BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 416

Microprocessor and Embedded System Laboratory

Final Project Report

Section: B2 Group: 03

Smart Home Security System : Gas And Fire sensor alarm,Password Protection System, Intruder Image Capture

Course Instructors:

1, Sadman Sakib Ahbab
2, Rasin Mohammed Ihtemam

Signature of Instructor: _____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. *You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.*

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: _____**Shadman**_____ **Full Name: Shadman Sobhan** **Student ID: 1906109**	**Signature:** _____**Fahim**_____ **Full Name:Fahim Ahmed** **Student ID:1906110**
Signature: _____**Abrar**_____ **Full Name:Kazi Abrar Mahmud** **Student ID:1906112**	**Signature:** _____**Anik**_____ **Full Name: Anik Biswas** **Student ID: 1906125**

Tbale of Contents

1 Abstract

In the era of technological revolution, life is becoming easier as automation is effecting all aspects of life. Human life is becoming more secure, comfortable, efficient. Implementing technologies and devices like Internet of Things(Io T) , sensors in home system, it is possible to smartify home system making fire and gas detection simpler, also protect home from intruders. Even if any anomaly occurs, the user can know about it. Both hardware and software work in harmony to create a seamless and personalized living experience.

Smart home enables remote access and control of devices through smartphones or voice commands, allowing users to adjust settings, monitor activities, and receive alerts from anywhere. they enhance safety and security through real-time surveillance, intrusion detection, and smart locks, providing peace of mind to homeowners even when they are not at home. However there is a concern regarding privacy breach, but that is possible to overcome. Regarding all factors, smart homes represent a paradigm shift in residential living, offering unparalleled convenience, efficiency, and safety. While they hold tremendous promise for enhancing our quality of life, addressing privacy, security, and usability concerns is imperative to realize their full potential.

2 Introduction

Our primary objective was to create a efficient and user-friendly solution that enhances convenience, efficiency, and security within the home environment. The project aims to automate fire and gas detection, optimize resource usage, and improve overall quality of life for homeowners. Our project includes 5 key features including some preexisting features and some new features. Features like password protection and alarm systems existed before but they were not combined. In this project we combined all those features to make a super module.

IoT is the new standard of smart home security system. To see what is happening in your house while you are away is needed in that age of mobility. This project emphasized on this aspect and we built an IoT system that provides all those features.Given all the features, a homeowner can feel secure about his home even when he is not at home.

3 Design

3.1 Problem Formulation (PO(b)

3.1.1 Identification of Scope

By this project, we can ensure low power system as we need only 3.3V to 5V to run the whole system.

The notification will be sent to phone so there is a constant mobile alarm system. Multiple protection system in one module will also lower the cost and increase the efficiency

3.1.2 Formulation of Problem

We classified our project into 5 major parts:

1. A Decoding Circuit for handling the output of the sensors and identify the fault room number.
2. Building A PCB for this circuit.
3. Making a password protection system via keypad and ESP Cam.
4. Capturing the image of intruder if password attempt is wrong.
5. Sending all data to 'Blynk' app of Android to send notifcation to the user.

3.1.3 Analysis

The sensors we used are active low, that means they give 0V when are triggered. The output of tge sensors are given as a input of a **MUX**. The control signal of the MUX comes from a **clock generator and a counter**. The output of MUX then is sent to **ESP32** to detect the output of every sensor after clock period and detect any trigger and the **trigger source.**The ESP32 also takes input from a **keypad** which is used for input for checking the password.

If a password attempt is wrong, a signal is send to **ESP Cam** which takes picture of the intruder and uploads it on the **google drive.** All data of sensor triggering and password fault alarm goes to the server of **blynk** and gives **notification** to the user. All codes are uploaded to the ESP32.

3.2 Design Method (PO(a))

A multi vibrator was used in astable mode to generate a clock of desired frequency.Then the knowledge of logic circuit such a counter was used to generate a frequency half of the clock. This two clocks were used as the control signal bits of the MUX.Buck Module is used for a regulated power supply.

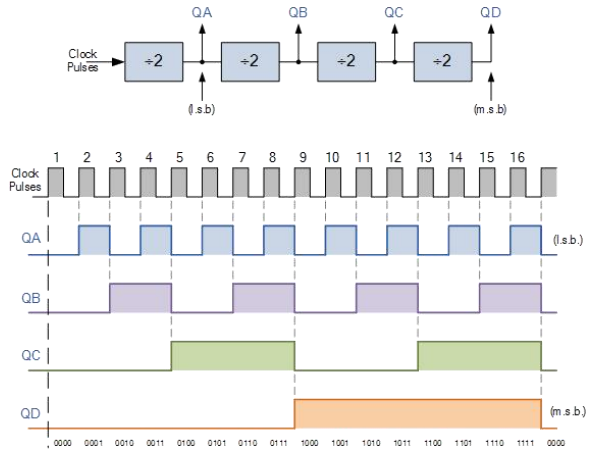

Fig1 : Frequency division by counter

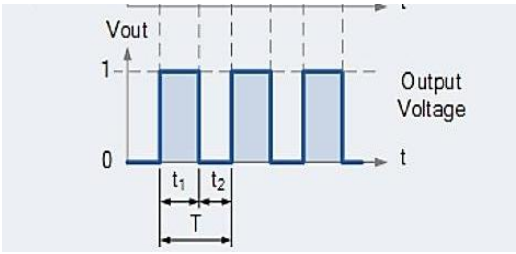

Fig2 : Astable mode multivobrator

ESP32 was used as microcontroller and all codes were written in ARDUINO IDE. The codes were written in a way such that all processes can run parallely without the need for any delay. The idea from STM32F4 was very helpful while using ESP32.

