**Expt- 6**

**Aim - : Design a sketch to interface DHT sensor with Arduino/ESP-32 board.**

**Parts Required:-**

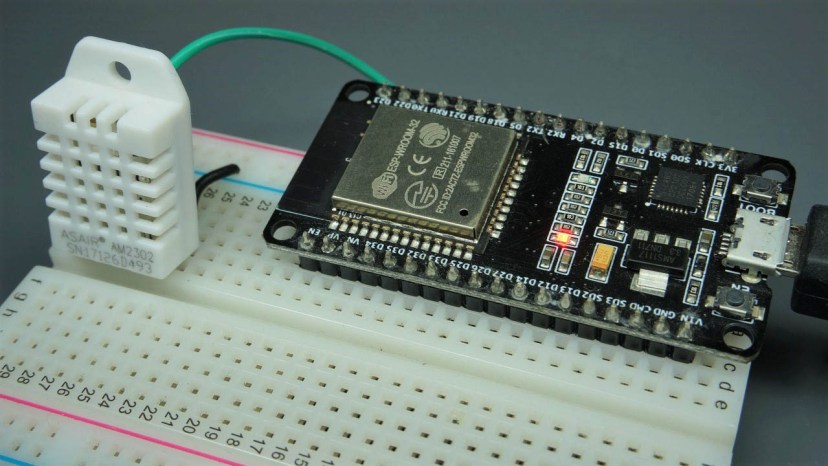
To follow this tutorial you need to wire the DHT11 or DHT22 temperature sensor to the ESP32. You need to use a 10k Ohm pull-up resistor.

Here’s a list of parts you need to build the circuit:

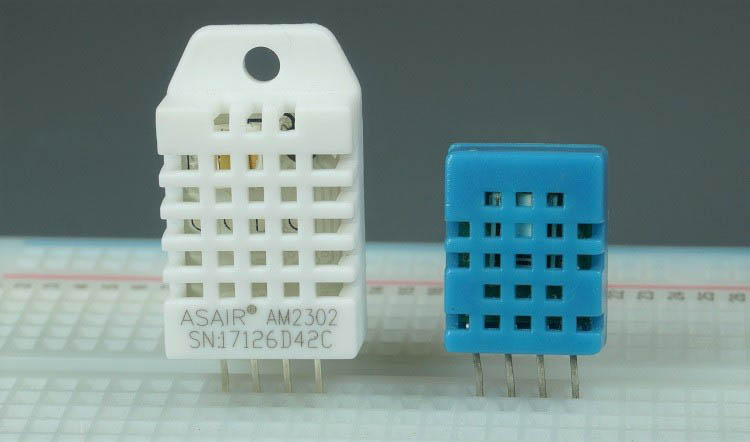
* [ESP32](https://makeradvisor.com/tools/esp32-dev-board-wi-fi-bluetooth/) (read [Best ESP32 development boards](https://makeradvisor.com/esp32-development-boards-review-comparison/))
* [DHT11](https://makeradvisor.com/tools/dht11-temperature-humidity-sensor/)or [DHT22](https://makeradvisor.com/tools/dht22-temperature-humidity-sensor/)temperature and humidity sensor
* [10k Ohm resistor](https://makeradvisor.com/tools/resistors-kits/)
* [Breadboard](https://makeradvisor.com/tools/mb-102-solderless-breadboard-830-points/)
* [Jumper wires](https://makeradvisor.com/tools/jumper-wires-kit-120-pieces/)

**ESP32 with DHT11/DHT22 Temperature and Humidity Sensor using Arduino IDE**

DHT11 and DHT22 temperature and humidity sensors with the ESP32 using Arduino IDE. introduction to these sensors, pinout, wiring diagram, and finally the Arduino sketch.



