NEW HORIZON
COLLEGE OF ENGINEERING
New Horizon Knowledge Park, Ring Road, Marathalli
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

Project Report

Title:

SMART THERMAL HYGROMTER

Prepared by:

Mohammed Omer Ali 1NH18EE031
Sahana B 1NH18EE050
Siddhartha Sunil Singh 1NH18EE057
Rachana R 1NH18EE044

Guided By:

Mr. Vinod Kumar S

Sr. Asst. Professor, Dept. of Electrical and

Electronics Engineering, NHCE, Bengaluru.

NEW HORIZON
COLLEGE OF ENGINEERING
New Horizon Knowledge Park, Ring Road, Marathalli
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

CERTIFICATE

Certified that the Mini Project work entitled "**Wireless AC Power Control Circuit**" carried out by "M.D. Omer Ali 1NH18EE031", "Siddhartha Sunil Singh 1NH18EE057", "Sahana 1NH18EE050" students of New Horizon College of Engineering submitted the report in completion of project at Department of Electrical and Electronics Engineering, New Horizon College of Engineering during the Academic Year 2019-20. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for said Degree.

Project Guide	**HOD-EEE**
Mr. Vinod Kumar S	**Dr. Mahesh M**

ABSTRACT

Temperature can be defined as the measure of warmth or coldness of an object or substance with reference to some standard value. In the similar way humidity can be defined as the measure of amount of water vapor present in air.

To measure both of these a device called **Smart Thermal Hygrometer** is developed to perform thermal screening and hence both temperature and humidity can be measured without having physical contact.

The necessity of designing thermal hygrometer with thermal screening is that in few cases like measuring temperature of certain machines like DC motor it is not possible to have physical contact, and also nowadays Corona has led to the circumstances that the temperature has to be measured with no physical contact.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of ant task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to Dr. Manjunath, Principle NHCE, for his constant support and encouragement.

I am grateful to **Dr. Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. Mahesh M**, Professor and Head, Department of Electrical & Electronics Engineering, for his constant support.

I express my gratitude to **Mr. Vinod Kumar S**, Sr. Asst. Professor, my project guide, for constantly monitoring the development of the project and setting up precise deadlines.

His valuable suggestions were the motivating factors in completing the work**.**

Table of Contents:

CHAPTER 1:
INTRODUCTION

The Thermal Hygrometer is a device used to measure the temperature and humidity of a substance or the surrounding environment using the technique of thermal screening. Thermal screening is a process where the temperature is being scanned with the amount of moisture present on the surface of the particular substance.

On an average use, determining dew temperature is not really necessary, however if for example some paint job has to be done, determining the dew temperature is necessarily important. The surface temperature will be some 3^{o}C above dew temperature to avoid the formation of a moisture film between the coating and the base that could compromise the quality of the paint job. For this, in a single instrument it is possible to sense the surface temperature contact less, and at the same time determine if the site conditions are appropriate for the job. Apart from determining dew temperature, the device can also display other parameters that the usual infrared thermometer cannot, such as relative humidity and apparent temperature.

CHAPTER 2:

LITERATURE SURVEY

1. Dario Camuffo- in microclimate for cultural Heritage (3rd edition)- 2019
2. D.K Roveti- Choosing a Humidity Sensor: a review of three technologies, sensors magazine- (2001)
3. Wexler, Arnold; Hyland- NBS standard Hygrometer, www.nist.gov. National Bureau of Standards- (2017)
4. Spectral Hygrometer- AMS Glossary, glossary.ametsoc.org. - (2019)

CHAPTER 3:

HARDWARE ARCHITECTURE AND IMPLEMENTATION

COMPONENTS:

- ESP8266 microcontroller
- OLED Display 128*64 pixels
- MLX90614 Sensor
- DHT11 Sensor
- 9V Battery
- Breadboard
- Connecting Wires
- Resistors
- LEDS

1. ESP8266 MICROCONTROLLER: An ESP8266 Microcontroller is a low cost Wi-Fi along with a full TCP/IP stack and microcontroller capability, produced by Espressif, Shanghai, China. It is highly affordable and hence is being used in this project. It has external flash memory up to 16MB supported and the operating temperature range is -45℃ to 125℃ and the current consumption ranges from 15µA – 400mA. The power supply required ranges from 2.5V - 3.6V

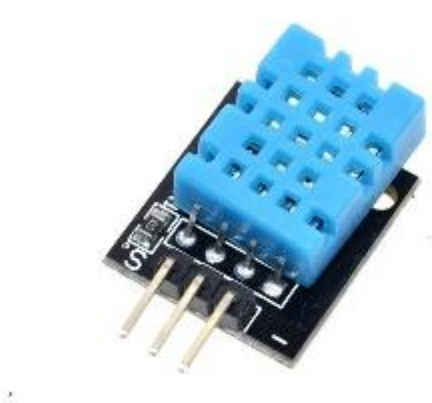
2. OLED Display 128*64 pixels: Organic Light Emitting Diodes is a flat light emitting technology, made by placing series of organic thin films in between two conductors. When the electrical current is applied, a bright light is emitted. OLEDs do not require a backlight and so are thinner and is more efficient than LCD displays which do require a white backlight.

