#include <LiquidCrystal.h>

#include <EEPROM.h>

#include <SoftwareSerial.h>  //Create software serial object to communicate with SIM800L

SoftwareSerial GSM(11, 12);    //SIM800L Tx is 11 & Rx is 12 connected to Arduino

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

char phone\_no[] = "+84973915035";  //change +84 with Vietnam country code and 973915035 with phone number to sms

#define FlameSensorPin 8  // choose the input pin for Flame sensor

#define GasSensorPin 2    // choose the input pin for Gas sensor

#define TempsensorPin 3   // choose the input pin for Temperature sensor

#define buzzer 6          // choose the pin for the Buzzer

#define led\_C 9           // choose the pin for the Green LED Call/Message active indication

#define led\_M 10          // choose the pin for the Yellow LED Message active indication

#define led\_F 5           // choose the pin for the Red LED Fire detection indication

int read\_value;           // variable for reading the sensorpin status

int sms\_Status, call\_Status;

int flag = 0;

int var\_1 = 0, var\_2 = 0;

char input\_string[15];  // will hold the incoming character from the GSM field

int val\_gas = 0;

int val\_temp = 0; int temp\_range = 0;

int val\_flam = 0;

int temp = 0, i = 0;

char str[15];

void setup() {

  Serial.begin(9600);              // Begin Serial communication Arduino and Arduino IDE (Serial Monitor)

  GSM.begin(9600);                 // Begin serial communication with Arduino and SIM800L

  lcd.init();                      // initialize the lcd

  lcd.backlight();                 // print a message to the LCS

  pinMode(GasSensorPin, INPUT);    // GAS sensor

  pinMode(TempsensorPin, INPUT);   // Temperature Sensor

  pinMode(led\_F, OUTPUT);          // Declare Red LED as output

  pinMode(led\_C, OUTPUT);          // declare Green LED as output

  pinMode(led\_M, OUTPUT);          // declare Yellow LED as output

  pinMode(buzzer, OUTPUT);         // Buzzer

  pinMode(FlameSensorPin, INPUT);  // FLame Sensor

  Serial.println("Initializing...");

  delay(1000);

  initModule("AT", "OK", 1000);                 //Once the handshake test is successful, it will back to OK

  initModule("ATE1", "OK", 1000);               //this command is used for enabling echo

  initModule("AT+CPIN?", "READY", 1000);        //this command is used to check whether SIM card is inserted in GSM Module or not

  initModule("AT+CMGF=1", "OK", 1000);          //Configuring TEXT mode

  initModule("AT+CNMI=2,2,0,0,0", "OK", 1000);  //Decides how newly arrived SMS messages should be handled

  Serial.println("Initialized Successfully");

  sendSMS(phone\_no, "Start GSM Fire Alert System");

  sms\_Status = EEPROM.read(1);

  call\_Status = EEPROM.read(2);

  lcd.setCursor(0, 0);  // (col, row): move to new position, the text written to the LCS will be displayed from 0,0

  lcd.print(" GSM Base Fire     ");

  lcd.setCursor(0, 1);

  lcd.print(" Security System ");

  delay(2000);

  lcd.clear();

  Serial.println("AT+CNMI=2,2,0,0,0");  // message indication to TE

  delay(500);

  Serial.println("AT+CMGF=1");  // select message format

  delay(1000);

}

void loop() {

  updateSerial();

  if(temp==1)

  {

   check();

   temp=0;

   i=0;

   delay(1000);

  }

  val\_gas =  digitalRead(2); // gas sensor output pin connected

  val\_temp = analogRead(A0); // temperature sensor output pin connected

  val\_flam = digitalRead(8); // flame sensor output pin connected

  Serial.println(val\_gas); // see the value in serial monitor in Arduino IDE

  Serial.println(val\_temp); // see the value in serial monitor in Arduino IDE

  Serial.println(val\_flam); // see the value in serial monitor in Arduino IDE

  delay(100);

  /\* calculate the value (Volt) from temp sensor, lm35 range (-55,150) -> 55+150 = 205

  205\*5 = 1025

  \*/

  float temp\_range = val\_temp\*5.0/1024.0;

  if(val\_gas == 0 || temp\_range == 1 || val\_flam == 0)  // fire

  {

   Serial.print("\r");

   delay(1000);