**DHT11 and DHT22 Temperature and Humidity Sensors**



**DHT11 vs DHT22**

The DHT11 and DHT22 are very similar, but differ in their specifications. The following table compares some of the most important specifications of the DHT11 and DHT22 temperature and humidity sensors. For a more in-depth analysis of these sensors, please check the sensors’ datasheet.

|  |  |  |
| --- | --- | --- |
|  | **DHT11** | **DHT22** |
|  |  |  |
| **Temperature range** | 0 to 50 ºC +/-2 ºC | -40 to 80 ºC +/-0.5ºC |
| **Humidity range** | 20 to 90% +/-5% | 0 to 100% +/-2% |
| **Resolution** | Humidity: 1% Temperature: 1ºC | Humidity: 0.1% Temperature: 0.1ºC |
| **Operating voltage** | 3 – 5.5 V DC | 3 – 6 V DC |
| **Current supply** | 0.5 – 2.5 mA | 1 – 1.5 mA |
| **Sampling period** | 1 second | 2 seconds |

The DHT11 has a smaller range and it’s less accurate. However, you can request sensor readings every second. It’s also a bit cheaper.

Despite their differences, they work in a similar way, and you can use the same code to read temperature and humidity. You just need to select in the code the sensor type you’re using.

**DHT Pinout**

DHT sensors have four pins as shown in the following figure. However, if you get your DHT sensor in a breakout board, it comes with only three pins and with an internal pull-up resistor on pin 2.

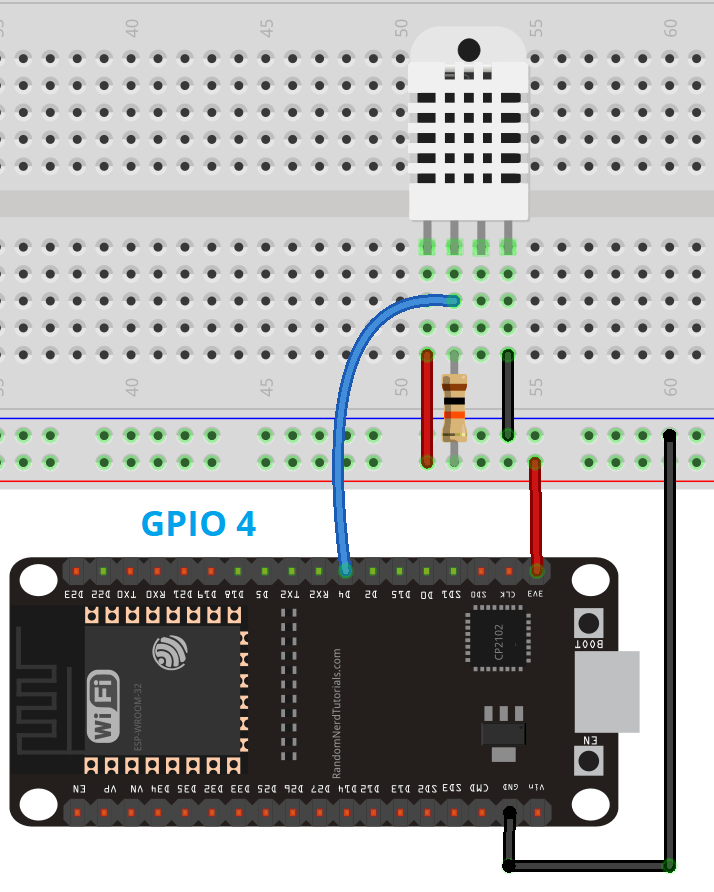


The following table shows the DHT22/DHT11 pinout. When the sensor is facing you, pin numbering starts at 1 from left to right

|  |  |
| --- | --- |
| **DHT pin** | **Connect to** |
| **1** | 3.3V |
| **2** | Any digital GPIO; also connect a 10k Ohm pull-up resistor |
| **3** | Don’t connect |
| **4** | GND |

**Schematic Diagram**

Wire the DHT22 or DHT11 sensor to the ESP32 development board as shown in the following schematic diagram.



In this example, we’re connecting the DHT data pin to GPIO 4. However, you can use any other suitable digital pin.

