Humidity and temperature

Sensor

Nume : Unguroiu Victor

Grupa : 237/2

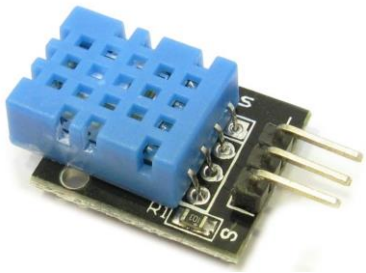
Ever wanted to know the temperature and humidity around you? If yes, this project will be very helpful for you. If no, it'll be cool anyway.

For this project you need following things :

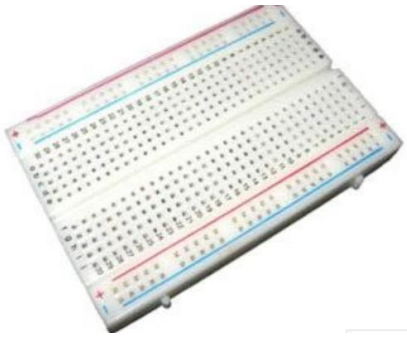
1x Arduino uno :

https://store.arduino.cc/products/arduinounorev3?queryID=826717a7c95881418581f7c757c86bf

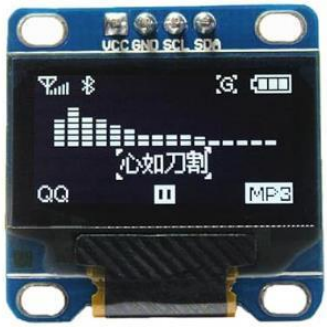
1x DHT11 temperature-humidity sensor : https://projecthub.arduino.cc/arcaegecengiz/using-dht11-12f621



1x breadboard : https://store.arduino.cc/products/breadboard-400contacts?queryID=80ac52d8f2ef83557f511642ffdbf4e6

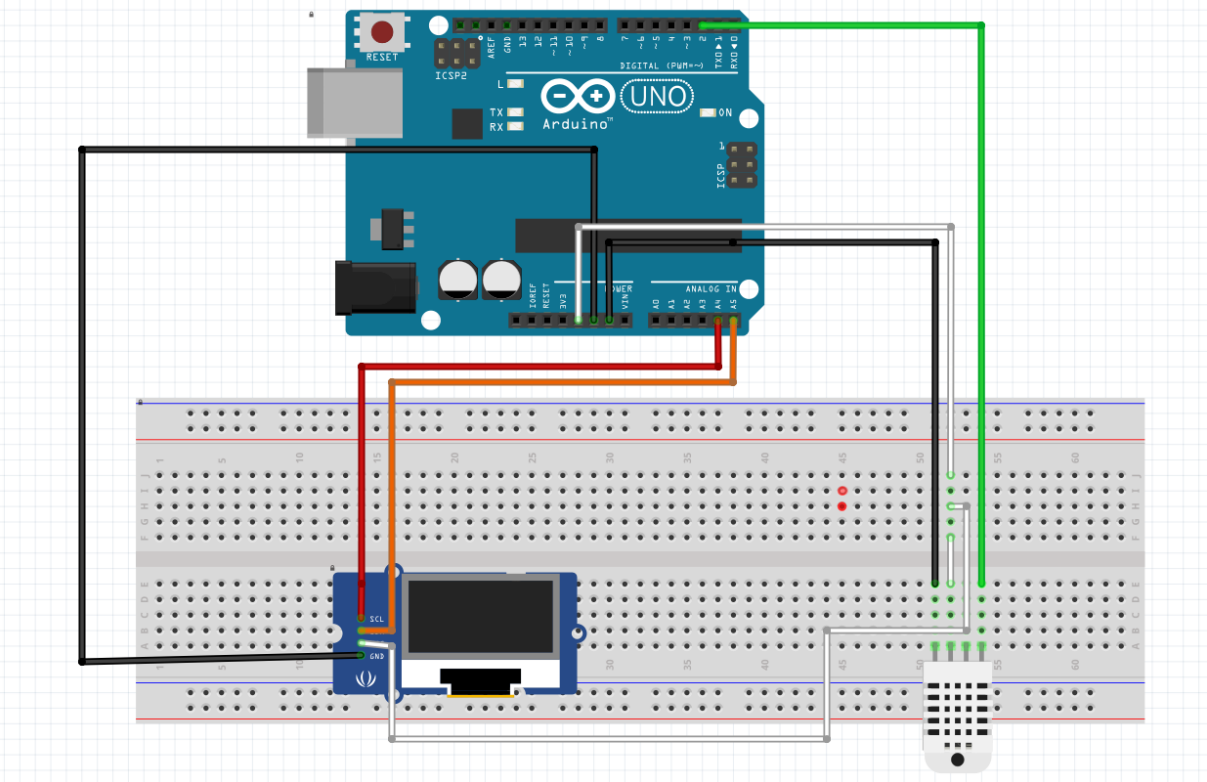


1x oled display 0.96 ssd1306 i2c : https://www.amazon.com.au/DIYmall-Serial-128x64-Display-Arduino/dp/B00O2KDQBE



