

a- Temperature warning system

Use the LM35 to measure the current air temperature and print the readings in Centigrade (°C) on the LCD screen. If the temperature exceeds a certain limit (30°C for example), a buzzer should turn on and the LCD should print a warning message 'WARNING!! High Temperature'.

NOTE: In order for you to increase the surrounding temperature use any heating device that doesn't use flames. (Hairdresser for example or any similar device).

IMPORTANT: If you are using the LCD shield then the correct codes are available in this link:

[https://www.dfrobot.com/wiki/index.php/Arduino_LCD_KeyPad_Shield_\(SKU:_DFR0009\)](https://www.dfrobot.com/wiki/index.php/Arduino_LCD_KeyPad_Shield_(SKU:_DFR0009))

Implementation:

```

1 #include <IRremote.h>
2 #include <NewTone.h>
3 #include <LiquidCrystal.h>
4
5
6 #define recPin 6
7 #define buzzerPin 7
8 #define ledPin 13
9
10 IRrecv irrecv(recPin);
11 decode_results results;
12
13
14 unsigned long lastPrintTime = 0;
15 String threshold = "";
16 String sensorThreshold = "0";
17 int remoteInputLength = 0;
18
19
20
21 // initialize the library by associating any needed LCD interface pin
22 // with the arduino pin number it is connected to
23 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
24 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
25
26
27
28 void setup() {
29   pinMode(ledPin, OUTPUT);
30   pinMode(recPin, INPUT);
31   pinMode(buzzerPin, OUTPUT);
32   Serial.begin(9600);
33   irrecv.enableIRIn();
34
35   // set up the LCD's number of columns and rows:
36   lcd.begin(16, 2);
37   // Print a message to the LCD.
38 }

```

Here we initialize some I/O pins:

Receiver pin → INPUT → digital pin 6

RS → digital pin 12, ENABLE → digital pin 11

Buzzer pin → OUTPUT → digital pin 7

LCD DATA pins → digital pins 5, 4, 3, 2

```

void loop() {
    float analogValue = analogRead(A0);
    float mv = (analogValue / 1023) * 5000;
    float celsius = mv / 10;

    if (irrecv.decode(&results)) {
        long remoteInVal = results.value;

        irrecv.resume();

        switch (remoteInVal) {
            case 33444015 : threshold.concat("1"); remoteInputLength ++; break;
            case 33478695 : threshold.concat("2"); remoteInputLength ++; break;
            case 33486855 : threshold.concat("3"); remoteInputLength ++; break;
            case 33435855 : threshold.concat("4"); remoteInputLength ++; break;
            case 33468495 : threshold.concat("5"); remoteInputLength ++; break;
            case 33452175 : threshold.concat("6"); remoteInputLength ++; break;
            case 33423615 : threshold.concat("7"); remoteInputLength ++; break;
            case 33484815 : threshold.concat("8"); remoteInputLength ++; break;
            case 33462375 : threshold.concat("9"); remoteInputLength ++; break;
            case 33480735 : threshold.concat("0"); remoteInputLength ++; break;
            case 33431775 : threshold.remove(remoteInputLength - 1); remoteInputLength --; break;
            case 33441975 :
                sensorThreshold = threshold;
                threshold = "";
                break;
            default : break;
        }
    }
}

```

Code to Map Values read from the LM35 temperature sensor to Celsius degrees.

Code to read raw values from the Remote through IR Receiver.

```

if (celsius > sensorThreshold.toInt()) {
    lcd.clear();
    digitalWrite(ledPin, HIGH);
    NewTone(buzzerPin, 3000);
    lcd.setCursor(0, 0);
    lcd.print("Warning high temp");
    lcd.setCursor(0, 1);
    lcd.print(celsius);
    lcd.print(" ");
    lcd.print(sensorThreshold.toInt());
    lcd.print(" ");
    lcd.print(threshold);

    Serial.println(celsius);
}
else {
    lcd.clear();
    digitalWrite(ledPin, LOW);
    noNewTone(buzzerPin);
    Serial.println(celsius);
    lcd.setCursor(0, 0);
    lcd.print("Max Temp: ");
    lcd.print(sensorThreshold.toInt());
    lcd.setCursor(0, 1);
    lcd.print(celsius);
    lcd.print(" ");
    lcd.print(threshold);
}

```

Code to display on the LCD screen the Warning Statement provided that the Reading from the Sensor is Greater than the Maximum Threshold.