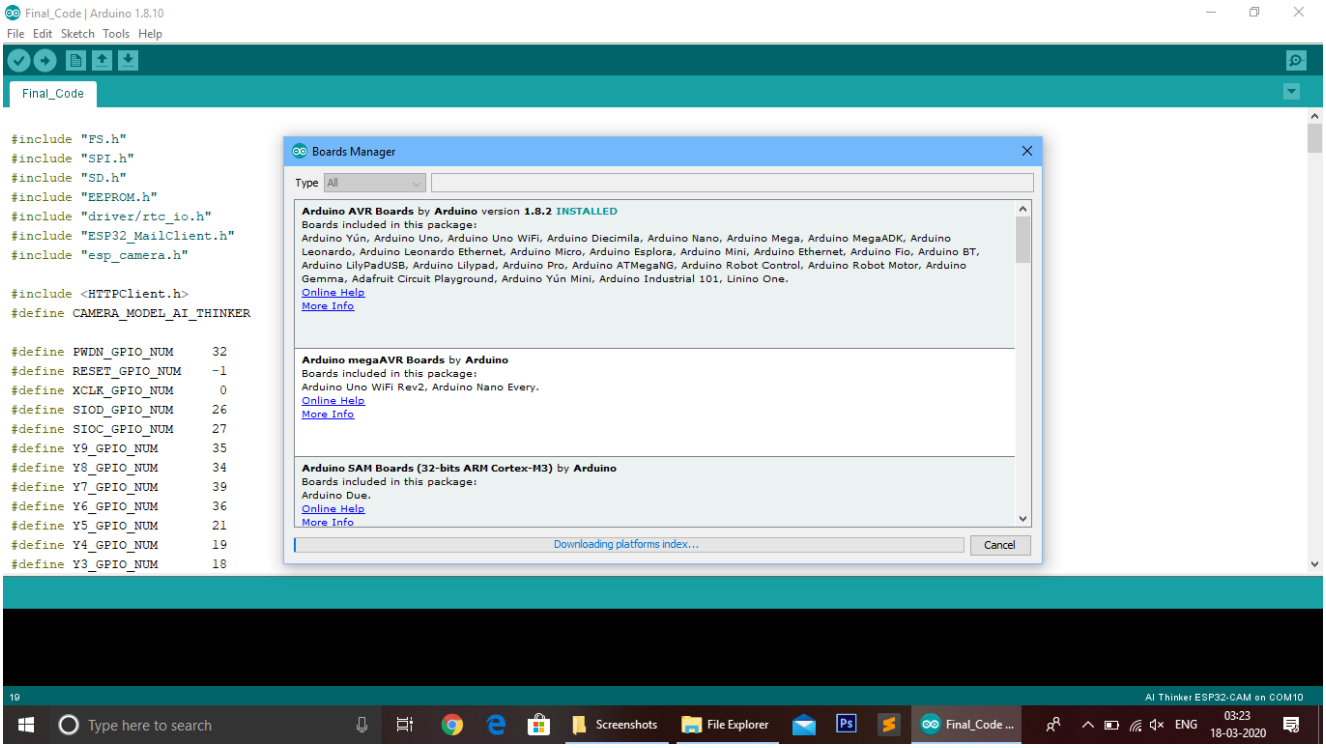



Final_Code | Arduino 1.8.10
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
Arduino Uno WiFi Dev Ed Library
ArduinoBearSSL
ArduinoECCX08
ArduinoHttpClient
Arduino_JSON
Arduino_OAuth
Arduino_TensorFlowLite
Audio
Bridge
Esplora
Ethernet
Firmata
GSM
Keyboard
LiquidCrystal
Mouse
Robot Control
Robot IR Remote
Robot Motor
SD
#include "EEPROM.h"
#include "driver/rtc_io.h"
#include "ESP32_MailClient.h"
//#include "http_client.h"
#include <HTTPClient.h>
//#include <Bridge.h>
#define CAMERA_MODEL_AI_THINKER
//#define CAMERA_MODEL_WOVER_KIT
#include "esp_camera.h"
#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
AI Thinker ESP32-CAM on COM11
Type here to search
Final_Code | Arduin...
ENG
10:02
14-03-2020