3.3 Circuit Diagram

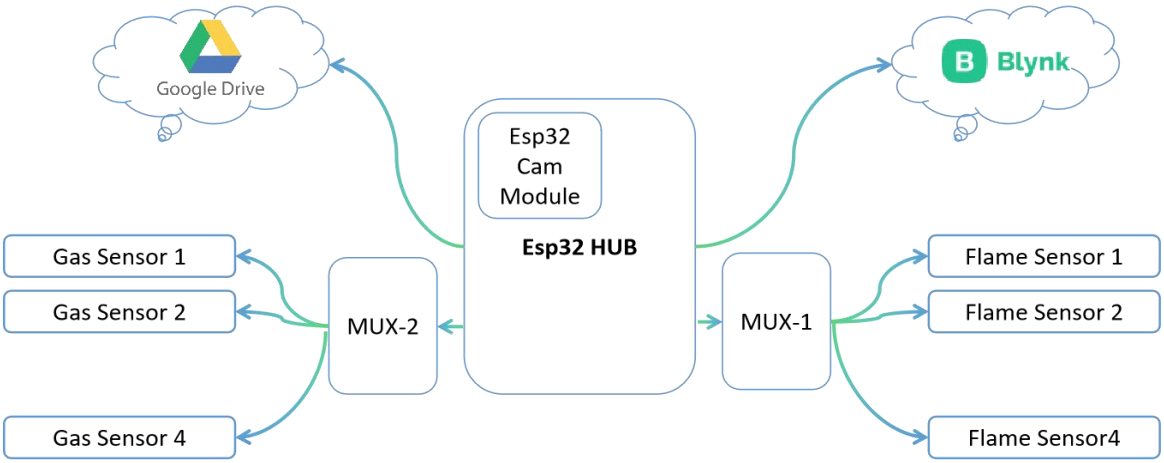

Fig3 : System Configuration

3.4 Simulation Model

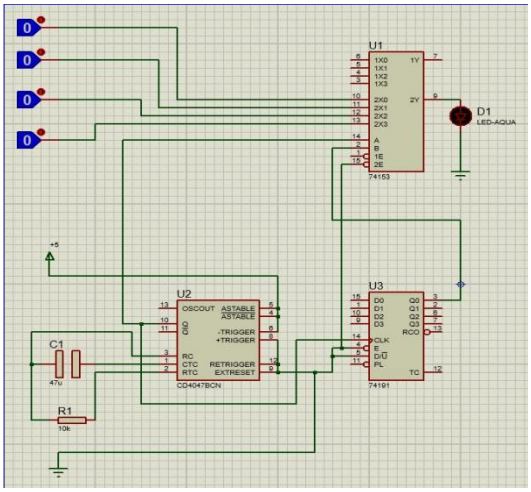

Fig4 : Proteus Simulation Circuit (initial) to check output

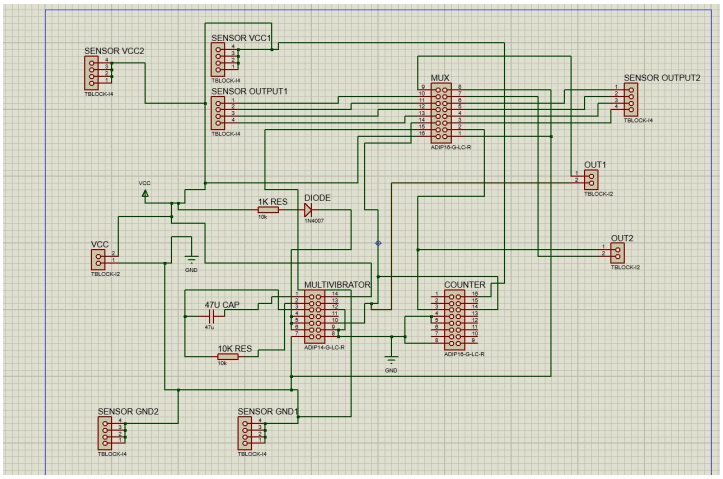

Fig5 : Proteus Simulation Circuit(final) to build PCB

3.5 PCB Design

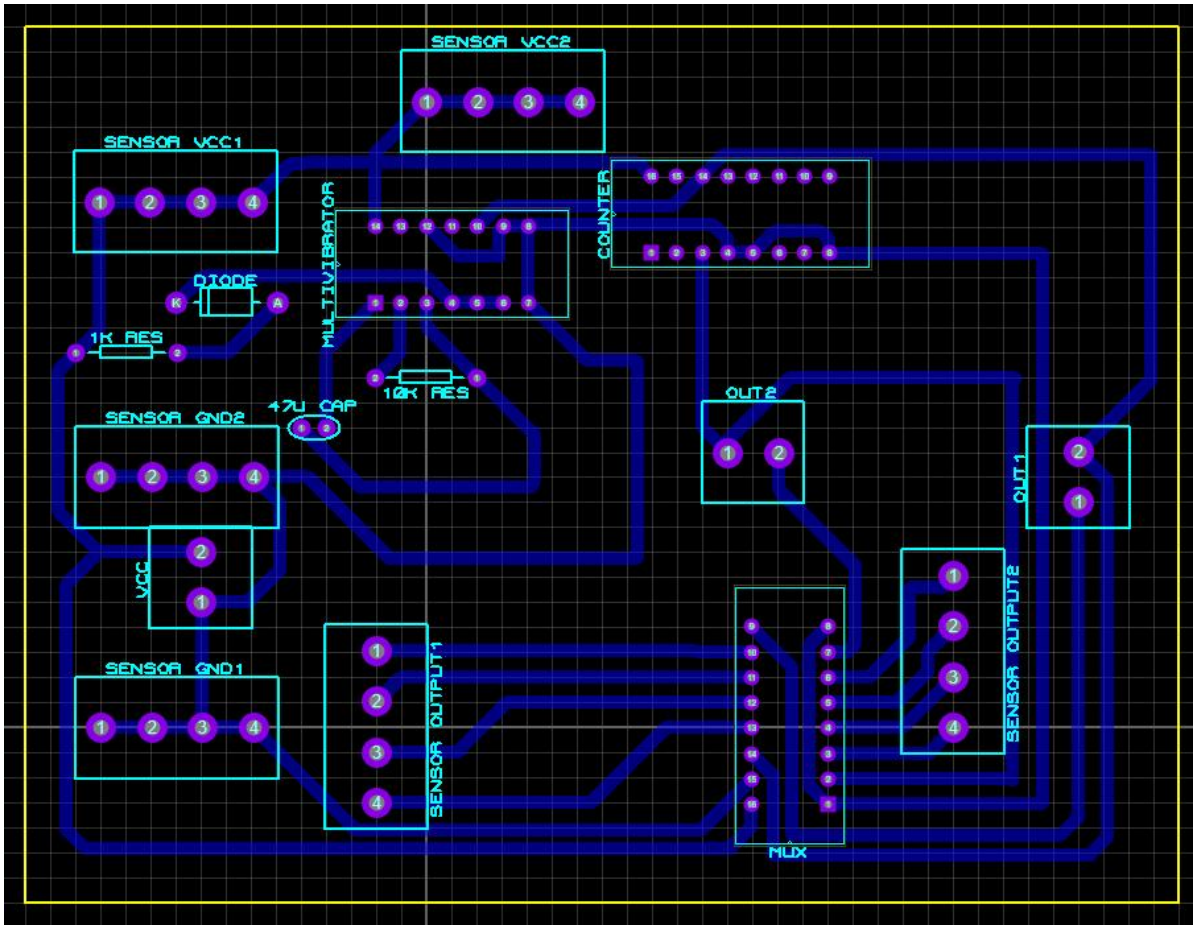

Fig6 : PCB Layout in Proteus

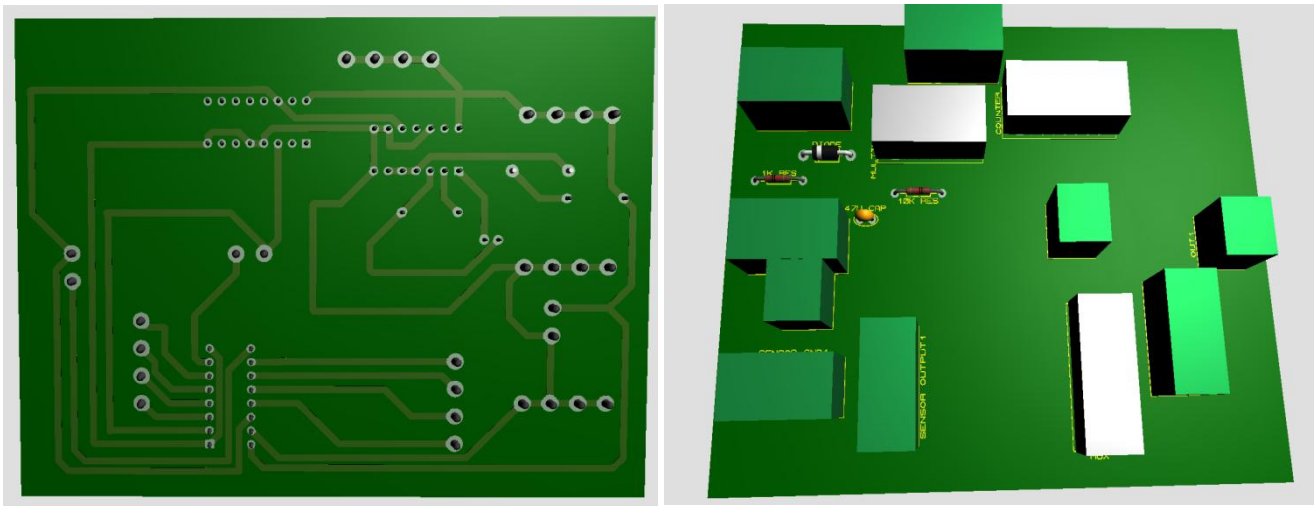

Fig 7: 3D Visualizer (a) Bottom view (b) Top View

3.6Full Source Code of Firmware

```
///CODE IN ESP32 BOARD

#define BLYNK_TEMPLATE_ID "TMPL6jfYERH8m"
#define BLYNK_TEMPLATE_NAME "mainHub1"

#define BLYNK_PRINT Serial
//#define BLYNK_DEBUG
#define APP_DEBUG

#include "BlynkEdgent.h"
#include <Keypad.h>
#include <string.h>
#include <LiquidCrystal_I2C.h>
#include <Preferences.h>

Preferences preferences;

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns


//////////////////////////////SERVER///////////////////////////////////
const int flm_sensIn = 13; //Flame()
const int gas_sensIn = 14;
const int clk1 = 34;  //MSB(white Wire)
const int clk0 = 35;  //LSB(Blue wire)
byte rowPins[ROWS] = {18, 19, 23, 25}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {26, 27, 32, 33}; //connect to the column pinouts of the keypad
static int brd_to_cam = 17;
static int dr_lock = 16;

volatile uint8_t itr = 0;


//char state[]={'0','0'};
int smk[]={0,0,0,0};//Fire
int gas[]={0,0,0,0};
int gas_state = 0;
int filtr_thres = 3;
//Blynk//////////////////////////
int clk_1 =0;
int clk_0=0;
int flm_sens_val=0;
int gas_sens_val=0;
//password///////////////////////
static int pass_size = 6;
char pass_ent[] = {'0','0','0','0','0','0'};// it must be 6 digit pass
char pass[] = {'0','0','0','0','0','0'};

int idx = 0;
char key='\0',key_prev='\0';
int mod_stat = 0;
int pass_entered = 0;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

//////////////////////////////////////////////////////////////////////
LiquidCrystal_I2C lcd_i2c(0x27, 16, 2);
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
BlynkTimer timer;

void shiftLeft(char array[], int size) {
    for (int i = 0; i < size - 1; i++) {
        array[i] = array[i + 1];
    }
    idx++;
}

void myTimer()
{
  // This function describes what will happen with each timer tick
  // e.g. writing sensor value to datastream V0-V7
  Blynk.virtualWrite(V4, gas_state);
  if(gas_state == 1)
    Blynk.logEvent("Gas_Leakage_Alert","Possible Gas leakage in Room 1 !");
  if(gas_state == 2)
    Blynk.logEvent("Gas_Leakage_Alert","Possible Gas leakage in Room 2 !");
  if(gas_state == 3)
```

```
    Blynk.logEvent("Gas_Leakage_Alert","Possible Gas leakage in Room 3 !");
  if(gas_state == 4)
    Blynk.logEvent("Gas_Leakage_Alert","Possible Gas leakage in Room 4 !");
  // if(gas[0]==1 ||gas[1]==1 ||gas[2]==1 ||gas[3]==1 )
  //   Blynk.virtualWrite(V4, 1);
  // else
  //   Blynk.virtualWrite(V4, 0);


  Blynk.virtualWrite(V0, smk[0]);
  Blynk.virtualWrite(V1, smk[1]);
  Blynk.virtualWrite(V2, smk[2]);
  Blynk.virtualWrite(V3, smk[3]);

}


void setup()
{

  // Debug console. Make sure to have the same baud rate selected in serial monitor
  timer.setInterval(1000L, myTimer);
  Serial.begin(115200);
  delay(100);
  //Input
  pinMode(flm_sensIn, INPUT);
  pinMode(gas_sensIn, INPUT);
  pinMode(brd_to_cam,OUTPUT);
  pinMode(dr_lock,OUTPUT);
  //output
  //pinMode(clk2, OUTPUT);
  pinMode(clk1, INPUT);
  pinMode(clk0, INPUT);

  lcd_i2c.init(); // initialize the lcd
  lcd_i2c.backlight();

  lcd_i2c.clear();
  lcd_i2c.setCursor(0, 0);
  lcd_i2c.print("Enter Password:");
  lcd_i2c.setCursor(4, 1);
  lcd_i2c.print(">      <");
  preferences.begin("PASSWORD", false); //read+Write

  pass[0] = preferences.getChar("pass0",'0');
  pass[1] = preferences.getChar("pass1",'0');
  pass[2] = preferences.getChar("pass2",'0');
  pass[3] = preferences.getChar("pass3",'0');
  pass[4] = preferences.getChar("pass4",'0');
  pass[5] = preferences.getChar("pass5",'0');

  preferences.end();

  //currently stored
  Serial.printf("Current value: %c %c %c %c %c %c\n\n", pass[0],pass[1],pass[2],pass[3],pass[4],pass[5]);

  //.....................................................

  BlynkEdgent.begin();
  Serial.println(configStore.wifiPass);

}

void loop() {
  BlynkEdgent.run();

  // runs BlynkTimer
  timer.run();
  clk_1 = digitalRead(clk1);
  clk_0 = digitalRead(clk0);
  flm_sens_val = digitalRead(flm_sensIn);// pin-13 MUX.....................
  gas_sens_val = digitalRead(gas_sensIn);// pin-14 MUX.....................


  if(clk_1 == 0 && clk_0 == 0){
    smk[0] = !flm_sens_val;
    gas[0] = !gas_sens_val;
    gas_state = gas[0]? 1:0;}
  if(clk_1 == 0 && clk_0 == 1){
    smk[1] = !flm_sens_val;
    gas[1] = !gas_sens_val;
    gas_state =  gas[1]? 2:0;}
  if(clk_1 == 1 && clk_0 == 0){
    smk[2] = !flm_sens_val;
    gas[2] = !gas_sens_val;
    gas_state =  gas[2]? 3:0;}
```

```
  if(clk_1 == 1 && clk_0 == 1){
    smk[3] = !flm_sens_val;
    gas[3] = !gas_sens_val;
    gas_state =  gas[3]? 4:0;}


////////////////////////////////////////////////////////////////////////////////



