**NAME :Dharsini S**

**Tittle :Temperature alarm by using arduino**

#include "TTSDisplay.h"

#include "TTSTemp.h"

TTSTemp temp;

TTSDisplay rtcshield;

boolean alarmOnOff = false;

int highTemp = 40;

int lowTemp = 10;

int currentTemp;

void LEDsoff()

{

// function to turn all alarm high/low LEDs off

digitalWrite(2, LOW);

digitalWrite(4, LOW);

}

void setup() {

// initalise digital pins for LEDs and buzzer as outputs

pinMode(2, OUTPUT); // LED 1

pinMode(3, OUTPUT); // LED 2

pinMode(4, OUTPUT); // LED 3

pinMode(5, OUTPUT); // LED 4

pinMode(6, OUTPUT); // buzzer

// initalise digital pins for buttons as inputs

// and initialise internal pullups

pinMode(9, INPUT\_PULLUP); // button K1

pinMode(10, INPUT\_PULLUP); // button K2

pinMode(11, INPUT\_PULLUP); // button K3

}

void loop()

{

// get current temperature

currentTemp = temp.get();

// if current temperature is within set limts

// show temperature on display

if (currentTemp >= lowTemp || currentTemp <= highTemp)

// if ambient temperature is less than high boundary

// OR if ambient temperature is grater than low boundary

// all is well

{

LEDsoff(); // turn off LEDs

rtcshield.num(currentTemp);

}

// if current temperature is above set high bounday, show red LED and

// show temperature on display

// turn on buzzer if alarm is set to on (button K3)

if (currentTemp > highTemp)

{

LEDsoff(); // turn off LEDs

digitalWrite(4, HIGH); // turn on red LED

rtcshield.num(currentTemp);

if (alarmOnOff == true) {

digitalWrite(6, HIGH); // buzzer on }

}

}

// if current temperature is below set lower boundary, show blue LED and

// show temperature on display

// turn on buzzer if alarm is set to on (button K3)

if (currentTemp < lowTemp)

{

LEDsoff(); // turn off LEDs

digitalWrite(2, HIGH); // turn on blue LED

rtcshield.num(currentTemp);

if (alarmOnOff == true)

{

digitalWrite(6, HIGH); // buzzer on }

}

}

// --------turn alarm on or off-----------------------------------------------------

if (digitalRead(11) == LOW) // turn alarm on or off

{

alarmOnOff = !alarmOnOff;

if (alarmOnOff == 0) {

digitalWrite(6, LOW); // turn off buzzer

digitalWrite(5, LOW); // turn off alarm on LED

}

// if alarm is set to on, turn LED on to indicate this

if (alarmOnOff == 1)

{

digitalWrite(5, HIGH);

}

delay(300); // software debounce

}

// --------set low temperature------------------------------------------------------

if (digitalRead(10) == LOW) // set low temperature. If temp falls below this value, activate alarm

{

// clear display and turn on blue LED to indicate user is setting lower boundary

rtcshield.clear();

digitalWrite(2, HIGH); // turn on blue LED

rtcshield.num(lowTemp);

// user can press buttons K2 and K1 to decrease/increase lower boundary.

// once user presses button K3, lower boundary is locked in and unit goes

// back to normal state

while (digitalRead(11) != LOW)

// repeat the following code until the user presses button K3

{

if (digitalRead(10) == LOW) // if button K2 pressed

{

--lowTemp; // subtract one from lower boundary

// display new value. If this falls below zero, won't display. You can add checks for this yourself :)

rtcshield.num(lowTemp);

}

if (digitalRead(9) == LOW) // if button K3 pressed

{

lowTemp++; // add one to lower boundary

// display new value. If this exceeds 9999, won't display. You can add checks for this yourself :)

rtcshield.num(lowTemp);

}

delay(300); // for switch debounce

}

digitalWrite(2, LOW); // turn off blue LED

}

// --------set high temperature-----------------------------------------------------

if (digitalRead(9) == LOW) // set high temperature. If temp exceeds this value, activate alarm

{

// clear display and turn on red LED to indicate user is setting lower boundary

rtcshield.clear();

digitalWrite(4, HIGH); // turn on red LED

rtcshield.num(highTemp);

// user can press buttons K2 and K1 to decrease/increase upper boundary.

// once user presses button K3, upper boundary is locked in and unit goes

// back to normal state

while (digitalRead(11) != LOW)

// repeat the following code until the user presses button K3

{

if (digitalRead(10) == LOW) // if button K2 pressed

{

--highTemp; // subtract one from upper boundary

// display new value. If this falls below zero, won't display. You can add checks for this yourself :)

rtcshield.num(highTemp);

}

if (digitalRead(9) == LOW) // if button K3 pressed

{

highTemp++; // add one to upper boundary

// display new value. If this exceeds 9999, won't display. You can add checks for this yourself :)

rtcshield.num(highTemp);

}

delay(300); // for switch debounce

}

digitalWrite(4, LOW); // turn off red LED

}

}