3. MLX90614 Infrared Temperature Sensor: It is an infrared temperature sensor designed for non-contact temperature sensing with an internal 17-bit ADC and it has a powerful DSP contribute to the MLX90614's high accuracy and resolution. It has wide range of applications including body temperature measurement and movement detection. The MLX90614 provides two methods of output: PWM and SMBus (i.e. TWI, I2C). The 10-bit PWM output gives a resolution of 0.14°C, while the TWI interface has a resolution of 0.02°C. The MLX90614 is factory calibrated in wide temperature ranges: -40 to 85°C for the ambient temperature and -70 to 382.2°C for the object temperature. The measured

value is average temperature of all objects in the Field Of View of sensor. The MLX90614 offers a

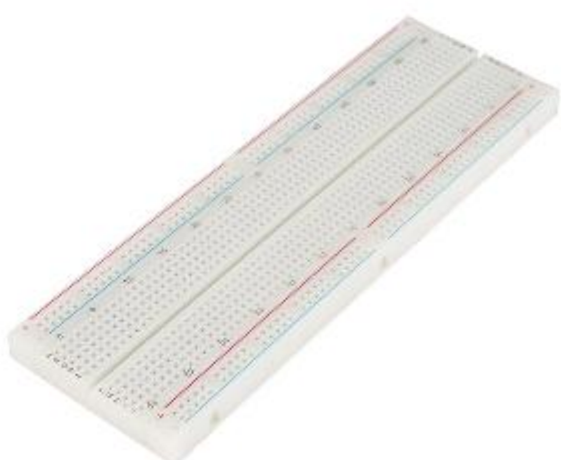
standard accuracy of 0.5°C around room temperatures.

4. DHT11 Temperature and Humidity Sensor: It is a basic, low cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermister to measure the surrounding air, and spits out a digital signal over the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. data will be updated from it once in every 2seconds.

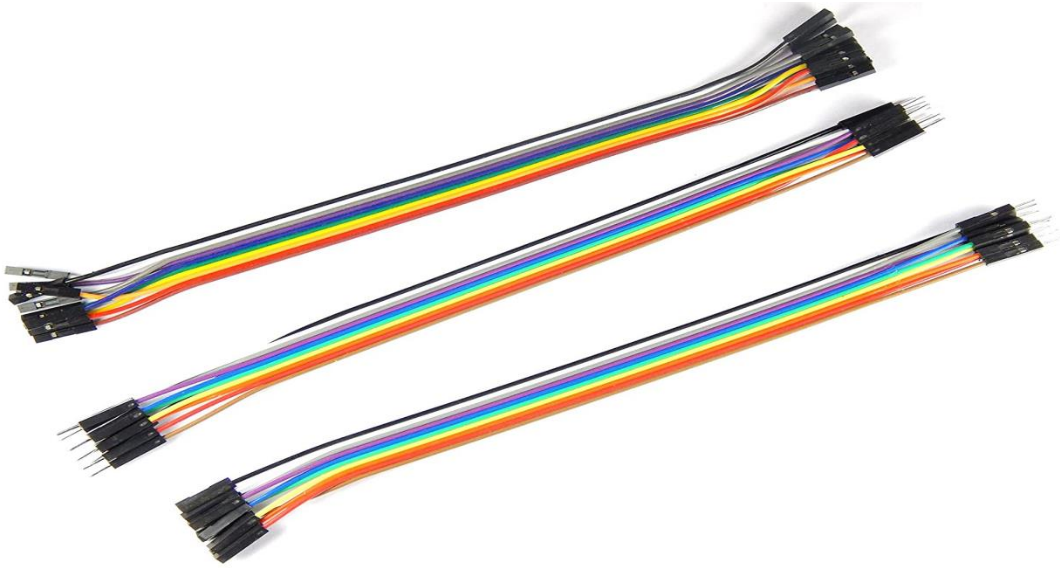


5. 9V Battery: It is a common size of battery which is used to power the circuit. It has a rectangular prism shape with rounded edges and a polarized snap connector at the top.

6. BREAD BOARD: It is a thin plastic board used to hold electronic components (transistors, resistors, chips, etc.) that are wired together. The breadboard is being used here to make connections of circuit.