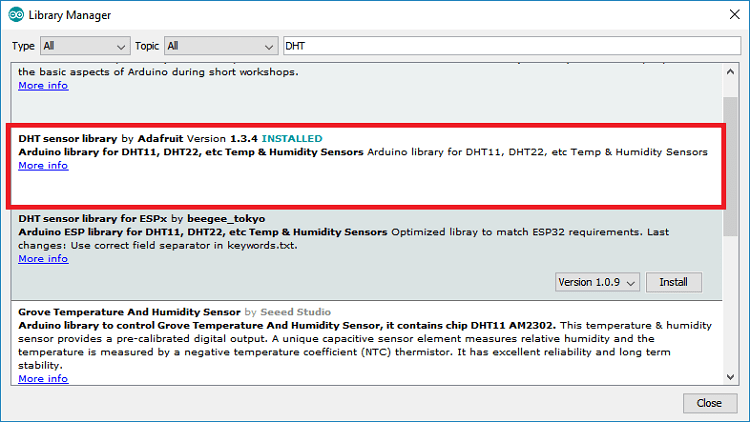
Learn how to use the ESP32 GPIOs with our guide: [ESP32 Pinout Reference: Which GPIO pins should you use?](https://randomnerdtutorials.com/esp32-pinout-reference-gpios/)

**Installing Libraries**

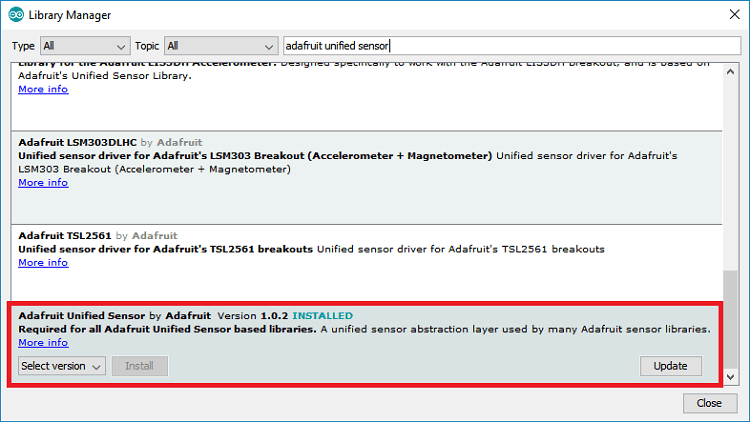
To read from the DHT sensor, we’ll use the [DHT library from Adafruit](https://github.com/adafruit/DHT-sensor-library). To use this library you also need to install the [Adafruit Unified Sensor library](https://github.com/adafruit/Adafruit_Sensor). Follow the next steps to install those libraries.

Open your Arduino IDE and go to **Sketch**> **Include Library** > **Manage Libraries**. The Library Manager should open.

Search for “**DHT**” on the Search box and install the DHT library from Adafruit.



After installing the DHT library from Adafruit, type “**Adafruit Unified Sensor**” in the search box. Scroll all the way down to find the library and install it.



After installing the libraries, restart your Arduino IDE.

**ESP32 Reading Temperature and Humidity Sketch**

To read temperature and humidity from the DHT sensor, we’ll use an example based on the Adafruit DHT library. Copy the following code to your Arduino IDE.

// Example testing sketch for various DHT humidity/temperature sensors written by ladyada

// REQUIRES the following Arduino libraries:

// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library

// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit\_Sensor

#include "DHT.h"

#define DHTPIN 4 // Digital pin connected to the DHT sensor

// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --

// Pin 15 can work but DHT must be disconnected during program upload.

// Uncomment whatever type you're using!

//#define DHTTYPE DHT11 // DHT 11

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V

// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1

// to 3.3V instead of 5V!

// Connect pin 2 of the sensor to whatever your DHTPIN is

// Connect pin 4 (on the right) of the sensor to GROUND

// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.

// Note that older versions of this library took an optional third parameter to

// tweak the timings for faster processors. This parameter is no longer needed

// as the current DHT reading algorithm adjusts itself to work on faster procs.

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(9600);

Serial.println(F("DHTxx test!"));

dht.begin();

}

void loop() {

// Wait a few seconds between measurements.