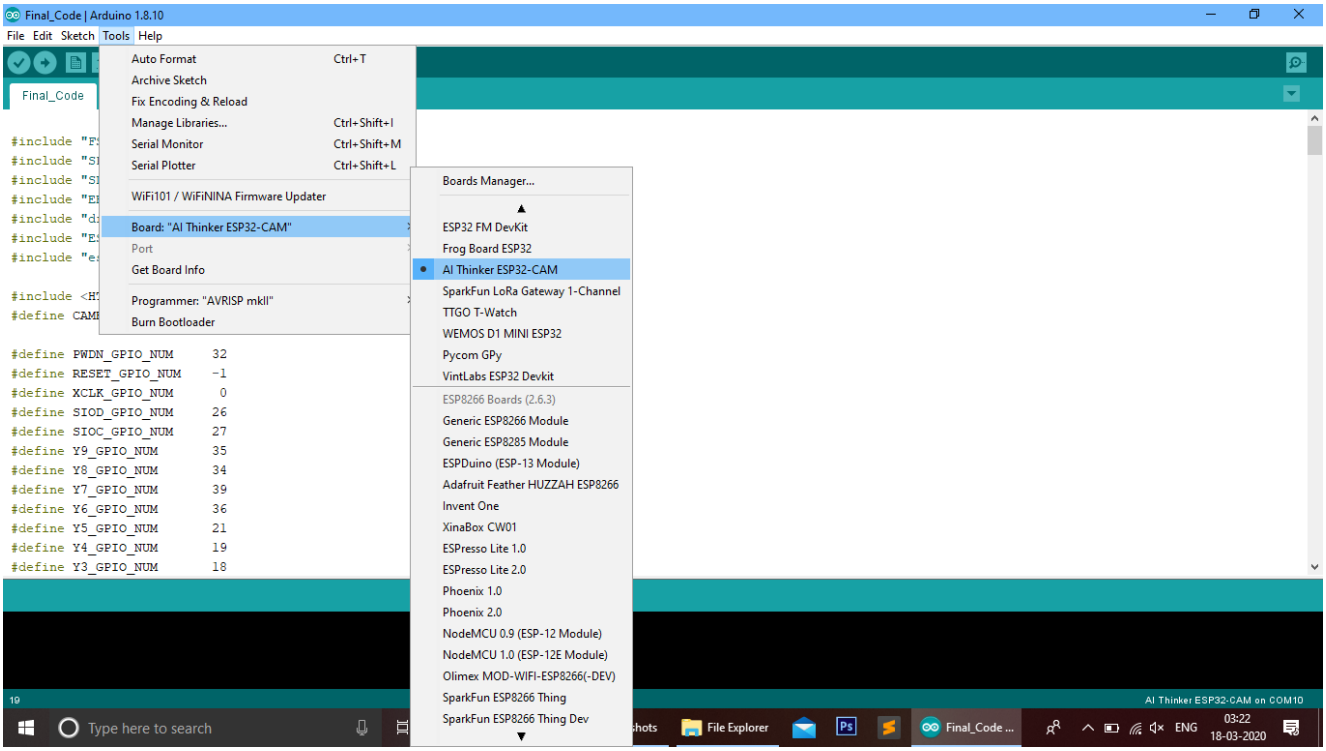

Final_Code | Arduino 1.8.10
File Edit Sketch Tools Help
Final_Code
#include "FS.h"
#include "SPI.h"
#include "SD.h"
#include "EEPROM.h"
#include "driver/rtc_io.h"
#include "ESP32_MailClient.h"
#include "esp_camera.h"
#include <HTTPClient.h>
#define CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM 32
#define 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
Library Manager
Type All Topic All
Arduino Cloud Provider Examples by Arduino
Examples of how to connect various Arduino boards to cloud providers
More info
Arduino Low Power by Arduino
Power save primitives features for SAMD and nRF52 32bit boards With this library you can manage the low power states of newer Arduino boards
More info
Arduino SigFox for MKRFox1200 by Arduino
Helper library for MKRFox1200 board and ATAB8520E Sigfox module This library allows some high level operations on Sigfox module, to ease integration with existing projects
More info
Arduino Uno WiFi Dev Ed Library by Arduino Version 0.0.3 INSTALLED
This library allows users to use network features like rest and mqtt. Includes some tools for the ESP8266. Use this library only with Arduino Uno WiFi Developer Edition.
More info
Updating list of installed libraries
Cancel
AI Thinker ESP32-CAM on COM10
Type here to search
Screenshots
File Explorer
Final_Code ...
ENG
03:21
18-03-2020

