



instructables

## ESP32 WiFi Weather Station With a BME280 Sensor



by educ8s

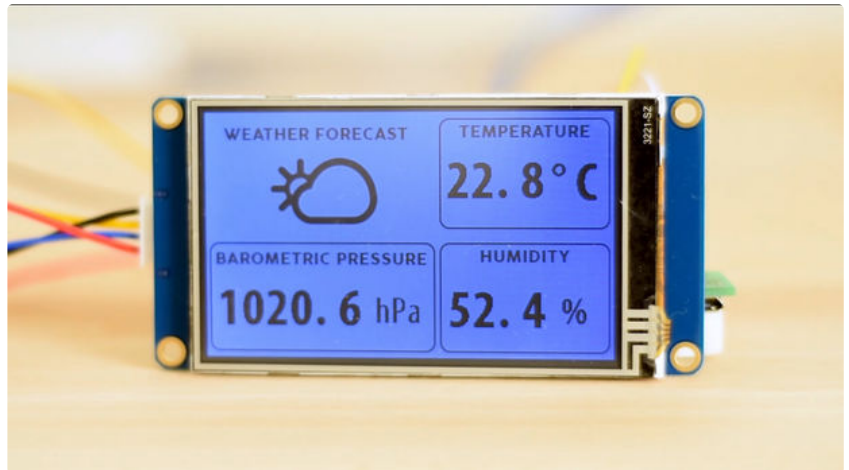
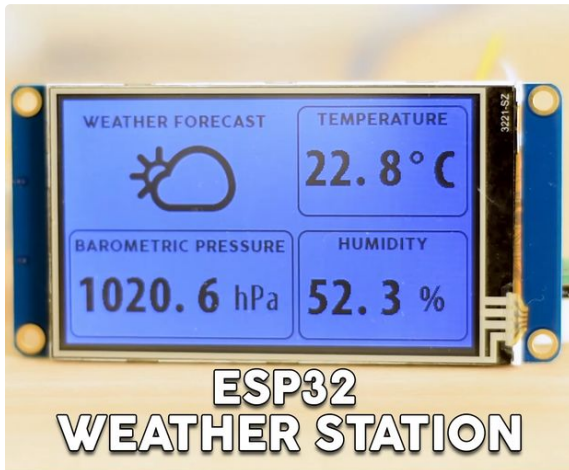
Dear friends welcome to another tutorial! In this tutorial we are going to build a **WiFi enabled weather station project**! We are going to use the new, impressive ESP32 chip for the first time along with a Nextion display.

In this video, we are going to make this. It is yet another weather station project I know, but this time we use the new ESP32 chip! We also use the new BME280 sensor which measures the temperature, the humidity, and the barometric pressure. When we power up the project, it connects to the WiFi network, and it is going to retrieve the weather forecast for my location from the openweathermap website. Then it

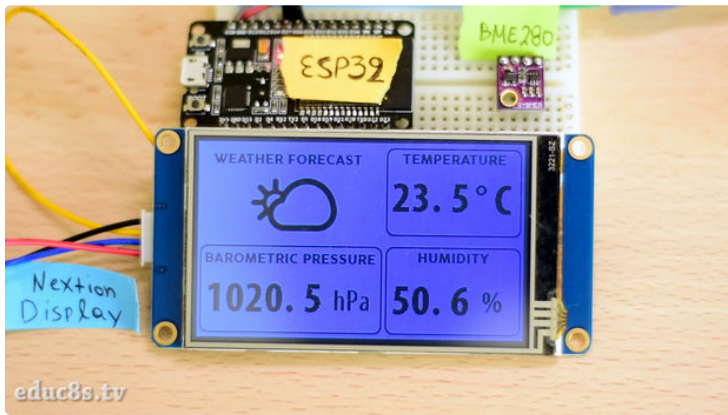
will display the forecast on this 3.2" Nextion Touch Display along with the readings from the sensor! The readings are updated every two seconds and the weather forecast every hour! As you can see, in this project we use the latest technologies available to a maker today! If you are a DIY veteran, you can build this project in five minutes.

If you are a beginner, you have to watch a couple of videos before attempting this project. You can find links to these video in this Instructable, don't worry.

Let's start!