 key = keypad.getKey();// Read the key

  // Print if key pressed
  if (key){
    Serial.print("Key : ");
    Serial.println(key);
    //Serial.println(!strncmp(pass, pass_ent, pass_size));
    if(idx==0&&!mod_stat)
      lcd_i2c.clear();
    if(!mod_stat)
    {
      lcd_i2c.setCursor(0, 0);
      lcd_i2c.print("Enter Password:");
      lcd_i2c.setCursor(0+4, 1);
      lcd_i2c.print(">");
      lcd_i2c.setCursor(7+4, 1);
      lcd_i2c.print("<");
    }



    if(key != '*' && key != '#'){
    shiftLeft(pass_ent, pass_size);
    pass_ent[pass_size-1] = key;
    lcd_i2c.setCursor(idx+4, 1);
    lcd_i2c.print(key);

    }

  }
  if((key =='#')&&!strncmp(pass, pass_ent, pass_size)&&pass_entered) //pass-compare mode '#*'
  {
    Serial.println("Pass_com_mode:Dorr_open");

    lcd_i2c.clear();
    lcd_i2c.setCursor(0, 1);
    lcd_i2c.print("Welcome Home");

    digitalWrite(dr_lock,1);
    delay(1000);
    digitalWrite(dr_lock,0);


    lcd_i2c.clear();
    lcd_i2c.setCursor(0, 1);
    lcd_i2c.print("Welcome Home");

    pass_entered = 0;
    idx = 0;
  }else if((key =='#')&&strncmp(pass, pass_ent, pass_size)&&pass_entered){
    Serial.println("Wrong Password");
    lcd_i2c.clear();
    lcd_i2c.setCursor(0, 1);
    lcd_i2c.print("Wrong Password !");


    digitalWrite(brd_to_cam, 1);
    delay(500);
    digitalWrite(brd_to_cam, 0);

    lcd_i2c.clear();
    lcd_i2c.setCursor(0, 0);
    lcd_i2c.print("Enter Password:");

    pass_entered = 0;
    idx = 0;
  }
  if((key =='')&&!strncmp(pass, pass_ent, pass_size)&&pass_entered) //pass-modification mode '#'
  {
    Serial.println("Pass_mod_mode:activated");
    //Serial.println(compareArrays(pass, pass_ent, pass_size));
    lcd_i2c.setCursor(0, 0);
    lcd_i2c.print("Change Password:");
    mod_stat = 1;
    pass_entered = 0;
    idx = 0;
```

```
  }
  if(mod_stat && pass_entered) // modifying password
  {
    pass[0] = pass_ent[0];
    pass[1] = pass_ent[1];
    pass[2] = pass_ent[2];
    pass[3] = pass_ent[3];
    pass[4] = pass_ent[4];
    pass[5] = pass_ent[5];
    //strcpy(pass,pass_ent);


    //.............................................................
    preferences.begin("PASSWORD", false);


    preferences.putChar("pass0", pass[0]);
    preferences.putChar("pass1", pass[1]);
    preferences.putChar("pass2", pass[2]);
    preferences.putChar("pass3", pass[3]);
    preferences.putChar("pass4", pass[4]);
    preferences.putChar("pass5", pass[5]);
    // Close the Preferences
    preferences.end();
    //''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Serial.println("_____Eiditted_____");
    delay(100);
    lcd_i2c.clear();
    lcd_i2c.setCursor(0, 1);
    lcd_i2c.print("Pass Modified ");
    for(int i=0;i<pass_size;i++)
    {
      lcd_i2c.setCursor(i+5, 0);
      lcd_i2c.print(pass_ent[i]);
    }

    Serial.print(pass[0]);
    Serial.print(pass[1]);
    Serial.print(pass[2]);
    Serial.print(pass[3]);
    Serial.print(pass[4]);
    Serial.println(pass[5]);
    mod_stat = 0;
    pass_entered = 0;
  }



  if(idx==6)
  { // 6 digit pass_enter complete

    Serial.print(pass_ent[0]);
    Serial.print(pass_ent[1]);
    Serial.print(pass_ent[2]);
    Serial.print(pass_ent[3]);
    Serial.print(pass_ent[4]);
    Serial.println(pass_ent[5]);

    pass_entered = 1;
    idx = 0;
  }
////////////////////////////////////////////////////////////////////////////////
  delay(10);
  //Serial.println( configStore.wifiPass);
}




////////////////////////////////////////////////////////////////////////////////
//CODE IN ESP32 CAM

#include "esp32-hal-ledc.h"
#include "config.h"
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "Base64.h"
#include "esp_camera.h"
// Time
#include "time.h"
#include "lwip/err.h"
#include "lwip/apps/sntp.h"
// MicroSD
#include "driver/sdmmc_host.h"
```

```
#include "driver/sdmmc_defs.h"
#include "sdmmc_cmd.h"
#include "esp_vfs_fat.h"
//
#include <EEPROM.h>
// OTA
#include <ArduinoOTA.h>
#include <ESPmDNS.h>

const char *folder = FOLDER;
const char *scriptID = SCRIPT_ID;
const char *wifiNetwork = WIFI_NETWORK;
const char *password = PASSWORD;
const char *host = HOST;
const int port = PORT;
const int sleepTime = SLEEP_TIME;
const int sleepTimeNight = SLEEP_TIME_NIGHT;
const int nightStart = NIGHT_START;
const int nightEnd = NIGHT_END;
const int waitingTime = WAITING_TIME;
const int otaWaitingTime = OTA_WAITING_TIME;

#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
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

// 4 for flash led or 33 for normal led
#define LED_GPIO_NUM       4

#define CONFIG_LED_ILLUMINATOR_ENABLED 1

// LED FLASH setup
#if CONFIG_LED_ILLUMINATOR_ENABLED

#define LED_LEDC_CHANNEL 2 //Using different ledc channel/timer than camera
#define CONFIG_LED_MAX_INTENSITY 255

#endif

bool isNight = false;
int pic_num = 0;
bool connectWifi()
{
  WiFi.mode(WIFI_STA);

  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(wifiNetwork);
  WiFi.begin(wifiNetwork, password);
  int retry = 20;
  while (WiFi.status() != WL_CONNECTED && retry > 0)
  {
    Serial.print(".");
    delay(500);
    retry--;
  }