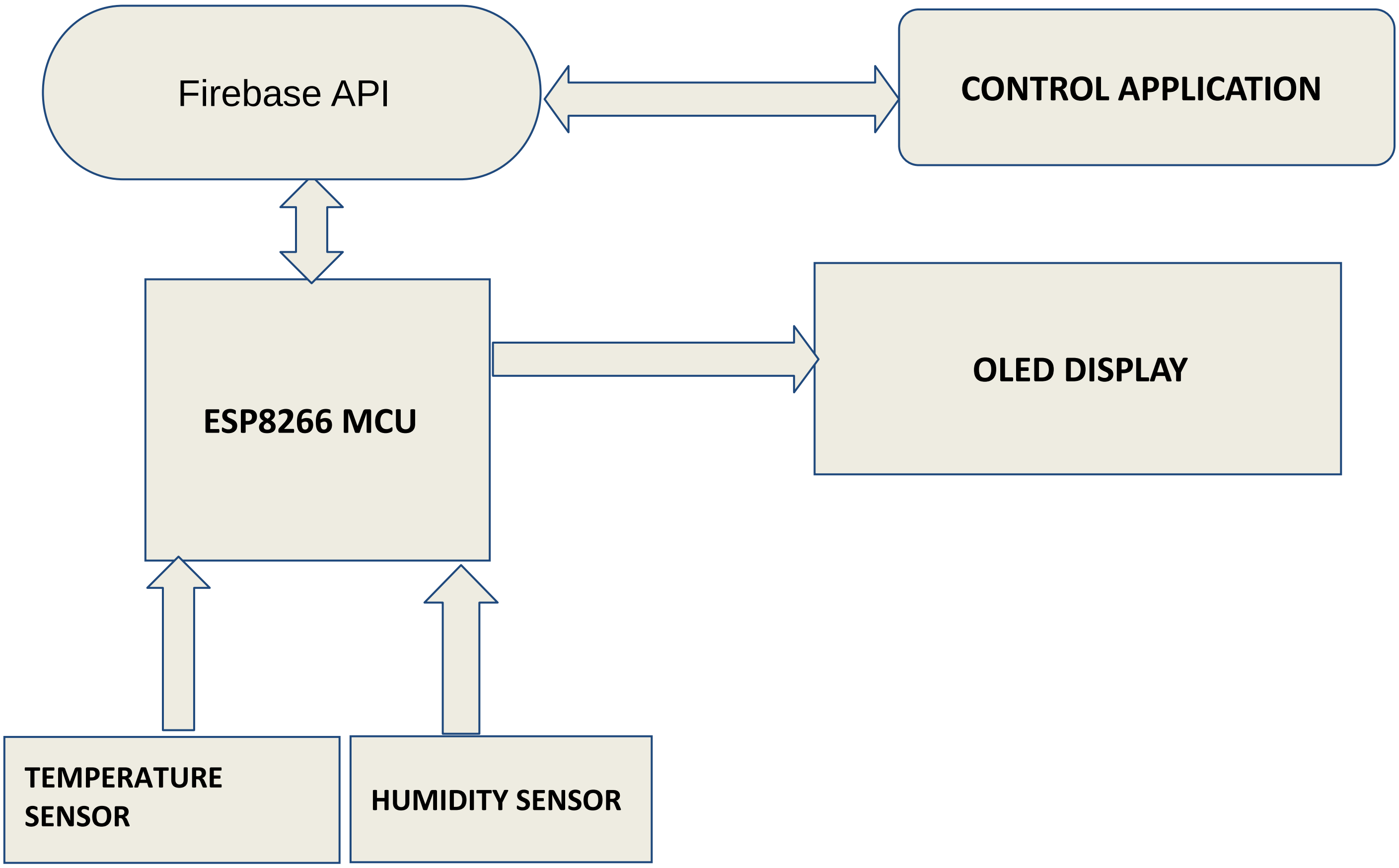
7. CONNECTING WIRES: Connecting wires provide a medium to the electrical current so that they can travel from one point on a circuit to another.



8. RESISTORS: It is a passive 2 terminal electrical component that implements electrical resistance as a circuit element. They are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.

9. LEDS: A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it.

BLOCK DIAGRAM



CHAPTER 4:

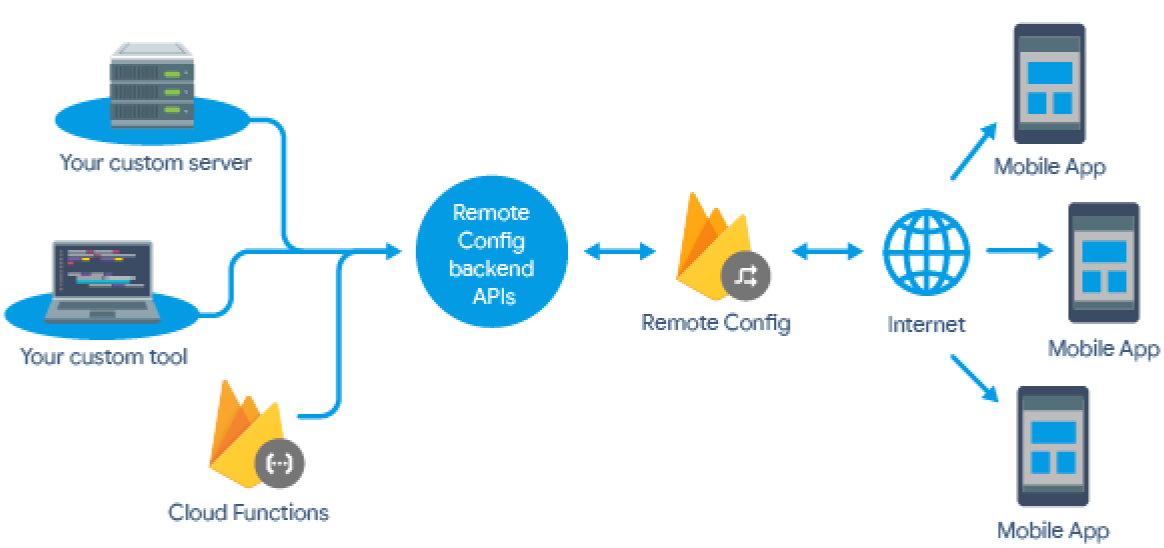
PROPOSED SYSTEM DESIGN

- In this project we are using the ESP8266 as the processing unit due to the Wi-Fi feature of the microcontroller, as we can transmit and receive data from other devices and it's also among one of the most affordable microcontrollers.
- The ESP8266 is firstly programmed to gather the input from both temperature sensor (MLX90614) and a humidity sensor(DHT11) which senses surrounding environment to measure the temperature and humidity, and then input given by the sensors will be processed by ESP8266 and then the processed output will be displayed on OLED display since ESP8266 is coupled with it.
- Both the sensors then send the data parallelly to the MCU which processes the data and then prints it to the OLED display showing the temperature.
- All the processed data displayed is then gathered and sent to another device using Firebase API for more detailed data analysation and the data can also be stored.

CHAPTER 5:

FIREBASE API SYSTEM

Firebase is an API that will allow developers easily sync and store data in real time. Developers can use the service to build their apps on their own without having to manage servers or write server-side code. There are clients for Android, iOS and JavaScript (including the bindings for Ember, React, Angular and Backbone). The API allows users to access the functionality of Firebase programmatically. The API supports standard CRUD operations over the data within the databases as well as for query of the data. Firebase makes use of WebSocket's to achieve realtime communication between clients. Data can also be accessed over the Firebase REST API. Firebase Simple Login is an additional service that enables the developers to authenticate users using only client-side code. Enable authentication via a number of third-party providers, anonymous login, or email / password authentication without having to manually store authentication credentials or to run a server. Firebase Hosting offers developers a fast, secure and reliable way to host their app's static assets such as HTML, CSS, JavaScript, and media files. It will be backed by a global CDN, serves content over SSL by default, and is available for either a custom domain or on a sub-domain of firebaseapp.com.



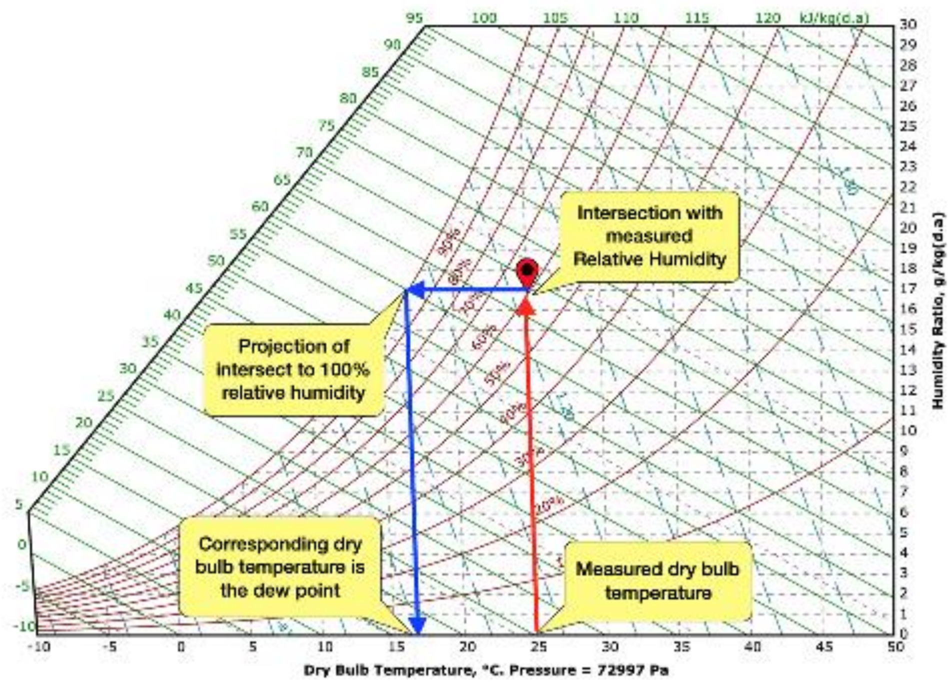
CHAPTER 6:
HARDWARE WORKING

Initially, all the components are assembled and their working condition is checked. Later on, the connections are made as per the circuit diagram.

The device is designed in such a way that the user can switch between two modes: object temperature mode, and the hygrometer mode. The device contains two push buttons. One of them will allow the user to switch between modes, while the other push button enables the infrared sensor. Only while the button is pressed will the device display the measured temperature. If the button is released the display will freeze into the last measurement taken. Adding on. whenever the device is in the object temperature mode and if its corresponding push button is pressed, the LED lights up. If the mode is switched, another LED lights up. In the hygrometer mode anyway, it is not necessary to keep any push button pressed. The device will update measurements every quarter of a second and the LED will remain on.

adding another feature to the apparatus is a battery level indicator. To achieve this, a voltage divider was added. From the circuit shown above, Z1 corresponds to a 2.7 kΩ resistor and Z2 corresponds to a 5 kΩ trimmer set at 2.98 kΩ. The voltage divider proportionally drops down the voltage of the 9V battery to a scale of 0-5V readable by the analog pins of the board. Within the code a scale is set up which will in turn draw a battery icon on the display, showing the level of the battery.