5.1.2. Board setup:

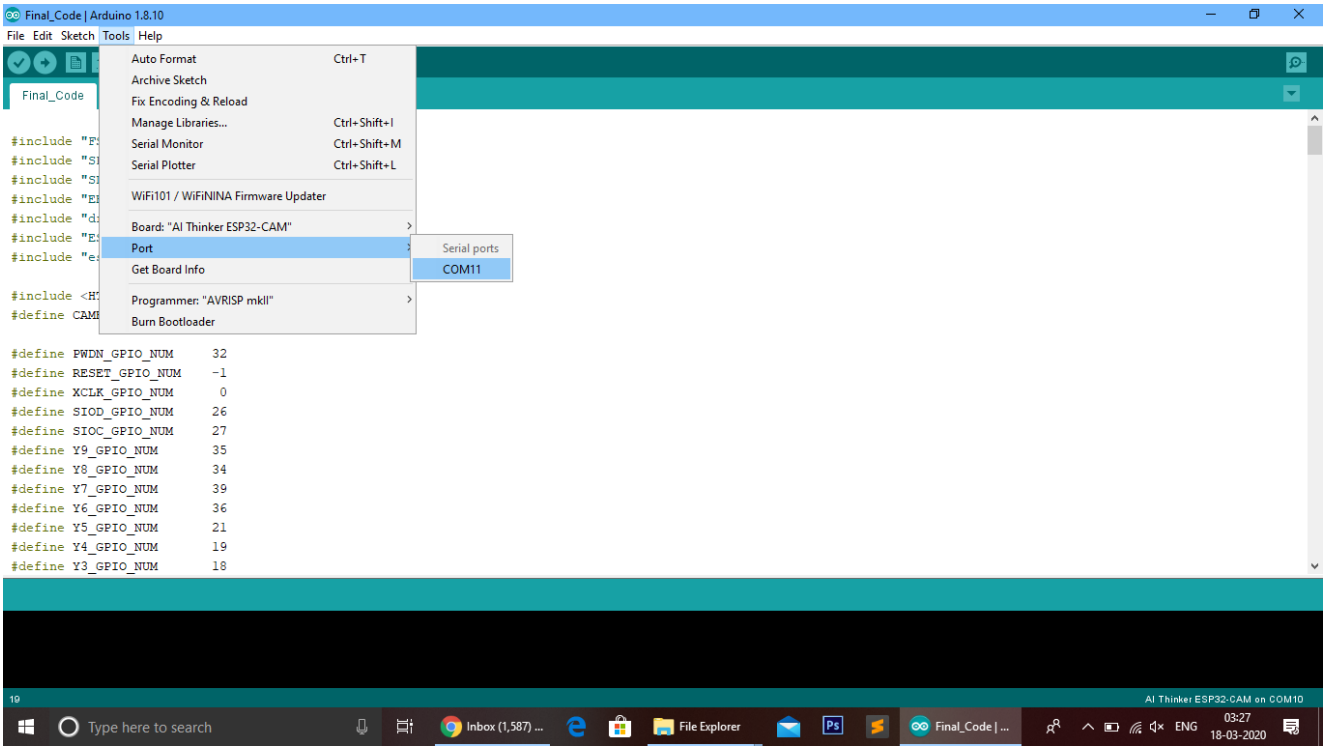




5.1.3. Board seletion:



5.1.4. Port Setup:



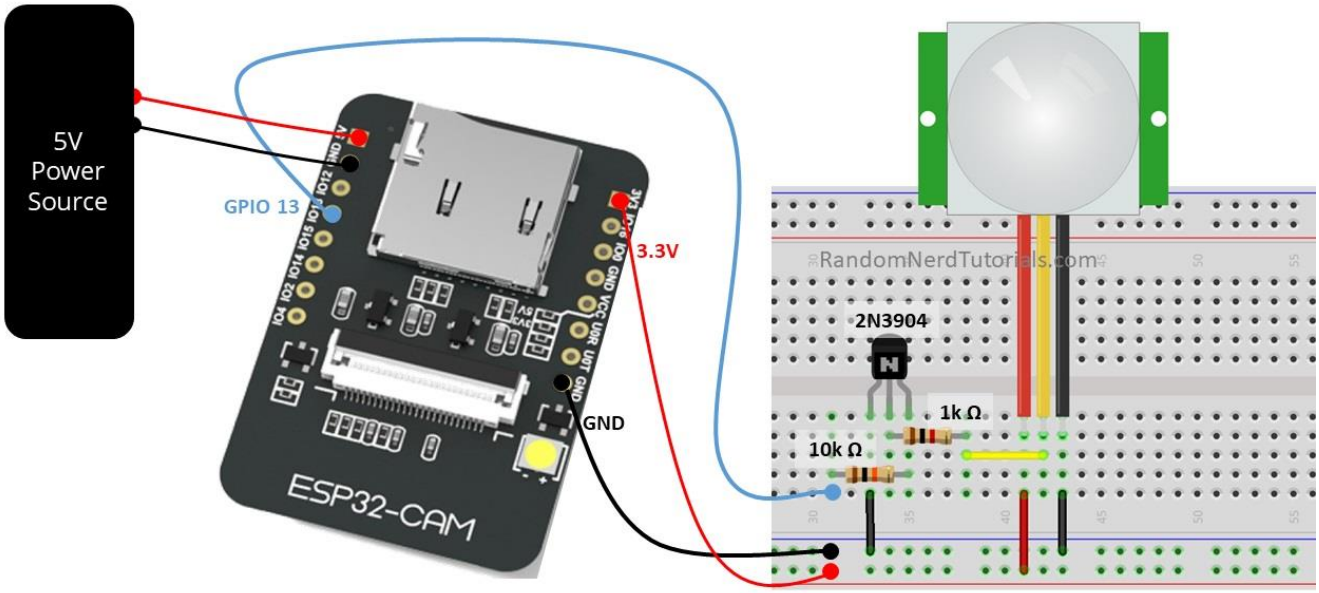
5.2. Hardware Assembly:

Connection between ESP32 Camera and Ch340 Board:

CH340 Board	→	ESP32 Camera
Ground	→	Ground
5V	→	5V
TX	→	RX
RX	→	TX
		GND-D0

After upload code to ESP32 Cam:

PIR-ESP32 Cam Connection:



5.2.1. Circuit Diagram

5.3. CODE SNIPPET:

ARDUINO CODE:

```
#include "FS.h"
#include "SPI.h"
#include "SD.h"
#include "EEPROM.h"
#include "driver/rtc_io.h"
#include "ESP32_MailClient.h"
#include <HTTPClient.h>
#define CAMERA_MODEL_AI_THINKER
#include "esp_camera.h"
#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
#define ID_ADDRESS  0x00
#define COUNT_ADDRESS   0x01
#define ID_BYTE  0xAA
```

```
#define EEPROM_SIZE 0x0F
uint16_t nextImageNumber = 0;
#define WIFI_SSID  "Sasurie-CoE-Shinelogics"
#define WIFI_PASSWORD  "72FB78D0D0"
#define emailSenderAccount"esp32cammodule@gmail.com"
#define emailSenderPassword  "esp32-cam"
#define emailRecipient  "sandhiya73krish@gmail.com"

//The Email Sending data object contains config and data to send
SMTPData smtpData;

//Callback function to get the Email sending status
void sendCallback(SendStatus info);
void setup()
{
Serial.begin(115200);
Serial.println();
Serial.println("Booting...");
pinMode(4, OUTPUT); //GPIO for LED flash
digitalWrite(4, LOW);
rtc_gpio_hold_dis(GPIO_NUM_4);  //disable pin hold if it was enabled before sleeping

//connect to WiFi network
Serial.print("Connecting to AP");
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
while (WiFi.status() != WL_CONNECTED)
{
Serial.print(".");
delay(200);
}
Serial.println("");
```

```
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
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;

//init with high specs to pre-allocate larger buffers
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

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

//set the camera parameters
sensor_t * s = esp_camera_sensor_get();
s->set_contrast(s, 2);  //min=-2, max=2
s->set_brightness(s, 2); //min=-2, max=2
s->set_saturation(s, 2); //min=-2, max=2
delay(100);  //wait a little for settings to take effect

//mount SD card
Serial.println("Mounting SD Card...");
MailClient.sdBegin(14,2,15,13);
```

```
if(!SD.begin())
{
Serial.println("Card Mount Failed");
return;
}


//initialize EEPROM & get file number
if (!EEPROM.begin(EEPROM_SIZE))
{
Serial.println("Failed to initialise EEPROM");
Serial.println("Exiting now");
while(1);  //wait here as something is not right
}
if(EEPROM.read(ID_ADDRESS) != ID_BYTE) /there will not be a valid picture
number
{
Serial.println("Initializing ID byte & restarting picture count");
nextImageNumber = 0;
EEPROM.write(ID_ADDRESS, ID_BYTE);
EEPROM.commit();
}
else //obtain next picture number
{
EEPROM.get(COUNT_ADDRESS, nextImageNumber);
nextImageNumber += 1;
Serial.print("Next image number:");
Serial.println(nextImageNumber);
}
pinMode(4, OUTPUT);  //GPIO for LED flash
digitalWrite(4, HIGH);//turn ON flash LED
```

```
delay(50);

//take new image
camera_fb_t * fb = NULL;

//obtain camera frame buffer
fb = esp_camera_fb_get();
if (!fb)
{
Serial.println("Camera capture failed");
Serial.println("Exiting now");
while(1);  //wait here as something is not right
}

//save to SD card
//generate file path
String path = "/IMG" + String(nextImageNumber) + ".jpg";
fs::FS &fs = SD;

//create new file
File file = fs.open(path.c_str(), FILE_WRITE);
if(!file)
{
Serial.println("Failed to create file");
Serial.println("Exiting now");
while(1); //wait here as something is not right
}
else
{
file.write(fb->buf, fb->len);
EEPROM.put(COUNT_ADDRESS, nextImageNumber);
```

```
EEPROM.commit();
}
file.close();


//return camera frame buffer
esp_camera_fb_return(fb);
Serial.printf("Image saved: %s\n", path.c_str());
pinMode(4, OUTPUT);      //GPIO for LED flash
digitalWrite(4, LOW);  //turn OFF flash LED
rtc_gpio_hold_en(GPIO_NUM_4);  //make sure flash is held LOW in sleep
delay(2000);

//Send SMS
if ((WiFi.status() == WL_CONNECTED)) { //Check the current connection status
HTTPClient http;
http.begin("https://sasurieinfo.tech/autosecuritycamera/");  //Specify the URL
int httpCode = http.GET();   //Make the request
if (httpCode > 0) { //Check for the returning code
{
String payload = http.getString();
Serial.println(httpCode);
Serial.println(payload);
}
else {
Serial.println("Error on HTTP request");
}
http.end(); //Free the resources
}

//send Email
```

```
Serial.println("Sending email...");
smtpData.setLogin("smtp.gmail.com", 465, emailSenderAccount,
emailSenderPassword); //Set the Email host, port, account and password
smtpData.setSender("ESP32-CAM", emailSenderAccount);

//Set the sender name and Email
smtpData.setPriority("Normal");

//Set Email priority or importance High, Normal, Low or 1 to 5 (1 is highest)
smtpData.setSubject("Motion Detected - ESP32-CAM");

//Set the subject
smtpData.setMessage("<div style=\"color:#003366;font-size:20px;\">Security
Alert</div>", true);//Set the message - normal text or html format
smtpData.addRecipient(emailRecipient);

//Add recipients, can add more than one recipient
smtpData.addRecipient("priyankaofficial1620@gmail.com");

/Add recipients, can add more than one recipient
smtpData.addAttachFile(path);
smtpData.addAttachFile("/hello.txt");
smtpData.setFileStorageType(MailClientStorageType::SD);

//Set the storage types to read the attach files (SD is default)
smtpData.setSendCallback(sendCallback);

if (!MailClient.sendMail(smtpData))
//Start sending Email, can be set callback function to track the status
Serial.println("Error sending Email, " + MailClient.smtpErrorReason());
smtpData.empty();
```

```
//Clear all data from Email object to free memory
Serial.println("Entering deep sleep mode");
Serial.flush();
esp_sleep_enable_ext0_wakeup(GPIO_NUM_13, 0); //wake up when pin 13 goes
LOW
delay(100);

//wait for 10 seconds to let PIR sensor settle
esp_deep_sleep_start();
}


void loop()
{


}

void sendCallback(SendStatus msg)    //Callback function to get the Email sending
status
{
Serial.println(msg.info());    //Print the current status
if (msg.success())
{
Serial.println("----------------")
```