[https://www.youtube.com/watch?v=el-4\\_QyVenw](https://www.youtube.com/watch?v=el-4_QyVenw)



---

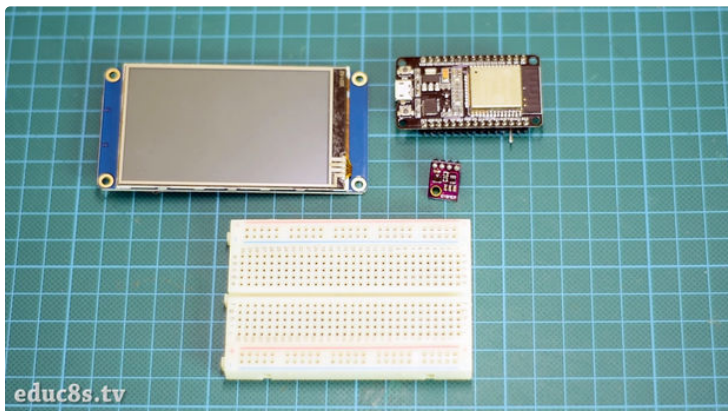
## Step 1: Get All the Parts

In order to build this project we need the following parts:

- An ESP32 Board <http://educ8s.tv/part/ESP32>
- A BME280 I2C sensor <http://educ8s.tv/part/BME280>
- A 3.2" Nextion display <http://educ8s.tv/part/Nextion32>
- A Small Breadboard <http://educ8s.tv/part/SmallBreadboard>
- Some wires <http://educ8s.tv/part/Wires>

The cost of the project is around 30\$.

Instead of the ESP32, we could use the cheaper ESP8266 chip, but I decided to use the ESP32 to gain some experience with it and see what works and what not.



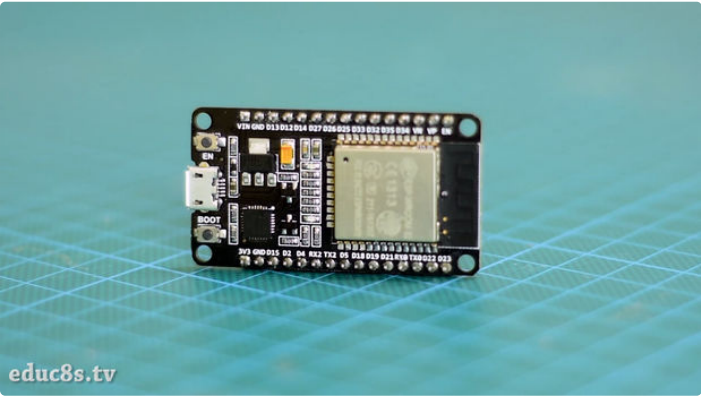
## Step 2: The ESP32

This is the first project I ever build with the **new ESP32 chip**.

If you are not familiar with it, the ESP32 chip is the successor of the popular ESP8266 chip we have used many times in the past. The ESP32 is a beast! It offers two 32 processing cores which operate at 160MHz, a massive amount of memory, WiFi,

Bluetooth and many other features with a cost of around 7\$! Amazing stuff!

Please watch the detailed review I have prepared for this board. I have attached the video on this Instructable. It will help understand why this chip will change the way we make things forever!



SPECS/BOARD	ESP32	ESP8266	ARDUINO UNO
Number of Cores	2	1	1
Architecture	32 Bit	32 Bit	8 Bit
CPU Frequency	160 MHz	80 MHz	16 MHz
WiFi	YES	YES	NO
BLUETOOTH	YES	NO	NO
RAM	512 KB	160 KB	2 KB
FLASH	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
ADC Pins	18	1	6
DAC Pins	2	0	0

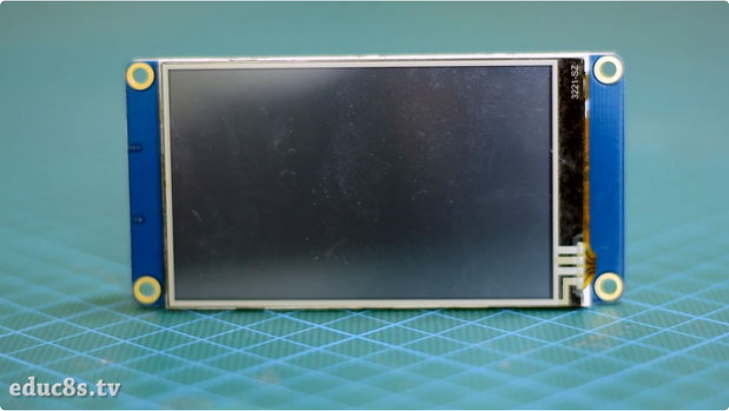
<https://www.youtube.com/watch?v=Mq1YhgS5VkY>

## Step 3: The Nextion Display

Also, this is the first project I build with a **Nextion touch display**.