  return WiFi.status() == WL_CONNECTED;
}
void IRAM_ATTR isr() {
  pic_num++;
}


void setup()
{
  //WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);
  //pinMode(14,INPUT_PULLDOWN);
  pinMode(14, INPUT_PULLDOWN);
          attachInterrupt(14, isr, RISING);
```

```
  if (!connectWifi())
  {
    //ota();
      Serial.println("Faild to Connect");
  }
}

void loop()
{

  if(pic_num>0)
  {

    camera_fb_t *fb = getPicture();
    if (WiFi.status() == WL_CONNECTED)
    {
      initTime();
      sendPhotoDrive(fb);
      pic_num--;
    }
    else
    {
      Serial.println("Cannot connect to Wifi");
    }
    esp_camera_fb_return(fb);
    //while(digitalRead(14));
  }
}



esp_err_t initCamera()
{
  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())
  {
    Serial.println("PSRAM found");
    config.frame_size = FRAMESIZE_SVGA;
    //config.frame_size = FRAMESIZE_240X240;
    config.jpeg_quality = 2;
    config.fb_count = 2;
  }
  else
  {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 2;
    config.fb_count = 1;
  }
    // 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 err;
  }

  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  s->set_brightness(s, 2); // up the brightness just a bit
  s->set_saturation(s, 0); // lower the saturation
  s->set_exposure_ctrl(s,12);
```

```
  #if defined(LED_GPIO_NUM)
  setupLedFlash(LED_GPIO_NUM);
  #endif
  //return esp_camera_init(&config);
  return err;
}

void sendPhotoDrive(camera_fb_t *fb)
{
  Serial.println("Connect to " + String(host));
  WiFiClientSecure client;
  client.setInsecure();
  if (client.connect(host, port))
  {
    Serial.println("Connection successful");
    Serial.println("Sending image to Google Drive.");
    Serial.println("Size: " + String(fb->len) + "byte");

    String url = "/macros/s/" + String(scriptID) + "/exec?folder=" + String(folder);

    client.println("POST " + url + " HTTP/1.1");
    client.println("Host: " + String(host));
    client.println("Transfer-Encoding: chunked");
    client.println();

    int fbLen = fb->len;
    char *input = (char *)fb->buf;
    int chunkSize = 3 * 1000; // must be multiple of 3
    int chunkBase64Size = base64_enc_len(chunkSize);
    char output[chunkBase64Size + 1];

    Serial.println();
    int chunk = 0;
    for (int i = 0; i < fbLen; i += chunkSize)
    {
      int l = base64_encode(output, input, min(fbLen - i, chunkSize));
      client.print(l, HEX);
      client.print("\r\n");
      client.print(output);
      client.print("\r\n");
      delay(100);
      input += chunkSize;
      Serial.print(".");
      chunk++;
      if (chunk % 50 == 0)
      {
        Serial.println();
      }
    }
    client.print("0\r\n");
    client.print("\r\n");

    Serial.println("Waiting for response.");
    long int StartTime = millis();
    while (!client.available())
    {
      Serial.print(".");
      delay(100);
      if ((StartTime + waitingTime * 1000) < millis())
      {
        Serial.println();
        Serial.println("No response.");
        break;
      }
    }
    Serial.println();
    while (client.available())
    {
      Serial.print(char(client.read()));
    }
  }
  else
  {
    Serial.println("Connected to " + String(host) + " failed.");
  }
  client.stop();
}

void initTime()
{
  sntp_setoperatingmode(SNTP_OPMODE_POLL);
  sntp_setservername(0, "pool.ntp.org");
  sntp_init();
  // wait for time to be set
  time_t now = 0;
```

```
  struct tm timeinfo;
  timeinfo = {0};
  int retry = 0;
  const int retry_count = 10;
  while (timeinfo.tm_year < (2016 - 1900) && ++retry < retry_count)
  {
    Serial.printf("Waiting for system time to be set... (%d/%d)\n", retry, retry_count);
    delay(2000);
    time(&now);
    localtime_r(&now, &timeinfo);
  }
  setenv("TZ", "CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00", 1);
  tzset();

  if (timeinfo.tm_hour >= nightStart || timeinfo.tm_hour < nightEnd)
  {
    isNight = true;
  }
}
void setupLedFlash(int pin)
{
    #if CONFIG_LED_ILLUMINATOR_ENABLED
    ledcSetup(LED_LEDC_CHANNEL, 5000, 8);
    ledcAttachPin(pin, LED_LEDC_CHANNEL);
    #else
    log_i("LED flash is disabled -> CONFIG_LED_ILLUMINATOR_ENABLED = 0");
    #endif
}


void deepSleep()
{
  Serial.println("Deep sleep for " + String(sleepTime) + " seconds");
  Serial.flush();
  int sleep = sleepTime;
  if (isNight)
  {
    sleep = sleepTimeNight;
  }
  esp_sleep_enable_timer_wakeup(sleep * 1000000);
  esp_deep_sleep_start();
}


camera_fb_t *getPicture()
{
  esp_err_t err = initCamera();


  if (err != ESP_OK)
  {
    Serial.printf("Camera init failed with error 0x%x", err);
    deepSleep();
  }
  //delay(150);//<----Improves image quality for high-res image////////
  //enable_led(true);
  ledcWrite(LED_LEDC_CHANNEL, CONFIG_LED_MAX_INTENSITY);
  vTaskDelay(200 / portTICK_PERIOD_MS); // The LED needs to be turned on ~150ms before the call to
esp_camera_fb_get()
  camera_fb_t *fb = esp_camera_fb_get();
  ledcWrite(LED_LEDC_CHANNEL, 0);
  //enable_led(false);

  if (!fb)
  {
    Serial.println("Camera capture failed, check if camera is connected");
    deepSleep();
  }

  return fb;
}
```