PHP CODE:

```
<?php

// Account details
$apiKey = urlencode('kOQLwvmG11w-DxK5TQ1MbfhqFPrl7CB2gMLxC3GWhh');

// Message details
$numbers = array(9789752409, 8110942767);
$sender = urlencode('TXTLCL');
$message = rawurlencode('SecurityAlert from Esp32Cam');
$numbers = implode(',', $numbers);

 // Prepare data for POST request
$data = array('apikey' => $apiKey, 'numbers' => $numbers, 'sender' => $sender,
'message' => $message);


// Send the POST request with cURL
$ch = curl_init('http://api.textlocal.in/send/');
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($ch);
curl_close($ch);
// Process your response here
echo $response;
```

CHAPTER 6
SYSTEM TESTING

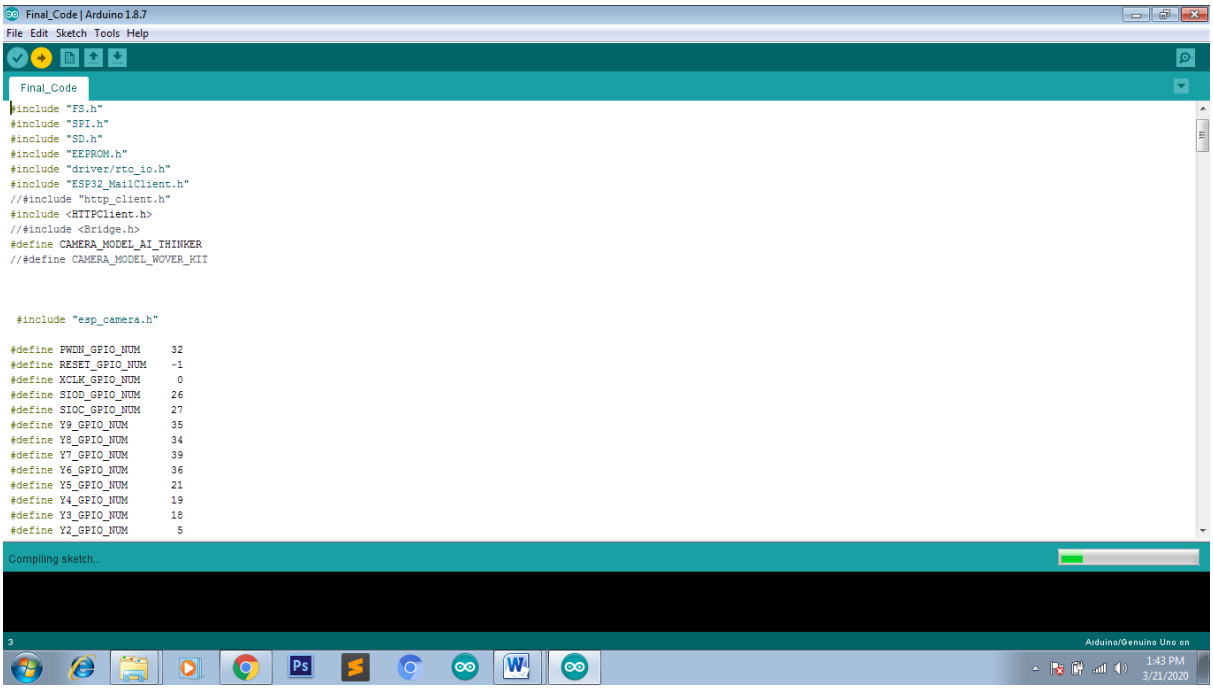
6.1 Software Testing:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1.1 Preview via Monitor :