The Nextion displays are new kind of displays. They have their own ARM processor at the back which is responsible for driving the display and creating the graphical user interface. So, we can use them with any microcontroller and achieve spectacular results.

I have prepared a detailed Instructable about this Nextion display which explains in depth how they work, how to use them and their drawbacks. You can read it by clicking here:  
<https://www.instructables.com/id/Arduino-Weather-S...>



## Step 4: BME280 Sensor

The BME280 is a new great sensor from Bosch.

So far I was using the BMP180 sensor which can measure temperature and barometric pressure. The BME280 sensor can measure temperature, humidity, and barometric pressure! How cool is that! We just need one sensor to build a complete weather station!

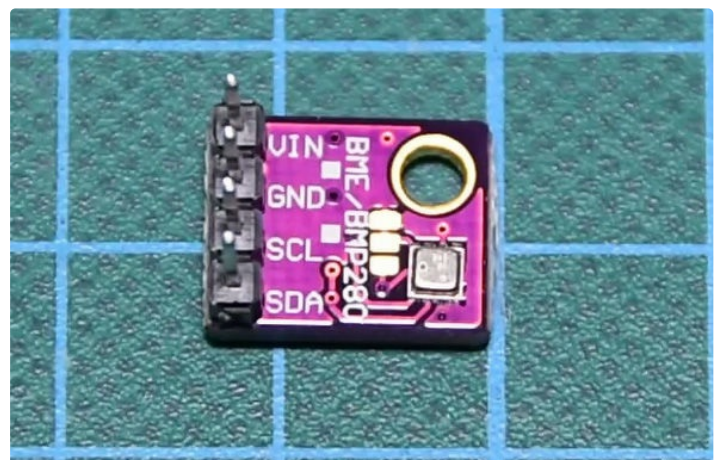
sensor so we can use it in our projects very easily! The cost of the sensor is about 5\$. You can get it here <http://educ8s.tv/part/BME280>

NOTE: We need the BME280 sensor. There is also a

In addition to that, the sensor is very small in size and very easy to use. The module we are going to use today uses the I2C interface so it makes communication with Arduino very easy. We only to connect power and two more wires to make it work.

There are already many libraries developed for this

BMP280 sensor which does not offer humidity measurement. Be careful to order to sensor you need.





## Step 5: Connecting the Parts

The connection of the parts is straightforward as you can see from the schematic diagram.

Since the BME280 sensor uses the I2C interface, we only need to connect two wires to communicate with ESP32. I have attached the sensor to Pins 26 and 27. In theory, every digital pin of the ESP32 board can be used with I2C peripherals. In practice though, I found out that some pins did not work because they are reserved for other uses. Pins 26 and 27 function great!