Table:Code for the ESP32 Board amd ESP Cam

4 Implementation

4.1 Description

CD4047 Multivibrator was used to generate a clock in astable mode.

IC74HC191 was used to generate the MSB of control signal by dividing the original clock frequency in half.

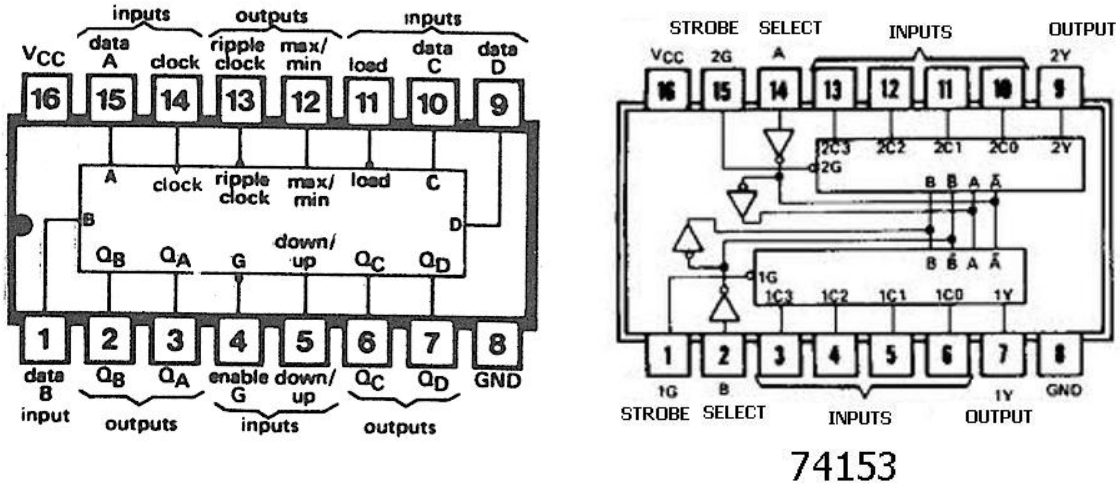

Fig 8 : IC 74191 and CD4047

ESP32 module was used to do the coding and detect the fault room number.
Then the output was sent to the blynk.

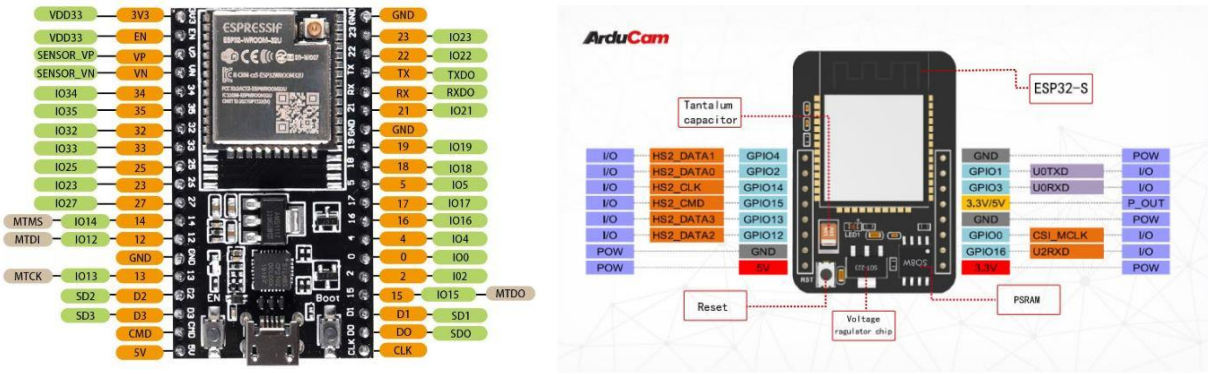

Fig 9 : ESP32 board and Cam

In Password system, we used a 4*4 keypad to input the password. The password length was fixed to simplify the approach. The password input was constaly shown on LCD. When any wrong attempt on password was found, a signal from esp32 board was sent to esp cam to capture an image.

The image is then uploaded to the google drive. The password is saved on EPROM so that during resetting of esp32, the password remains.

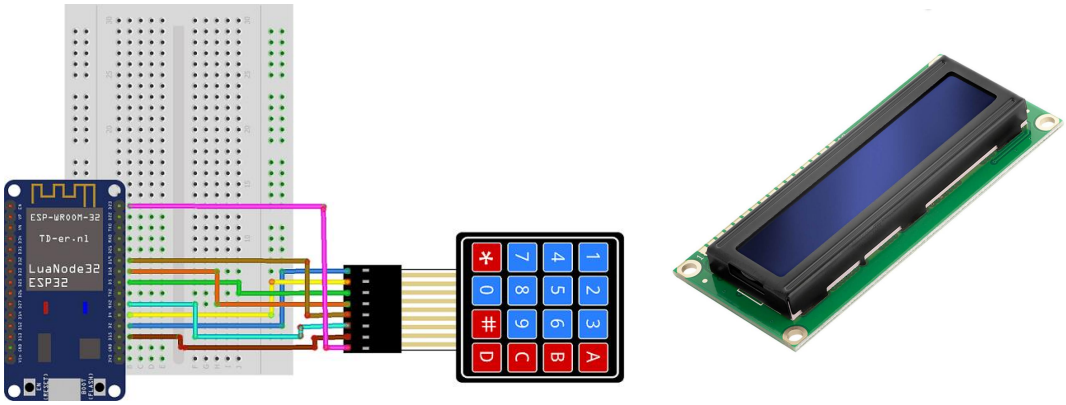

*Fig 10: (a) 4*4 Keypad with ESP32 (b) LCD*

Final Project setup :