Psychrometric Theory for the Code: As mentioned before, one of the uses of this device is to calculate and display the dew temperature. This parameter can be expressed as a function of the dry bulb temperature and the relative humidity, which are determined by the MLX90614 and DHT11 sensors respectively. Even though the DHT11 can sense dry bulb temperature, the MLX90614 sensor offers a function for reading the ambient temperature in addition to the point source temperature, and is capable of achieving more accurate readings than the DHT11.



The image shown above corresponds to the Psychrometric chart of air at 2680 meters above sea level. This chart correlates thermodynamic properties of "wet air" which all of us experience regardless where we live. The chart shows ways to determine the dew point temperature where the dry bulb temperature and relative humidity values are given. For the case considered above, a reading is taken indicating a dry bulb temperature of 25ºC and a relative humidity of 60%. Finding intersection of this point with the 100% relative humidity curve along with constant humidity ratio and then projecting the intersection to dry bulb temperature scale indicates the dew temperature value. For this case approx. 16ºC. This means that for the given condition, any surface below 16ºC will get compressed into the moisture of air.

The above chart helps in understanding the relationship between the parameter. The equations that govern the thermodynamics of air makes it even simpler.

The equations required are:

Es = 6.108 * (e^((17.27 * T) / (237.3 + T)))

B = (ln(E / 6.108)) / 17.27

RH = 100 * (E / Es)

D = (237.3 * B) / (1 - B)

Where,

T = Air Temperature (Dry Bulb) in Centigrade (C) degrees

Es = Saturation Vapor Pressure at Dry Bulb (mb)

E = Actual Vapor Pressure (mb)

B = intermediate value (no units)

RH = Relative Humidity in percent (%)

D = Dewpoint in Centigrade (C) degrees.

Rearranging terms, with T and RH we can solve for D.

In addition to calculating dew temperature, the code with also include the equations for calculating the apparent temperature; although this parameter is a function of wind speed and solar radiation, since the apparatus is unable to sense these variables, only indoors apparent temperature is calculated as follows:

AT = Ta+ 0.348 * e

$$e = (rh)/100 * 6.105 * e^{\wedge}((17.27 * Ta) / (237.7+Ta))$$

Where.

T = Apparent Temperature

Ta = Dry Bulb Temperature

e = Water Vapor Pressure

RH = Relative Humidity in percent (%).

CHAPTER 7:
PROGRAM CODE OF ESP8266

sketch_dec22a | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Edit Sketch Tools Help

sketch_dec22a §

```
// ARDUINO PROGRAM FOR THERMO - HYGROMETER
//Using Arduino Nano

//Sensor library
#include <Adafruit_MLX90614.h> //GY-906 infrared temperature sensor
#include <DHT.h> //Thermo hygrometer sensor
#include <DHT_U.h> //Thermo hygrometer sensor
#include <Adafruit_SSD1306.h> //OLED 0.96" screen
#include <Adafruit_GFX.h>

//Additional libraries
#include <Wire.h>
#include <AsyncDelay.h>
#include <SoftWire.h>
#include <splash.h>
#include <SPI.h>
#include <gfxfont.h>

struct Temps{
  float tr;
  float ta;
};
int pinLedAzul = 10; //Output for blue LED
int pinLedBlanco = 12;//Output for white LED
int pinSDA = A4; //Input signal SDA
int pinSCL = A5; //Input signal SCL
int pinSwitch = 7; //Button for switching between infrared thermometer and y Thermo hygrometer
int pinTrigger = 5; //trigger button for infrared measurent
int pinDHT = 2; //Input signal KY-015
int pinNivBat = A1; //Input battery level voltage
int medicion = 1; //Wildcard for switching between infrared thermometer and y Thermo hygrometer
float r1 = 2.7; //Value for R1 in voltage divider [kOhm]
float r2 = 3.0; //Value for R2 in voltage divider [kOhm]
int temp = 0; //Temperature for ruler display
bool load = true; //Load screens wildcard
```