Serial Monitor :

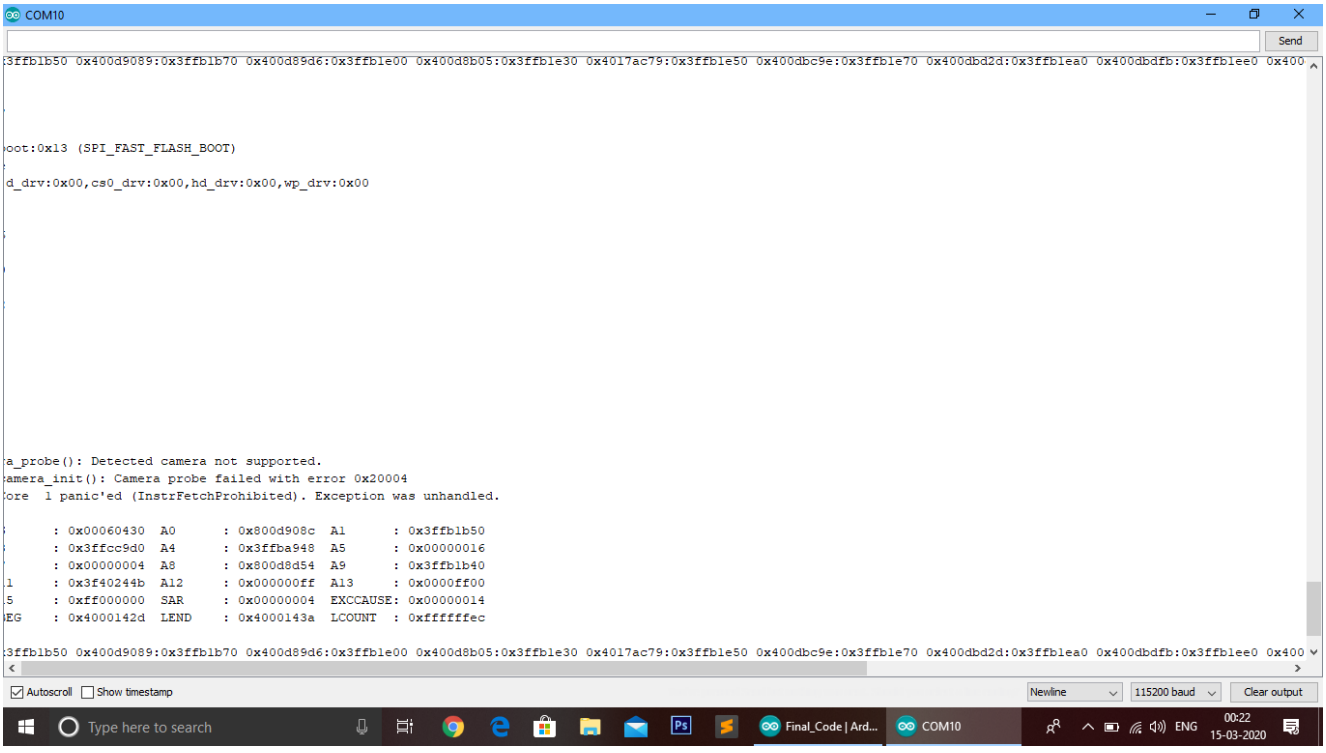
The Arduino IDE has a feature that can be a great help in debugging sketches or controlling Arduino from your computer's keyboard.



The **Serial Monitor** is a separate pop-up window that acts as a separate terminal that communicates by receiving and sending Serial Data. See the icon on the far right of the image above. Serial Data is sent over a single wire (but usually travels over USB in our case) and consists of a series of 1's and 0's sent over the wire. Data can be sent in both directions (In our case on two wires).

The small upper box is where you can type in characters (hit or click "Send")

- The larger area (Corner can be dragged to enlarge) is where characters sent From Arduino will be displayed.- At the bottom are two pulldowns:
- One sets the "line ending" that will be sent to Arduino when you or click Send
- The other sets the Baud Rate for communications. (If this does not match the value set up in your sketch in Setup, characters will be unreadable). Example: Serial.begin(9600); Some sketches or other applications may use a different Baud Rate.



6.1.2. Real time Testing'

Real time testing is done by testing the total set up or system . If any human being , animal or any other living system try to enter the restricted area means then immediately PIR motion detection sensor senses the motion . Here PIR sensor is tested under the coverage area and the distance between the sensor and the motion.

ESP32-CAM is tested in order to check the speed of the waking time of that electronic hardware from sleeping mode and Wi-Fi enabling time of the device. Latency to send mail as well as SMS notification is monitored under this real time testing.