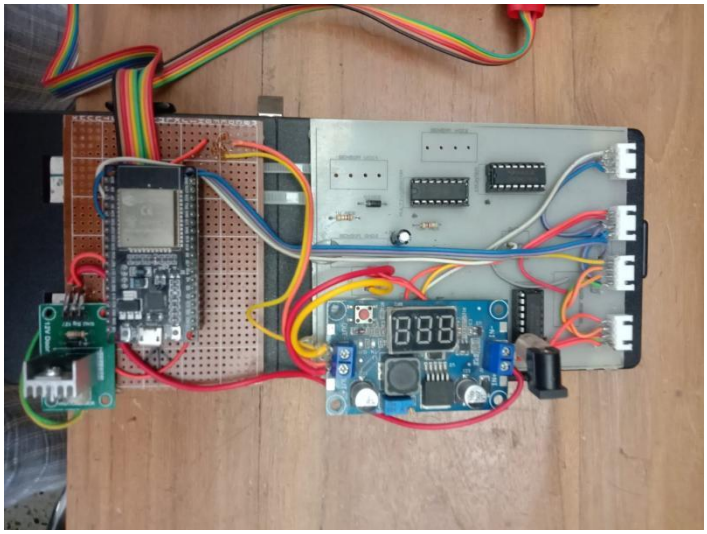

Fig 11 : Central decoding PCB with esp32 and buck converter

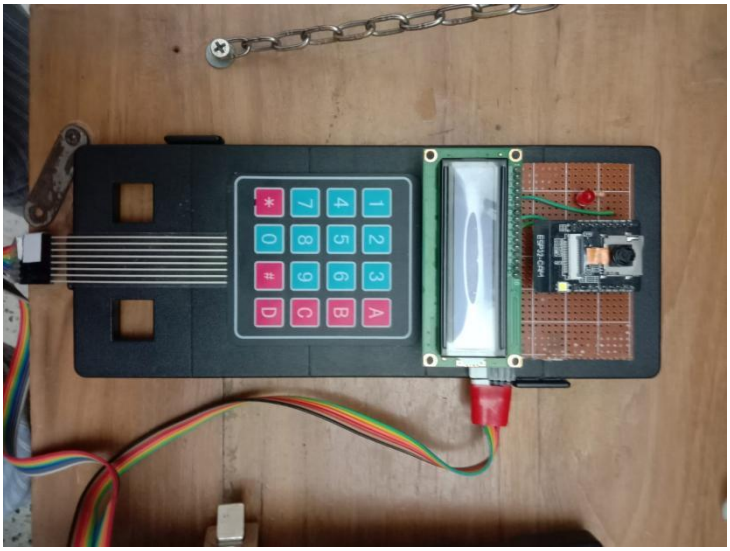

Fig 12 : Keypad, ESP cam and LCD

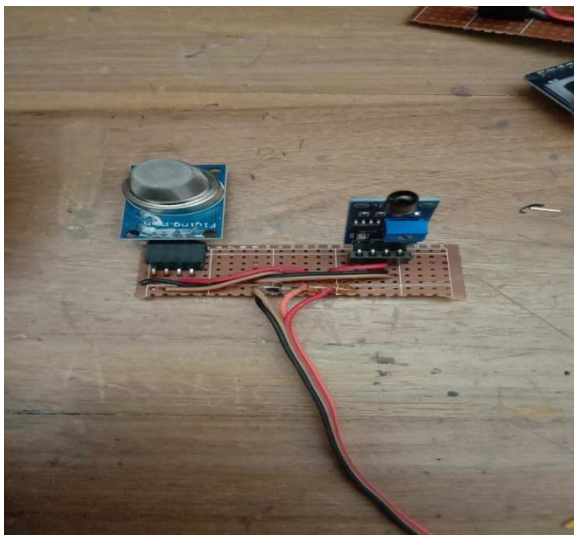

Fig 13: Sensor configuration

Sensors are placed in each 4 rooms in pair. So the sensors are placed in pairs in veroboard. Their output goes to PCB.

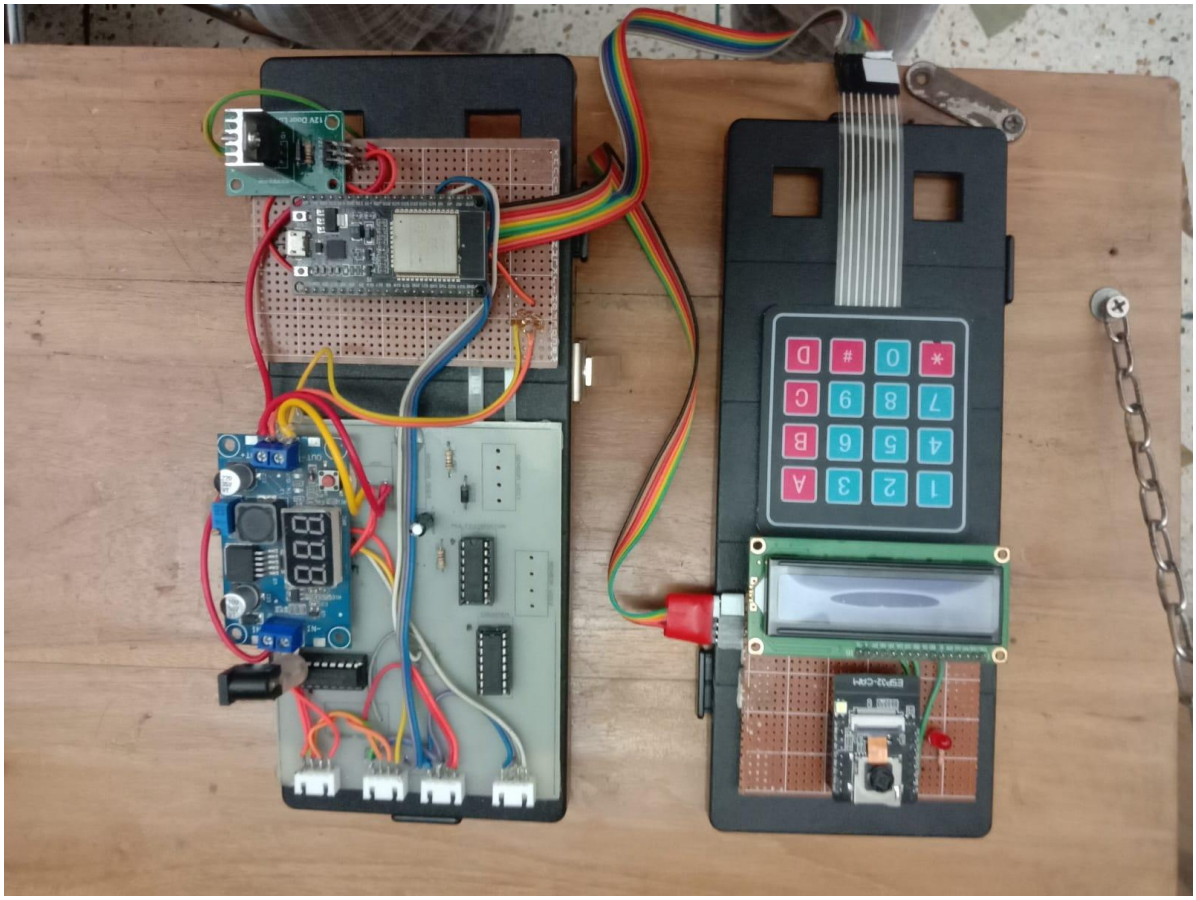

Fig 14: Total Circuit

5 Design Analysis and Evaluation

5.1 Novelty

We developed an ioT based scalable solution to fire hazard monitoring and home owner access control. Our main focus was to develop a user ready product ,that anyone can use with only having access to Local Area network credentials.We developed a Time Division Multiplexing (TDM) based central decoder system which allows us to monitor the state of 8 individual sensor sequentially using only 2 GPIO ports.This method was implemented to improve the scalibility of our project.We also developed Over-The-Air(OTA) WiFi provisioning based solution so that the user could enter his/her credentials directly from mobile app. Our home security system also consist of an outdoor unit.It is essentially a keypad based door lock unit. Whenever someone tries to break-in by trying random password it would trigger our outdoor camera to take a snapshot of the possible intruder.

5.2 Design Considerations (PO(c))

Our project considers both technological and environmental impacts. Developing a smart ioT based system was not our only target ,rather assessing its environmental social impacts was also an important aspect of our project.

5.2.1 Considerations to environment

We designed our system using minimum possible resource. Our target was to develop a system that would not only be smart monitoring system but also consume minimum power.This is important because if we expect a mass adoption of this system we must consider the overall enviromental impact too.
In order to reduce e-waste we provide the user with an option of smaller package with less sensor hubs.Hence if a potential consumer needs only to monitor a single room , he/she can opt for fewer number of hubs.

5.2.2 Considerations to cultural and societal needs

In order to maintain an undisturbed social harmony ,our system provides a built in digital secure lock system. Hence ill-doers can not break into someones house. Our secure solution prevents theft before incurring and fulfills societal needs of safe and secure environment.

5.3 Investigations (PO(d))

We designed our system to monitor each room separately.Our monitoring parameter consists of (1) Possibility of any type of Fire & (2) Any type of potential Gas leakage .Here we deployed a detachable sensor hub for each room. Our system currently can monitor up-to 4 rooms.Each sensor hub consists of Flame Sensor(KY-026 Series) & a Gas sensor(MQ-9 series).

5.3.1 Design of Experiment

Our first task was to calibrate the sensor and analyse their limitations.As further down the line we implemented a digital MUX based decoder ,so we need to calibrate the digital outputs of each sensor module.

For Fire detection the sensor that we are using is Flame Sensor(KY-026 Series).The internal circuit diagram for our Flame detection module is given below:

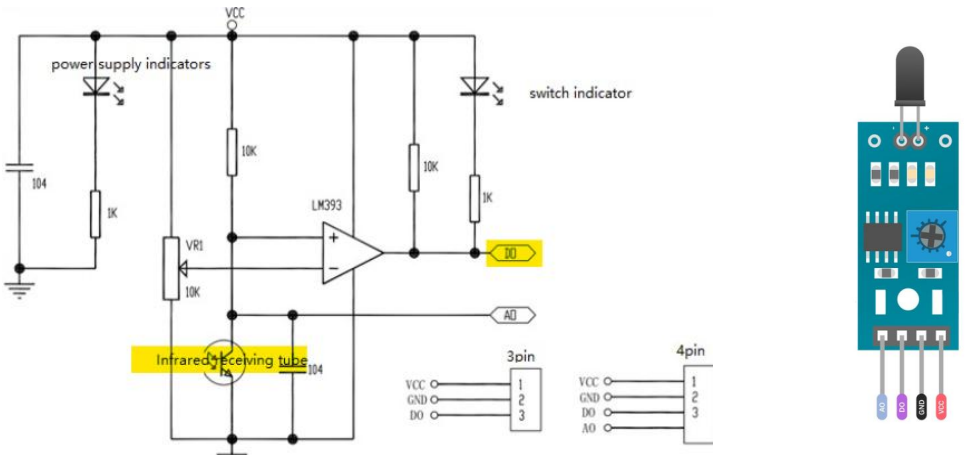

Fig 15 : *(a) Internal Circuit Diagram (b) Flame Sensor(KY-026 Series)*

The flame sensor is an active low sensor. So whenever its analog output crosses a certain threshold it outputs zero volt.The threshold can be selected by the end-user by rotating the potentiometer knob.