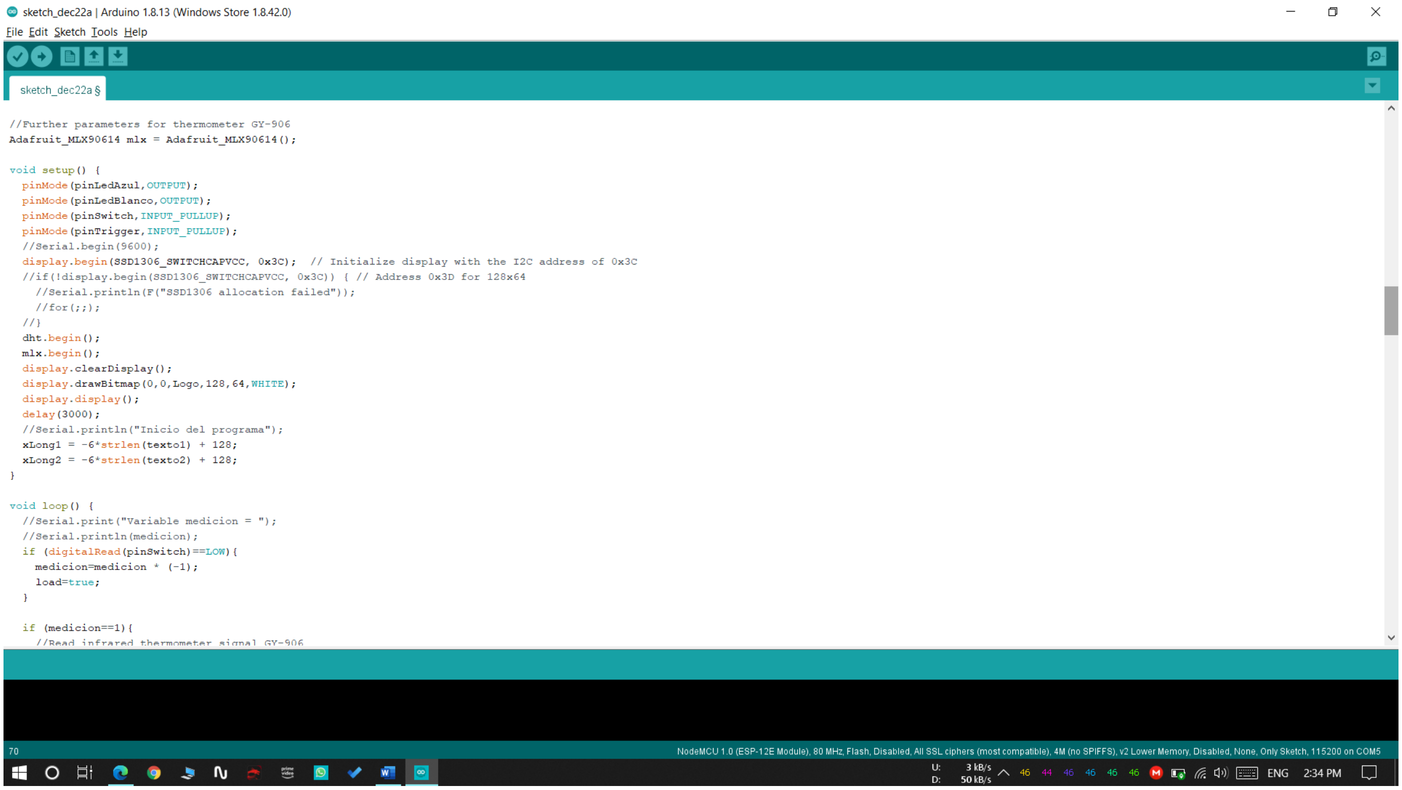
sketch_dec22a | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Edit Sketch Tools Help

sketch_dec22a §

```
bool load = true; //Load screens wildcard
char texto1[] = "Temperatura o punto de rocio:"; //Scrolling text 1
char texto2[] = "Temperatura aparente interior:"; //Scrolling text 2
int x1; //Scrolling coordinate text 1
int x2; //Scrolling coordinate text 2
int xLong1; //pixels over edge text 1
int xLong2; //pixels over edge text 2
bool hold;
bool scroll;
int j = 0;

//Further parameters for OLED screen
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
// Custom logo generated from <a href="https://diyusthad.com/image2cpp" rel="nofollow"> <a href="https://diyusthad.com/image2cpp" rel="nofollow"> https://diyusthad.com/image2cpp
</a>
</a>
const unsigned char Logo [] PROGMEM = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x1f, 0xc0, 0x00, 0x7f, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x01, 0x80, 0x00, 0x3f, 0xff, 0xff, 0xff, 0xff,
```