   Serial.print("AT+CMGF=1\r");

   digitalWrite( led\_F, HIGH); // led

   digitalWrite( buzzer, HIGH); // Buzzer

   lcd.setCursor(0,0);

   lcd.print(" Fire Detected     ");

   lcd.setCursor(0,1);

   lcd.print("   Be Safe    ");

   if(flag == 0 )

   {

     flag = 1;

     if (sms\_Status == 1)

     {

       sendSMS(phone\_no, "Fire is Detected Alert.....!!!");

     }

     delay(1000);

     if (call\_Status == 1)

     {

       callUp(phone\_no);

     }

   }

    delay(1000);

    /\*Replace XXXXXXXXXX to 10 digit mobile number & ZZ to 2 digit country code\*/

    Serial.print("AT+CMGS=\"+84973915035\"\r");

    delay(1000);

    //The text of the message to be sent.

    Serial.print("Fire Alert");

    delay(1000);

    Serial.write(0x1A);

  }

  else  // not fire

  {

     digitalWrite( led\_F,LOW); // led

     digitalWrite( buzzer,LOW); // Buzzer

     lcd.setCursor(0,0);

     lcd.print("  FIRE NOT    ");

     lcd.setCursor(0,1);

     lcd.print("  DETECTED    ");

     digitalWrite(buzzer, LOW);

     digitalWrite(led\_F, LOW);

     flag = 0;

  }

  if (var\_1 == 1)

  {

    if (!(strncmp(input\_string, "Sms On", 6)))

    {

      sms\_Status = 1;

      EEPROM.write(1, sms\_Status);

      sendSMS(phone\_no, "Message is Active");

    } else

    if (!(strncmp(input\_string, "Sms Off", 7)))

    {

      sms\_Status = 0;

      EEPROM.write(1, sms\_Status);

      sendSMS(phone\_no, "Message is Deactivate");

    } else

    if (!(strncmp(input\_string, "Call On", 7)))

    {

      call\_Status = 1;

      EEPROM.write(2, call\_Status);

      sendSMS(phone\_no, "Call is Active");

    } else

    if (!(strncmp(input\_string, "Call Off", 8)))

    {

      call\_Status = 0;

      EEPROM.write(2, call\_Status);

      sendSMS(phone\_no, "Call is Deactivate");

    }

    var\_1 = 0;

    var\_2 = 0;

  }

  digitalWrite(led\_M, sms\_Status); //LED On SMS Active, LED Off SMS Deactivate

  digitalWrite(led\_C, sms\_Status); //LED On Call Active, LED Off Call Deactivate

  updateSerial(); // for debug

}

void serialEvent() {

}

void check() {

}

void sendSMS(char \*number, char \*msg) {

  GSM.print("AT+CMGS=\"");

  GSM.print(number);

  GSM.println("\"\r\n");

  //AT+CMGS=”Mobile Number” <ENTER> - Assigning recipient’s mobile number

  delay(500);

  GSM.println(msg);  // Message contents

  delay(500);

  GSM.write(byte(26));  //Ctrl+Z  send message command (26 in decimal).

  delay(5000);

}

void callUp(char \*number) {

  GSM.print("ATD +");

  GSM.print(number);  // call to the specific number

  GSM.println(";");   // ends with semi-colon

  delay(1000);

}

void readSMS() {

  while (GSM.available() > 0) {

    //------------ PIN HERE --------------//

    if (GSM.find("/786"))  //'/786' where 786 is 3 digit PIN.

    //--------- PIN HERE ---------//

    {

      delay(1000);

      while (GSM.available()) {

        char input\_char = GSM.read();

        input\_string[var\_2++] = input\_char;

        if (input\_char == '/') {

          var\_1 = 1;

          return;

        }

      }

    }

  }

}

void initModule(String cmd, char \*res, int t) {

  while(1)

  {

  Serial.println(cmd);

  GSM.println(cmd);

  delay(1000);

  // Serial.println(GSM.available());

  while (GSM.available()) {

    Serial.write(GSM.read());

    if(GSM.find(res))

    {

      Serial.println(res);

      delay(t);

      return;

    } else {

      Serial.println("Error");

      // Serial.println(GSM.read());

    }

  }

  // while (Serial.available()) {

  //   GSM.write(Serial.read());  //Forward what Serial received to Software Serial Port

  // }

  delay(t);

  }

}

void updateSerial() {

  delay(500);

  while (Serial.available())

  {

    GSM.write(Serial.read());//Forward what Serial received to Software Serial Port

  }

  while(GSM.available())

  {

    Serial.write(GSM.read());//Forward what Software Serial received to Serial Port

  }

}