Testing & Result: Our tests suggests that the sensitivity of this sensor depends somewhat on ambient lighting .Initially we calibrated each sensor so that we can detect small lighter fire from close distance for the sake of our project demonstration.

For gas leakage detection we used MQ-9 Gas sensor.During our market research we came across several types of gas sensors. A comprehensive table consisting of all types of available gas sensor is given below:

Sensor Name	***Gas to measure***	***Sensor Name***	***Gas to measure***
MQ-2	Methane, Butane, LPG, Smoke	MQ-7	Carbon Monoxide
MQ-3	Alcohol, Ethanol, Smoke	MQ-8	Hydrogen Gas
MQ-4	Methane, CNG Gas	MQ-9	Carbon Monoxide, flammable gasses
MQ-5	Natural gas, LPG	MQ131	Ozone

Our project deals with a house hold monitoring system, hence detection of flambe gas was

our priority. Our findings suggests that MQ-9 series is best suited for this application.

Similar to the flame sensor The MQ-9 sensor module is an active low sensor. So whenever its analog output crosses a certain threshold it outputs zero volt.

Calibration of MQ-9 gas sensor: This sensor measures the gas concentration based on resistance ratio. This ratio includes R0 (sensor resistance in 1000ppm concentration of LPG) and Rs (Internal resistance of the sensor which changes by gas concentration).For our project we only require binary outputs from the gas sensor .Hence a comprehensive ppm calibration is not needed. But a word of cautions is that the user must preheat the sensors in order to get valid detection output.

Now after finalizing our sensor calibrations , we designed the main circuit .The circuit consists of three main part.

(1) MUX based decoder circuit
(2) (2)ESP-32 base main hub for internet of Things (ioT)
(3) Esp32-camera for intruder image capture.

5.3.2 Data Collection

We automated our alert system using Blynk server based Automation® and event & notification method.From technical point of view we configured 4 virtual pin™ of integer data types and the 5th virtual pin is configured to handle enumerate type data .

The data from fire alarm is passed through individual virtual pins ,Blynk server has limit of 5 virtual pins ,so we have decided to configure the 4 individual data stream from each gas sensor into a single enumerate data type .The enumerate data is declared in blynk server.

enum V4{“No Gas Leaked” ,”Gas Leaked Room 1” , ”Gas Leaked Room 2”, ”Gas Leaked Room 3”, ”Gas Leaked Room 4”};

The code snippet that accomplishes this task is given below:

```
void myTimer()
{
  // This function describes what will happen with each timer tick
  // Passing Gas sensor data through enumrate data type
  Blynk.virtualWrite(V4, gas_state);
  if(gas_state == 1)
    Blynk.logEvent("Gas_Leakage_Alert","Possible Gas leakage in Room 1 !");
  if(gas_state == 2)
    Blynk.logEvent("Gas_Leakage_Alert","Possible Gas leakage in Room 2 !");
  if(gas_state == 3)
    Blynk.logEvent("Gas_Leakage_Alert","Possible Gas leakage in Room 3 !");
  if(gas_state == 4)
    Blynk.logEvent("Gas_Leakage_Alert","Possible Gas leakage in Room 4 !");


  // Passing Flame sensor data to the cloud
  Blynk.virtualWrite(V0, smk[0]);
  Blynk.virtualWrite(V1, smk[1]);
  Blynk.virtualWrite(V2, smk[2]);
  Blynk.virtualWrite(V3, smk[3]);

}
```

Fig 16 : *code snippet for Blynk server communication.*

We configured our system to send data every 1000ms to blynk cloud. This slotted transmission of data is necessary .Because without it continuous loop transmission of data could spam the Blynk.Cloud with thousands of messages from our hardware. When this happens, Blynk.Cloud will cut off the connection between the hardware and the server.

Local data acquisition :Our system consists of 4X4 membrane keypad for realizing a digital lock system .it uses a scanning matrix technique to determine which key has been pressed.

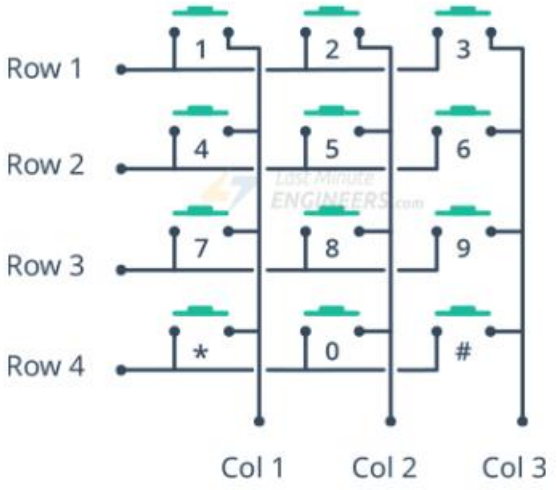

Fig 17 : membrane keypad schematic

When the button is pressed, one of the rows is connected to one of the columns, allowing current to flow between them. When the key '4' is pressed, for instance, column 1 and row 2 are connected.By identifying which column and row are connected, we can determine which button has been pressed.

*For intruder detection we used Esp32-cam :*Our camera is configured to to take snapshot of any possible intruder and immediately upload it to Google drive .Currently Google scripts do not support direct uploading of binary files. Photo is split into chunks which are converted to base64 and uploaded as text.

The google script that we needed to deploy is given below:

```
function doPost(e) {
  const name =
    Utilities.formatDate(new Date(), 'GMT+0', 'yyyyMMdd-HHmmss') + '.jpg'
  const subFolderName = Utilities.formatDate(new Date(), 'GMT+0', 'yyyyMMdd')
  const folderName = e.parameters.folder || 'ESP32-CAM'
  const data = Utilities.base64Decode(e.postData.contents)
  const blob = Utilities.newBlob(data, 'image/jpg', name)

  let folder
  const folders = DriveApp.getFoldersByName(folderName)
  if (folders.hasNext()) {
    folder = folders.next()
  } else {
    folder = DriveApp.createFolder(folderName)
  }
  let subFolder
  const subFolders = folder.getFoldersByName(subFolderName)
  if (subFolders.hasNext()) {
    subFolder = subFolders.next()
  } else {
    subFolder = folder.createFolder(subFolderName)
  }
  const file = subFolder.createFile(blob)
  return ContentService.createTextOutput('Done')
}
```

Fig 18 : Script for receiving text converted images in Drive.

5.3.3 Results and Analysis

Currently our user interface gets static message of whether Fire and Gas leakage hazard is detected or not .Currently there are two user UI .One is for direct web access and another is for The App that the user can download from the app store.

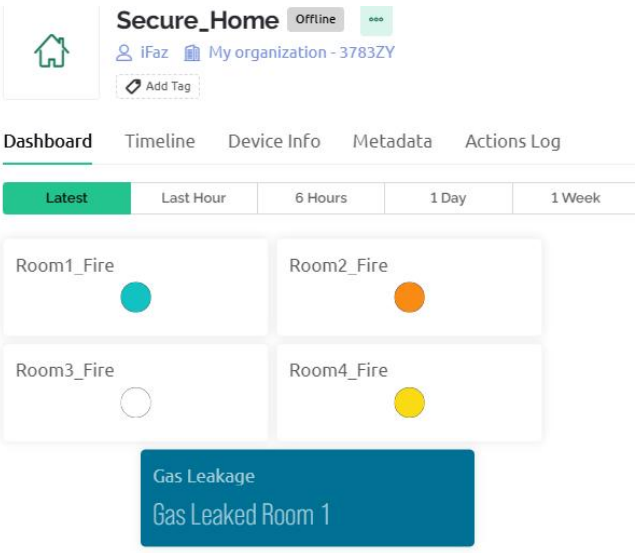

***Fig 19** : User interface for remote monitoring*

Note: Even if the user is not actively using the the application ,the user will receive a push notification in any hazardous event.Then by accessing the app the user can asses the situation remotely any call for emergency help if necessary.