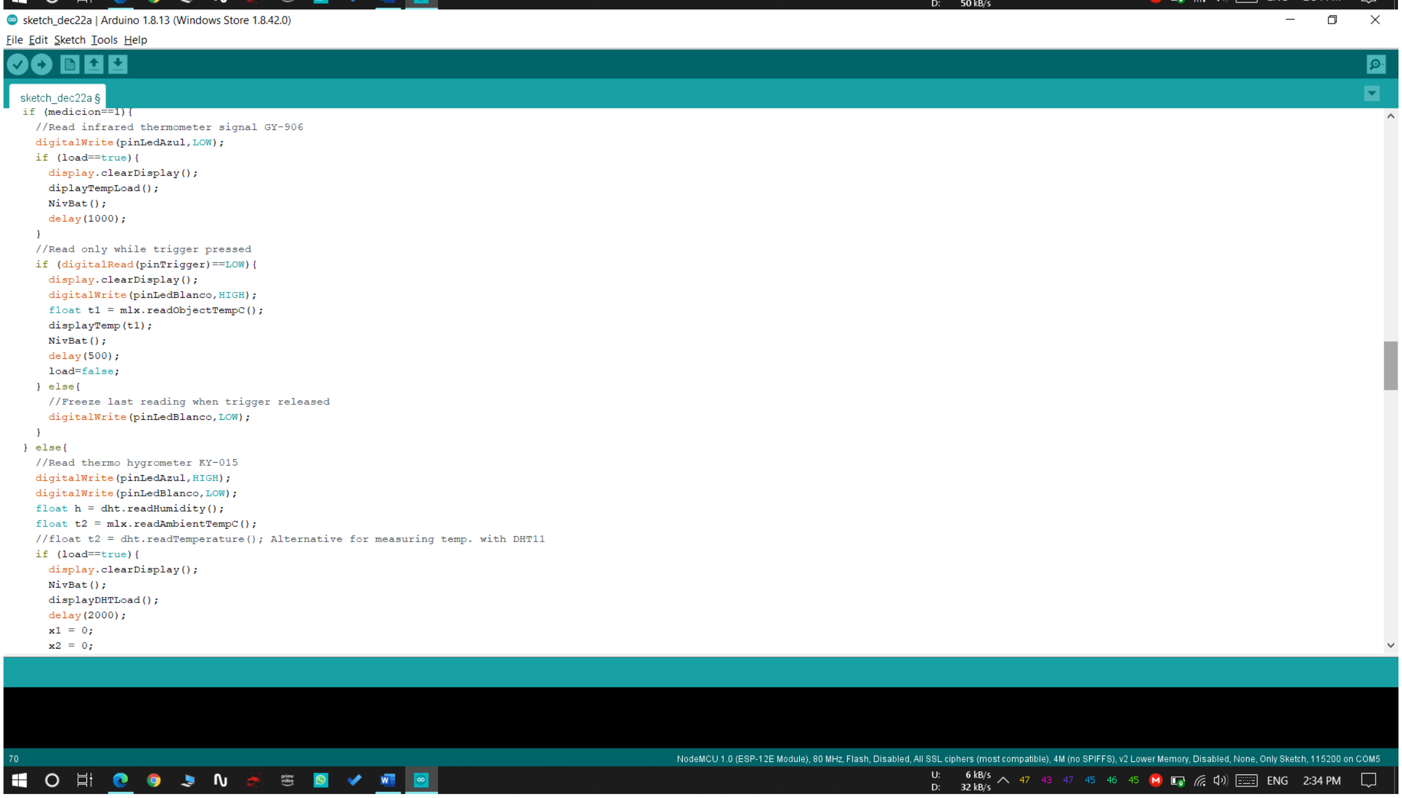



```
//Further parameters for thermometer GY-906
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

void setup() {
  pinMode(pinLedAzul,OUTPUT);
  pinMode(pinLedBlanco,OUTPUT);
  pinMode(pinSwitch,INPUT_PULLUP);
  pinMode(pinTrigger,INPUT_PULLUP);
  //Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  // Initialize display with the I2C address of 0x3C
  //if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    //Serial.println(F("SSD1306 allocation failed"));
    //for(;;);
  //}
  dht.begin();
  mlx.begin();
  display.clearDisplay();
  display.drawBitmap(0,0,Logo,128,64,WHITE);
  display.display();
  delay(3000);
  //Serial.println("Inicio del programa");
  xLong1 = -6*strlen(texto1) + 128;
  xLong2 = -6*strlen(texto2) + 128;
}

void loop() {
  //Serial.print("Variable medicion = ");
  //Serial.println(medicion);
  if (digitalRead(pinSwitch)==LOW){
    medicion=medicion * (-1);
    load=true;
  }

  if (medicion==1){
    //Read infrared thermometer signal GY-906
```




```
  if (medicion==1){
    //Read infrared thermometer signal GY-906
    digitalWrite(pinLedAzul,LOW);
    if (load==true){
      display.clearDisplay();
      diplayTempLoad();
      NivBat();
      delay(1000);
    }
    //Read only while trigger pressed
    if (digitalRead(pinTrigger)==LOW){
      display.clearDisplay();
      digitalWrite(pinLedBlanco,HIGH);
      float t1 = mlx.readObjectTempC();
      displayTemp(t1);
      NivBat();
      delay(500);
      load=false;
    } else{
      //Freeze last reading when trigger released
      digitalWrite(pinLedBlanco,LOW);
    }
  } else{
    //Read thermo hygrometer KY-015
    digitalWrite(pinLedAzul,HIGH);
    digitalWrite(pinLedBlanco,LOW);
    float h = dht.readHumidity();
    float t2 = mlx.readAmbientTempC();
    //float t2 = dht.readTemperature(); Alternative for measuring temp. with DHT11
    if (load==true){
      display.clearDisplay();
      NivBat();
      displayDHTLoad();
      delay(2000);
      x1 = 0;
      x2 = 0;
```

sketch_dec22a | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Edit Sketch Tools Help

sketch_dec22a §

```
      j=0;
    }
  }
}

//Function for updating and displaying battery level
void NivBat(){
  int anaBat = analogRead(pinNivBat);
  float volPinBat = float(anaBat) *(5.0/1023.0);
  float volBat = (volPinBat * (r1+r2))/r2;
  int x = 2;
  if (volBat>=9){
    x=10;
  } else if (volBat<9 && volBat>=8.5){
    x=8;
  } else if (volBat<8.5 && volBat>=8){
    x=6;
  } else if (volBat<8 && volBat>=7.5){
    x=4;
  } else if (volBat<7.5){
    x=2;
  }
  display.drawRect(112,1,12,8,WHITE);
  display.drawRect(123,3,4,4,WHITE);
  display.fillRect(113,2,x,6,WHITE);
  display.display();
  //Serial.println("Analog A1");
  //Serial.println(anaBat);
  //Serial.println("Voltaje A1");
  //Serial.println(volPinBat);
  //Serial.println("Voltaje bateria");
  //Serial.println(volBat);
  //Serial.println("ancho rectangulo");
  //Serial.println(x);
}
```