CHAPTER 7

RESULT AND DISCUSSION

7.1. Results:

In the time of motion detection

Serial Monitor:

Booting...

Connecting to AP . . . . . .

WiFi connected.

IP address:198.182.162.5

Mounting SD Card...

Initializing ID byte & restarting picture count

Next image number:12

Image saved:12

Sending email...

Login into Gmail

Attach image 12

Sending….

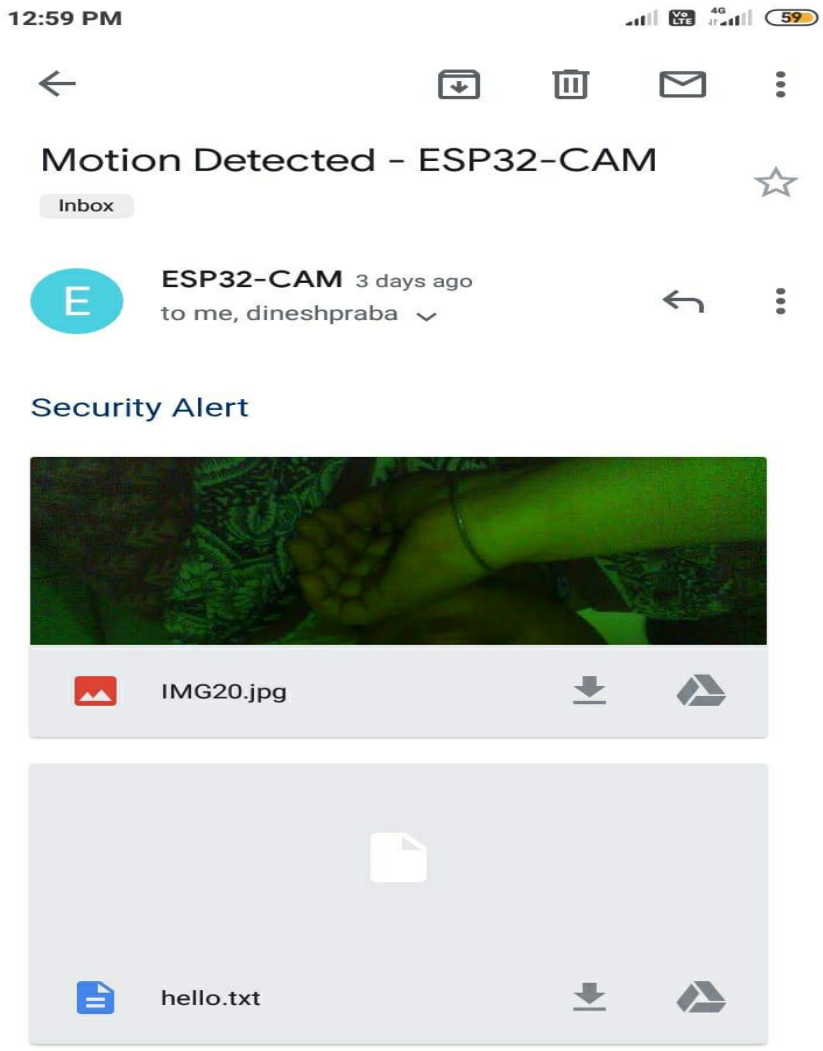
Successfully send to recipient

Entering deep sleep mode

7.2. Screenshots:

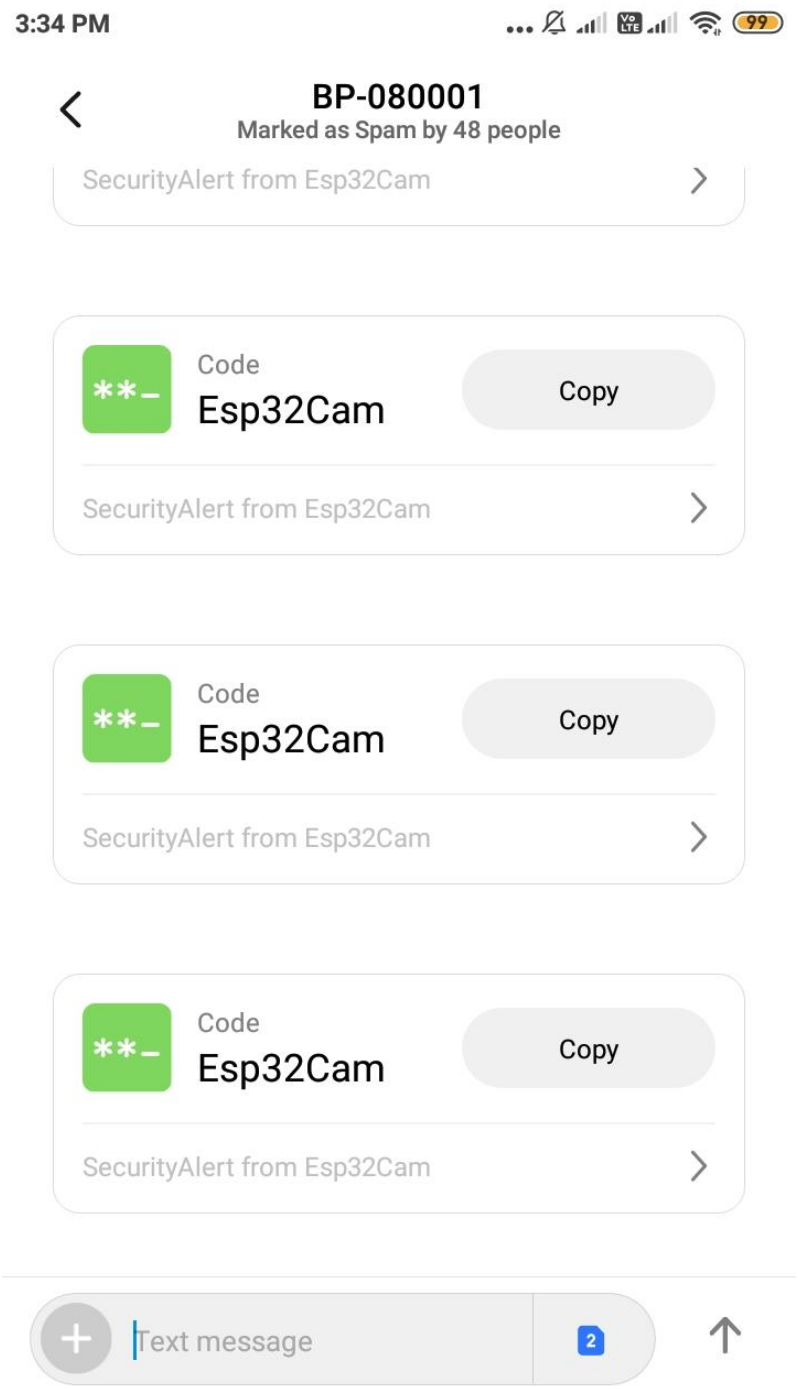
Mail Notification:

If motion gets detected then the system will send email notification in other 30 seconds to authorized user**.**



Message Notification:

If motion gets detected then the system will send SMS notification in other 30 seconds to authorized user**.**



CHAPTER 8
REFERENCE

[1] Shaalini, C.Shanmugam, International Journal of Advanced Research in ComputerScience and Software Engineering 3(12), Volume 3, Issue 12 December-2013, pp. 1070-1077

[2] Shih - Chia Huang An Advanced Motion Detection Algorithm with VideoQuality Analysis for Video Surveillance Systems in the IEEE Transactions onCircuits and Systems for Video Technology, Vol. 21,No.1, January 2011.

[3] T.Deepika , Dr.P.Srinivasa Babu,Motion Detection In Real-Time Video Surveillancewith Movement Frame Capture And Auto Record in International Journalof Innovative Research in Science, Engineering and Technology An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, January 2014

[4] Nishu Singla Motion Detection Based on Frame Difference Method in InternationalJournal of Information and Computation Technology ISSN 0974-2239Volume 4, Number 15 (2014), pp. 1559-1565.