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));

Serial.print(h);

Serial.print(F("% Temperature: "));

Serial.print(t);

Serial.print(F("°C "));

Serial.print(f);

Serial.print(F("°F Heat index: "));

Serial.print(hic);

Serial.print(F("°C "));

Serial.print(hif);

Serial.println(F("°F"));

}

[View raw code](https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/raw/master/Projects/Arduino/ESP32_DHTtester.ino)

There are many comments throughout the code with useful information. So, you might want to take a look at the comments. Continue reading to learn how the code works.

**How the Code Works**

First, you need to import the DHT library:

#include "DHT.h"

Then, define the digital pin that the DHT sensor data pin is connected to. In this case, it’s connected to GPIO 4.

#define DHTPIN 4 // Digital pin connected to the DHT sensor

Then, you need to select the DHT sensor type you’re using. The library supports DHT11, DHT22, and DHT21. Uncomment the sensor type you’re using and comment all the others. In this case, we’re using the DHT22 sensor.

//#define DHTTYPE DHT11 // DHT 11

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

Create a DHT object called dht on the pin and with the sensor type you’ve specified previously.

DHT dht(DHTPIN, DHTTYPE);

In the setup(), initialize the Serial debugging at a baud rate of 9600, and print a message in the Serial Monitor.

Serial.begin(9600);

Serial.println(F("DHTxx test!"));

Finally, initialize the DHT sensor.

dht.begin();

The loop() starts with a 2000 ms (2 seconds) delay, because the DHT22 maximum sampling period is 2 seconds. So, we can only get readings every two seconds.

delay(2000);

The temperature and humidity are returned in float format. We create float variables h, t, and f to save the humidity, temperature in Celsius and temperature in Fahrenheit, respectively.

Getting the humidity and temperature is as easy as using the readHumidity() and readTemperature() methods on the dht object, as shown below:

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

If you want to get the temperature in Fahrenheit degrees, you need to pass the true parameter as argument to the readTemperature() method.

float f = dht.readTemperature(true);

There’s also an if statement that checks if the sensor returned valid temperature and humidity readings.

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

After getting the humidity and temperature, the library has a method that computes the heat index. You can get the heat index both in Celsius and Fahrenheit as shown below:

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Finally, print all the readings on the Serial Monitor with the following commands:

Serial.print(F("Humidity: "));

Serial.print(h);

Serial.print(F("% Temperature: "));

Serial.print(t);

Serial.print(F("°C "));

Serial.print(f);

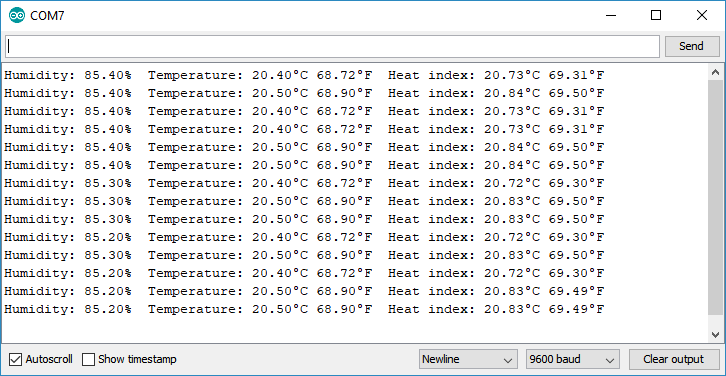
Serial.print(F("°F Heat index: "));

Serial.print(hic);

Serial.print(F("°C "));

Serial.print(hif);

Serial.println(F("°F"));

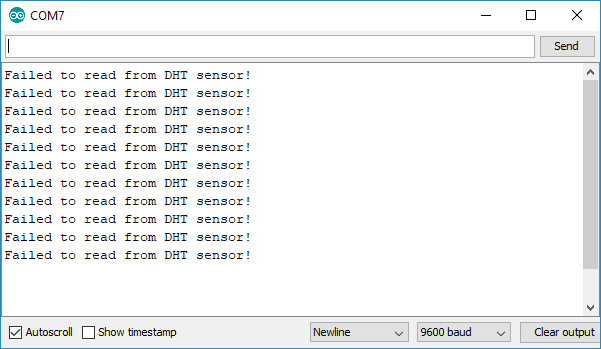


**Troubleshooting – Failed to read from DHT sensor**

If you’re trying to read the temperature and humidity from the DHT11/DHT22 sensor and you get an error message in your Serial Monitor, follow the next steps to see if you can make your sensor work (or read our dedicated [DHT Troubleshooting Guide](https://randomnerdtutorials.com/solved-dht11-dht22-failed-to-read-from-dht-sensor/)).