70 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5

ENG 2:34 PM

sketch_dec22a | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Edit Sketch Tools Help

sketch_dec22a §

```
//Function for displaying thermo hygrometer readings
void displayDHT(float H, float T, float D, float A, int X1, int X2){
  display.setTextColor(WHITE, BLACK);
  display.setTextSize(1);
  display.setTextWrap(false);
  display.setCursor(0,0);
  display.println("Temperatura:");
  display.print(T);
  display.println(" C");
  display.println("Humedad. Relativa:");
  display.print(H);
  display.println(" %");
  display.setCursor(X1,32);
  display.print(texto1);
  display.setCursor(0,40);
  display.print(D);
  display.println(" C");
  display.setCursor(X2,48);
  display.print(texto2);
  display.setCursor(0,56);
  display.print(A);
  display.print(" C");
  display.display();

}

//Function for loading screen of thermo hygrometer readings
void displayDHTLoad(){
  display.setTextSize(1);
  display.setTextColor(BLACK, WHITE); // 'inverted' text
  display.setCursor(10,2);
  display.println("Modo");
  display.setCursor(10,14);
  display.println("Termo-Hygrometro");
  display.display();
```

70 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5

ENG 2:35 PM

sketch_dec22a | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Edit Sketch Tools Help

sketch_dec22a §

```
  //Serial.print(T);
  //Serial.print(" C");
}

//Function for loading screen of infrared thermometer readings
void diplayTempLoad(){
  display.setTextSize(1);
  display.setTextColor(BLACK, WHITE); // 'inverted' text
  display.setCursor(25,10);
  display.println("Modo");
  display.setCursor(25,22);
  display.println("Temp. objeto");
  display.drawRoundRect(6, 3, 5, 49, 2, WHITE);  // Draw rounded rectangle (x,y,width,height,radius,color)
  display.drawCircle(8, 55, 7, WHITE);  // Draw circle (x,y,radius,color). X and Y are the coordinates for the center point
  display.fillCircle(8, 55, 6, BLACK);  // Draw black circle (x,y,radius,color). X and Y are the coordinates for the center point
  for (int i = 6; i<=45; i=i+3){
    display.drawLine(11, i, 12, i, WHITE);  // Draw line (x0,y0,x1,y1,color)
  }
  display.display();
  //Serial.println("Modo");
  //Serial.println("Temp. objeto");
}

//Function for calculating Dew Point at 2680 MASL and indoor precieved temperature
Temps tOut(float T, float RH){
  double h = 2680;
  double es = 6.108*exp((17.27*T)/(237.3+T));
  double e = (RH*es)/100.0;
  double b = (1/17.27)*(log(e/6.108));
  Temps t;
  t.tr = (273.0*b)/(1-b);
  t.ta = T+0.348*e-4.25;
  return t;
}

//Round function
```

70 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5

CONCLUSION & FUTURE SCOPE

The objective of the project Smart Thermal Hygrometer using ESP8266 module is to design a product which is very much useful in post COVID-19 era in order to perform activities with no physical contact. Our project model is mainly focused on designing a hygrometer in such a way that the sensed temperature and humidity is displayed and later on analysed and stored into Firebase API which can be accessed through a mobile. However, it not only helps us in reducing corona cases but also helps we performing activities contactless. It also helps to retain data for further analysation.

This model is very versatile and can be expanded into two major sectors i.e. software as well as hardware:
Since we are using ESP8266 Wi-Fi module it is possible to access the internet from any place and hence regardless of the location the value of temperature fed into microcontroller will be stored onto Firebase API which enables us to access the data whenever required.

EXPECTED OUTPUTS/ RESULTS

- It is possible to sense the surface temperature contactless, and at the same time determine if the site conditions are appropriate for the job.
- Apart from determining dew point temperature, the device can also display other parameters that the usual infrared thermometer cannot, such as relative humidity and apparent temperature (indoors).
- Thermal hygrometer can also be used to measure the humidity of painting walls.
- This proposed device is an essential requirement for the present world facing COVID-19 pandemic ,where it is necessary to maintain physical distance.

REFERENCE

1. D.K. Roveti. , Choosing a Humidity Sensor: A Review of Three Technologies. Sensors Magazine (2001).
2. Wexler, Arnold; Hyland, Richard W. (1964). "The NBS standard hygrometer". www.nist.gov. National Bureau of Standards. (2017).
3. "Spectral hygrometer - AMS Glossary". glossary.ametsoc.org. Retrieved 2019.
4. Makkonen, L., Laakso, T (2005) Humidity measurements in cold and humid environments. Boundary-Layer Meteorology.
5. J. Warne, The Practical Impacts of RTD and Thermometer Design on Wet and Dry Bulb Relative Humidity Measurements. Bureau of Meteorology, Melbourne (1998).
6. Robin E. Bentley, Temperature and humidity measurement, Springer Science & Business Media, 1998