[5] Hezbollah Mohammadian, Elaheh Esfandiarijahromi, Leili Esmailani, OptimizationReal-Time Tracking of Moving Objects Based on Differential andActive Blob in International Conference on Intelligent Systems (ICIS'2012)Penang (Malaysia) May 19-20, 2012.

[6] A.Mohand Sad Allili and B. Djemel Ziou., Object tracking in videos usingadaptive mixture models and active contours, British Machine Vision

Conference. England, pp. 1-10, 2008.

[7] HSU-YUNG CHENG,QUEN-ZONG WU, KUO-CHIN FAN AND BORSHENNJENG,Binarization Method Based on Pixel-level Dynamic Thresholdsfor Change Detection in Image Sequences journal of information science andengineering 22, 545-557 (2006).

[8] Akshay Choudhari, Pranay Kadam,Camera Surveillance System Using MotionDetection and TrackingInternational Journal of Innovative Research in AdvancedEngineering (IJIRAE) ISSN: 2349- 2163Volume 1 Issue 4 (May 2014)KBTCOE, Department of Computer Engineering 2015 38

[9] Amedome Min-DianeyKodjo, YangJinhua, Real-time Moving Object Trackingin Video, 978-1-4673- 2639-1/12/31.00/2012 IEEE.

[10] Sayemul Islam, Md. Saiduzzaman, International Journal of Scientific and EngineeringResearch, Volume 4, Issue 8, August 2013 ISSN 2229-5518.

CHAPTER 9

CONCLUSION

Security is important for our belongings like money, documents, etc. And any time making security for that things is not easy, if theft happen we don’t know who is that we only complain to police. In that situation this circuit is very helpful by taking thief image and notify authorized use by Email(with image) as well as send SMS, so the user gets alert and he/she can take steps according to that and also it is a big clue for police.