**“Failed to read from DHT sensor!” or Nan readings**

If your DHT sensor returns the error message “Failed to read from DHT sensor!” or the DHT readings return “Nan”:



Try one of the following troubleshooting tips:

* **Wiring:** when you’re building an electronics project, you need to double-check the wiring or pin assignment. After checking and testing that your circuit is properly connected, if it still doesn’t work, continue reading the next troubleshooting tips.
* **Power:** the DHT sensor has an operating range of 3V to 5.5V (DHT11) or 3V to 6V (DHT22). If you’re powering the sensor from the ESP32 3.3V pin, in some cases powering the DHT with 5V solves the problem.
* **Bad USB port or USB cable:** sometimes powering the ESP32 directly from a PC USB port is not enough. Try to plug it to a USB hub powered by an external power source. It might also help replacing the USB cable with a better or shorter one. Having a USB port that supplies enough power or using a good USB cable often fixes this problem.
* **Power source:** as mentioned in the previous tip, your ESP might not be supplying enough power to properly read from the DHT sensor. In some cases, you might need to power the ESP with a power source that provides more current.
* **Sensor type:** double-check that you’ve uncommented/commented in your code the right sensor for your project. In this project, we were using the DHT22:

//#define DHTTYPE DHT11 // DHT 11

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

* **Sampling rate:** the DHT sensor is very slow getting the readings (the sensor readings may take up to 2 seconds). In some cases, increasing the time between readings solves the problem.
* **DHT sensor is fried or broken:** unfortunately, these cheap sensors sometimes look totally fine, but they are fried/broken. So, even though you assembled the right circuit and code, it will still fail to get the readings. Try to use a different sensor to see if it fixes your problem.
* **Wrong baud rate or failed to upload code:** if you don’t see anything in your Arduino IDE Serial Monitor double-check that you’ve selected the right baud rate, COM port or that you’ve uploaded the code successfully.

While building our projects, we’ve experienced similar issues with the DHT and it was always solved by following one of the methods described earlier.

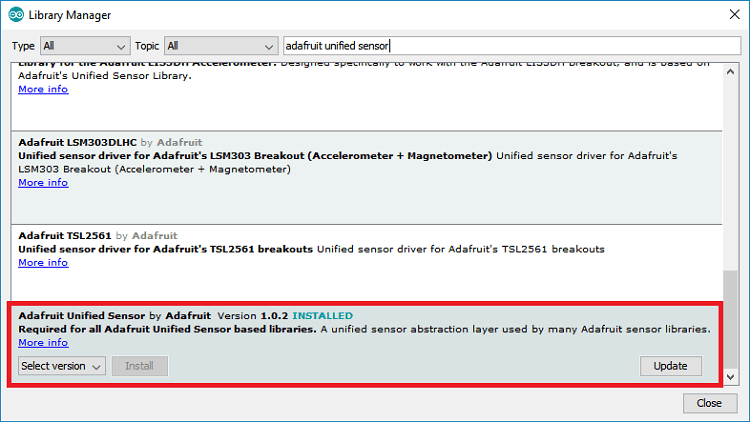
**Fatal error: Adafruit\_Sensor.h: No such file or directory**

There’s also a common error that happens when you try to compile the code. If you receive the following error:

fatal error: Adafruit\_Sensor.h: No such file or directory

#include <Adafruit\_Sensor.h>

You need to install the Adafruit Unified Sensor driver library. In your Arduino IDE, type in the search box “**Adafruit Unified Sensor**“, scroll all the way down to find the library and install it.



After installing the library, restart your Arduino IDE and the code should compile without the error message.