Smart Home

```
// Smart Home Practical

/* Includes Section */
#include <Wire.h>
#include <ThreeWire.h>
#include <RtcDS1302.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>

/* Variables Section */
#define RTC_CLOCK 2
#define RTC_DATA 3
#define RTC_RESET 4
#define KP_R1 5
#define KP_R2 6
#define KP_R3 7
#define KP_R4 8
#define KP_C1 9
#define KP_C2 10
#define KP_C3 11
#define KP_C4 12
#define ALARM A3
#define DOOR A2
#define LDR_OUT A1
#define OUTDOOR_LIGHT A0

// Number of pressed keys
int entered_Num = 5;

// LDR Variable
bool LDR_Status = 0;

// ALARM
bool ALARM_OFF_STATUS = 0;

// Password variables
String pad;
char keyPressed;
int pass_Flag = 0;

// Date and Time variables
const long Event_Time = 30000;
unsigned long Previous_Time = 0;

// DOOR Timer
unsigned long time_for_action;
#define OUTDOORINTERVAL 5000

// Keypad variables
const byte ROWS = 4; // Set up four rows for the Keypad
const byte COLS = 4; // Set up four coloumns for the Keypad

// Set up the modules
LiquidCrystal_I2C lcd(0x27, 16, 2);   // Setting the connection with the I2C #I2C Address: 0x27
ThreeWire myWire(RTC_DATA,RTC_CLOCK,RTC_RESET);
RtcDS1302<ThreeWire> Rtc(myWire);

char Keys[ROWS][COLS] =
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};


```

```
// Address the pins of the Keypad
byte rowPins[ROWS] = {KP_R1,KP_R2,KP_R3,KP_R4};
byte colPins[COLS] = {KP_C1,KP_C2,KP_C3,KP_C4};

Keypad customKeypad = Keypad(makeKeymap(Keys), rowPins, colPins, ROWS, COLS);

void setup()
{
  pinMode(LDR_OUT, INPUT);
  pinMode(ALARM, OUTPUT);
  pinMode(ACCESS_GRANTED, OUTPUT);
  pinMode(OUTDOOR_LIGHT, OUTPUT);
  pinMode(DOOR, OUTPUT);

  digitalWrite(ALARM, LOW);
  digitalWrite(DOOR, HIGH);

  // Starting Serial connection
  Serial.begin(115200);
  int myTimeout = 100; // milliseconds for Serial.readString
  Serial.setTimeout(myTimeout);

  // Starting the LCD
  lcd.init();
  lcd.backlight();
  lcd.clear();

  // Starting the Real Time Clock
  Rtc.Begin();
  RtcDateTime currentTime = RtcDateTime(__DATE__ , __TIME__);
  Rtc.SetDateTime(currentTime);
}

void loop()
{
  StartUP();
  //Set_Date_Time();
  String password = "123";    // Set up the password

  OUTDOOR();
  readKeypad();
  if(keyPressed == '#')
  {
    if(pad == password)
    {
      ACCESS_GRANTED();
    }
    else
    {
      ACCESS_DENIED();
    }
  }
}

void Set_Date_Time()
{
  if(pad == 0)
  {
    RtcDateTime now = Rtc.GetDateTime();            // Get the current Date & Time

    unsigned long Current_Time = millis();

    if(Current_Time - Previous_Time >= Event_Time)  // Timer to change every one sec
    {
      lcd.setCursor(0,0);
      lcd.print("Date: ");
      lcd.print(now.Day());
      lcd.print("/");
      lcd.print(now.Month());
      lcd.print("/");
      lcd.print(now.Year());
```

```
      lcd.setCursor(0,1);
      lcd.print("Time: ");
      lcd.print(now.Hour());
      lcd.print(":");
      lcd.print(now.Minute());
      lcd.print(":");
      lcd.print(now.Second());
    }
  }
}

void OUTDOOR()
{
  LDR_Status = digitalRead(LDR_OUT);

  String Serial_Read;
  Serial_Read = Serial.readStringUntil('\n');

  if(1 == LDR_Status)
  {
    digitalWrite(OUTDOOR_LIGHT, LOW);
  }
  else
  {
    digitalWrite(OUTDOOR_LIGHT, HIGH);
  }
}

void readKeypad()
{
  keyPressed = customKeypad.getKey();
  if (keyPressed != '#')
  {
    String typed_Pass = String(keyPressed);
    pad += typed_Pass;
    lcd.setCursor(0, 0);
    lcd.print("Taking password:");
    if(keyPressed)
    {
      lcd.setCursor(entered_Num++, 1);
      lcd.print("#");
    }
  }
}

void ACCESS_GRANTED()
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Correct Password");
  lcd.setCursor(0,1);
  lcd.print("Access Granted:)");

  String Serial_Read;

  Serial_Read = Serial.readStringUntil('\n');
  Serial.println(Serial_Read);

  if(Serial_Read[0] == '0')
  {
    StartUP();
  }

  digitalWrite(DOOR, LOW);
  delay(1000);
  digitalWrite(DOOR, HIGH);

  delay(5000);

  pad = "";

  lcd.clear();
```

```
  pass_Flag = 0;
  entered_Num = 5;
}

void ACCESS_DENIED()
{
  if(2 != pass_Flag)
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Incorrect Pass ");
    lcd.setCursor(0,1);
    lcd.print("Access Denied!!");

    pad = "";

    delay(5000);
    lcd.clear();
    entered_Num = 5;
    pass_Flag++;
  }
  else
  {
    lcd.clear();
    while(pass_Flag != 0)
    {
      lcd.setCursor(0,0);
      lcd.print("    ALERT!!!");
      lcd.setCursor(0,1);
      lcd.print("  Home Locked!  ");
      digitalWrite(ALARM, HIGH);
      delay(500);
      digitalWrite(ALARM, LOW);
      delay(500);
      StartUP();
    }
    lcd.clear();
    entered_Num = 5;
  }
}

void TURN_OFF_ALARM()
{
  String Serial_Read;
  Serial_Read = Serial.readStringUntil('\n');

  if(Serial_Read[0] == '3')
  {
    pass_Flag == 0;
  }
}

void Fire_Alert()
{
  lcd.setCursor(0,0);
  lcd.print("  FIRE ALERT!!  ");
  lcd.setCursor(0,1);
  lcd.print("Flame Detected!!");
  digitalWrite(ALARM, HIGH);
  delay(500);
  digitalWrite(ALARM, LOW);
  delay(500);
}

void StartUP()
{
  while(Serial.available() > 0)
  {
    String Serial_Read;
    long message;

    Serial_Read = Serial.readStringUntil('\n');

```

```
    Serial.println(Serial_Read);

    if(Serial_Read[0] == '2')
    {
      Fire_Alert();
    }
    else if(Serial_Read[0] == '1')
    {
      ACCESS_GRANTED();
    }
    else if(Serial_Read[0] == '0')
    {
      digitalWrite(DOOR, HIGH);
    }
    else if(Serial_Read[0] == '3')
    {
      pass_Flag = 0;
    }
    else if(Serial_Read[0] == '4')
    {
      ACCESS_DENIED();
    }
    else if(Serial_Read[0] == '7')
    {
      digitalWrite(OUTDOOR_LIGHT, LOW);
      delay(2000);
    }
    else if(Serial_Read[0] == '8')
    {
      digitalWrite(OUTDOOR_LIGHT, HIGH);
      delay(2000);
    }
    lcd.clear();
    break;
  }
}
```