Few jumpers : https://store.arduino.cc/products/10-jumper-wires-150mmmale?queryID=c6d1144eb16aa173093e4d6369717384

Below is the wiring diagram for how the components should be connected:



For the Code i used this Sketch in Arduino IDE

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Fonts/FreeMonoBold18pt7b.h>

#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);

int h;
int t;

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET     4 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define bitmap_height   128
#define bitmap_width    64
static const unsigned char PROGMEM logo_bmp[] =
{ 0x00, 0x00, 0x00, 0x00, 0xFF, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x3F, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x1F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x0F, 0x8F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x08, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x08, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x08, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x78, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x08, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x80, 0x08, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x01, 0x80, 0x78, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
```

```
  0x00, 0x01, 0x80, 0x08, 0x03, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x01, 0xC0, 0x08, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x01, 0xC0, 0x78, 0x03, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x01, 0xC0, 0x78, 0x07, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x03, 0xC0, 0x08, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x03, 0xE0, 0x08, 0x03, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x07, 0xE0, 0x78, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x07, 0xF0, 0x08, 0x03, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x0F, 0xF0, 0x08, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x0F, 0xF8, 0x78, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x1F, 0xF8, 0x08, 0x03, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x1F, 0xF8, 0x08, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x3F, 0xFC, 0x78, 0x03, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x3F, 0xFE, 0x08, 0x03, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x7F, 0xFE, 0x0F, 0x8F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x7F, 0xFF, 0x7F, 0x83, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0xFF, 0xFF, 0x0F, 0x83, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0xFF, 0xFF, 0x0F, 0x83, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x01, 0xFF, 0xFF, 0x7F, 0x80, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x03, 0xFF, 0xFF, 0x0F, 0x8F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x03, 0xFF, 0xFF, 0x0F, 0x87, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x03, 0xFF, 0xFF, 0x1F, 0x8F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x07, 0xFF, 0xFF, 0x7F, 0x83, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x0F, 0xFF, 0xFF, 0x0F, 0x83, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
```

```
  0x0F, 0xFF, 0xFF, 0x0F, 0x8F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x1F, 0xFF, 0xFF, 0x7F, 0x83, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x1F, 0xFF, 0xFF, 0x0F, 0x8F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x1F, 0x0F, 0xF3, 0xEF, 0x83, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x3E, 0x03, 0xE7, 0xCF, 0xC0, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x3C, 0x61, 0xC7, 0x9F, 0xE0, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x7C, 0x71, 0xCF, 0xBF, 0xE3, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x7C, 0x71, 0xCF, 0xBF, 0xF3, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x7C, 0x71, 0x9F, 0xBF, 0xE3, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0xFC, 0x71, 0x3F, 0xBF, 0xE3, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0xFC, 0x71, 0x38, 0x1F, 0xE7, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0xFC, 0x02, 0x20, 0x0F, 0x87, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0xFE, 0x06, 0x46, 0x03, 0x0F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0xFF, 0x8C, 0xC7, 0x18, 0x1F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0xFF, 0xFC, 0xC7, 0x1C, 0x7F, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0xFF, 0xF9, 0xC7, 0x1C, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0xFF, 0xF9, 0xC7, 0x1E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x7F, 0xF3, 0xC6, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x7F, 0xE3, 0xC0, 0x3E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x7F, 0xE7, 0xE0, 0x7E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x3F, 0xFF, 0xFF, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x1F, 0xFF, 0xFF, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x1F, 0xFF, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x0F, 0xFF, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
```

```
  0x07, 0xFF, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x03, 0xFF, 0xFF, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x7F, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x1F, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
  0x00, 0x03, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00
};

void setup() {
  Serial.begin(9600);
  dht.begin();

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for (;;); // Don't proceed, loop forever
  }
  // Clear the buffer
  display.clearDisplay();
  printText();
  delay(1500);
}

void loop() {
  h = dht.readHumidity();
  t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  showBitmap();
  printText();
  display.display();
  delay(500);
  display.clearDisplay();
}

void printText() {
  display.setFont(&FreeMonoBold18pt7b);
  display.setTextColor(WHITE);        // Draw white text
  display.setCursor(45, 28);            // Start at top-left corner
  display.print(t);
  display.drawCircle(92, 8, 3, WHITE);
  display.setCursor(100, 27);
  display.print("C");
```

```
  display.setCursor(45, 62);
  display.print(h);
  display.print("%");

}

void showBitmap(void) {
  display.drawBitmap(0, 0, logo_bmp, bitmap_height, bitmap_width, WHITE);
  //display.display();
}
```