5.4 Limitations of Tools (PO(e))

Our system uses state of the art ioT tool-kit from Blynk® Cloud .Blynk is an open source platform for developing ioT based application.But it has its own limitations.currently we are running the free version of Blynk ,hence we are limited to maximum 2 device use.This means currently our free subscription to blynk platform will allow us to deal with only two end user.But blynk itself is a scalable platform .So after successful beta testing we can easily upgrade our subscription and support multiple end user.A paid subscription would also allow us to mitigate the virtual pin limitation and push any future software update if necessary.

5.5 Impact Assessment (PO(f))

The main goal of our project was to develop a scalable home monitoring and security system.We also approached in way that allows the end-user to choose the number of detachable sensor hubs that is suited for his requirements.

5.5.1 Assessment of Legal Issues

Our home security system is capable of taking snapshots if it detects multiple false password entering.Hence if anybody tries to break in unauthorized, these snapshots would be useful for issuing legal against the intruder.Our script also organizes the snapshots on days basis on different folder.So it will be helpful for the end user to locate any old event that occurred in a particular day.

5.6 Sustainability Evaluation (PO(g))

During development we tried to use modules that utilizes maximum efficiency and minimizes power consumption. Hence we opted for Esp32 instead of esp-8266 .Although esp-8266 is a cheaper alternative but esp-32 have much higher power efficiency and much better performance for maintaining real time applications.

5.7 Ethical Issues (PO(h))

Our outdoor unit consists of a camera module that takes snapshots whenever it senses multiple failure while entering password .And this vary feature may pose some ethical issue.One might argue that taking picture without consent might go against the law. Our research shows that in most areas it is completely legal to take snapshots of public area. And as our outdoor unit will face the outside wall it is more likely to be categorized as a snapshot of public place,hence withdrawing the concern of individuals consent.

6 Reflection on Individual and Team work (PO(i))

6.1 Individual Contribution of Each Member

Shadman Sobhan 109 - Worked with password protection system which will open the door after entering correct password, also show status in lcd display and also giving the user chance to modify the password .

Fahim Ahmed 110 - Worked with central decoding circuit design for sensors and implemented the circuit. Also designed a PCB for this circuit and wrote the code to connect the circuit to ESP32 to detect fire alarm/gas alarm and corresponding room number.

Kazi Abrar Mahmud 112 - Implemented OTA(over the air) wifi connecectivity to improve user experience and blynk® based ioT hub (for managing user information), designed detachable sensor hubs for improving project scalability.

Anik Biswas 125 -Developed a code to capture a picture of the possible intruder. Also implemented Base64 encryption algorithm which will encrypt the data while uploading it to the cloud. Also contributed in modifying the google script.

6.2 Mode of TeamWork

As all the group members are attached, we divided the whole project into various features

and everyone worked with respective features. After individual success, all codes were merged and modified in some parts. On the other hands, respective hardwares were also merged and pcb was designed. Then the IoT part was done where owner will be sent notification.

6.3 Diversity Statement of Team

All the team members had to work with both software and hardware. However mostly work was distributed based on individual interest.

6.4 Log Book of Project Implementation

Date	Milestone achieved	Individual Role	Team Role	Comments
10/12/2023	Project assigned	-	-	-
14/12/2023	Work distribution	-	All members decided	
23/12/2023	Components listed and bought	-	Everyone went for shopping	All components bought for shopping
27/12/2023	Central decoding circuit design simulation	110	-	successful
04/01/2023	OTA wifi connectivity	112	-	
06/01/2023	Decoding circuit build in hardware	110		Successfully detects the fault room number
09/01/2024	Password checking for ESP-32	109	-	
16/01/2024	ESP-32 can detect fire and gas with room numbers	110	-	One feature was completed
18/01/2024	Intruder detection after wrong password	125	-	
25/01/2024	Blynk app	112	-	Notification to user could be sent
26/01/2024	PCB design	110		
28/01/2024	Respective activity for correct and wrong password	109	-	One feature was completed
02/02/2024	Encryption algorithm	125	-	Data security obtained
05/02/2024	Remaining parts discussed	-	Meeting for all members	Upcoming work and plan decided

10/02/2024	User password modification and LCD display	109	-	New feature included
15/02/2024	All code was merged	112	Meeting for all group members	Software/Coding part completed
25/02/2024	PCB soldering done	112	-	
01/03/2024	Final checking	-	Everyone was present to check whether everything is okay	Project completed

7 Communication to External Stakeholders (PO(j))

7.1 Executive Summary

The Smart Home security System can be thought as a more digital approach of the home security. We know There have been many instances where a small leakage of causes a huge fire breakout moreover there also has been instances where a huge causality can be avoided if there was a method of warning. Well, our project just deals with all the above discussed problems also adds more security features. To give a bit of description on this about the project we can point to the fact that there is a password lock of the door. Also to prevent and identify intruders there are is an implantation of a security camera which triggers and clicks a photo while a misdetection of the password. This is simultaneous as it shares the same LAN as the house it is implemented.

7.2 GitHub Link

Encryption repository: https://github.com/B-Con/crypto-algorithms/tree/master

8 Project Management and Cost Analysis (PO(k))

8.1 Bill of Materials

Equipment	Price(taka)
Esp32,ESP cam	2050
Servo Motor	250
Relay	88
Solenoidal lock	750
LCD 12c display	240
Fire Sensor	240

Gas Sensor	280
Keypad	100
PCB	1000
Jumper and Bread board	600
Total	5598 Taka

8.2 Calculation of Per Unit Cost of Prototype

While considering per unit cost we have keep in mind we don't have to use the materials for the initial scratch model again and again we can use the finalized PCB module and the related sensor and ESP32 camera and module to make the per unit cost . So, in a unit of our smart home security as product does not include the cost of jumper wire and Bread board. So the Estimated cost of the per unit is 5598-600 = 4998 Taka.
So the per unit cost of the prototype is 4098 taka

8.3 Calculation of Per Unit Cost of Mass-Produced Unit

When we consider mass production the price will reduce a way more as so far the components we calculated price for was based on the individual pricing of the product. When we want to make a mass production we will have to buy a lot of the equipment this can reduce our expenses way more . From our search if we but 100 pieces of ESP modules from AliExpress the is roughly 3usd which is around 400taka which we had to buy for 850 taka. So we can see there is almost 50% price gap in the mass production and creating a single unit. Considering the deviation of the price gap across all the component we came to conclusion that we could save around 35%-37% if we want to create a mass production of the production.
In that case the per unit cost is (5098-5098x0.35) = 3313.7 taka .

8.4 Timeline of Project Implementation

Date	Contribution
27 December	Central Decoding circuit build
16 January	Completed Sensor Implementation
25 January	Notification to user through Blynk app
02 February	Data security was obtained
15 February	Software/ coding part completed
25 February	PCB design and soldering
1 March	Project completed

9 Future Work (PO(l))

Enhanced Security: Implement the encryption algorithm to the password as well.

Improved reliability: Want to trigger the camera activation after a several time of mismatch of password . This way the actual owner is not detected as an intruder.

Compact design: Optimize the overall design for a smaller footprint which would increase it's appeal and also will be easy to install

Seamless integration: Refine the connections between working blocks to ensure smooth operation and reduce potential connection issues.

10 Reference

- https://esp32io.com/tutorials/esp32-servo-motor
- https://esp32io.com/tutorials/esp32-lcd
- https://esp32io.com/tutorials/esp32-door-lock-system-using-password#google_vignette
- https://www.aranacorp.com/en/using-the-eeprom-with-the-esp8266/
- 74191 Datasheet, PDF - Alldatasheet
- SN74153 Datasheet, PDF - Alldatasheet
- CD4047 Datasheet, PDF - Alldatasheet
- A132002_Flame sensor Module LM393_Datasheet.pdf (rajguruelectronics.com)
- MQ2.pdf (pololu.com)