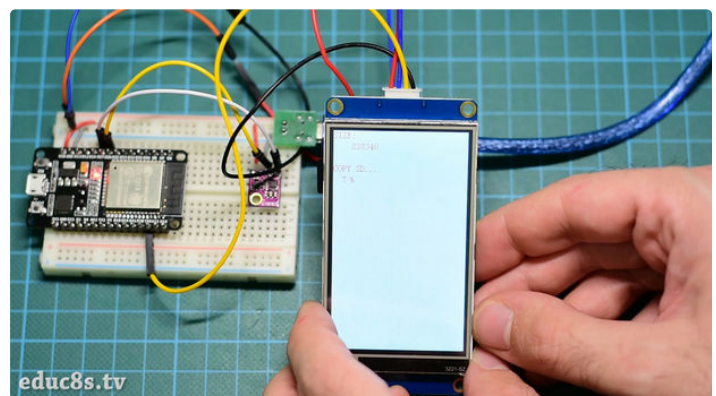
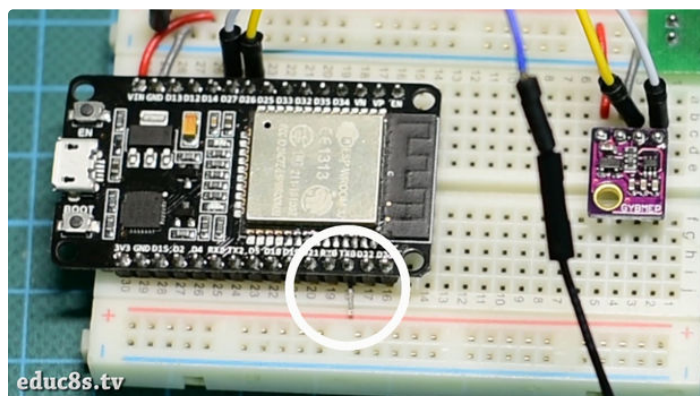
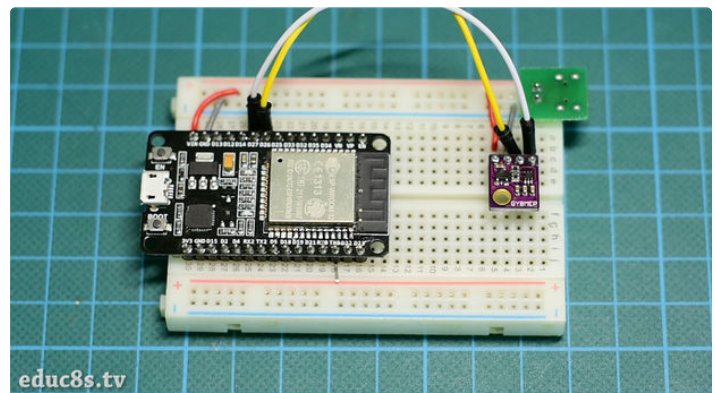
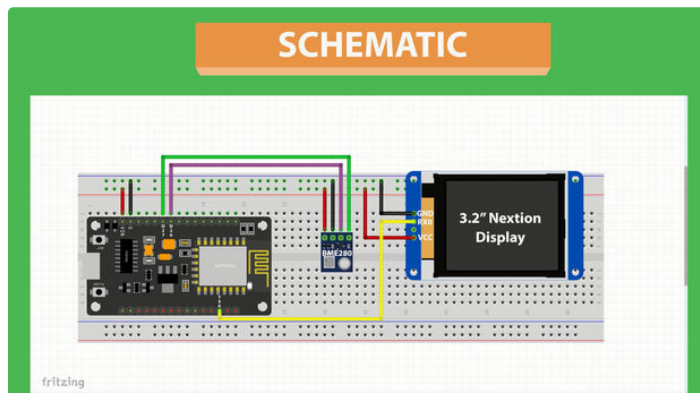
To send data to the display, we only need to connect one wire to the TX0 pin of the ESP32. I had to bend the pin like this to connect the female wire of the display since the ESP32 board is too big for this breadboard.

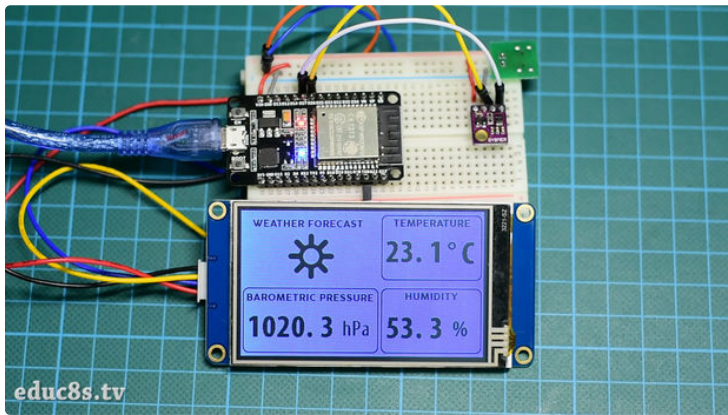
After connecting the parts, we have to load the code

to the ESP32, and we have to load the GUI to the Nextion display. If you have trouble uploading the program to the ESP32 board, hold down the BOOT button after pressing the upload button on the Arduino IDE.

To load the GUI to the Nextion display, copy the WeatherStation.tft file I am going to share with you to an empty SD card. Put the SD card into the SD card slot at the back of the display. Then power up the display, and the GUI will be loaded. Then remove the SD card and connect the power again.

After successfully loading the code the project will connect to the WiFi network, it will get the weather forecast from the openweathermap.org website, and it will display the readings from the sensor. Let's now see the software side of the project.





## Step 6: The Code of the Project

To parse the weather data, we need the excellent Arduino JSON library. We also need a library for the sensor.

? ESP32 BME280: <https://github.com/Takatsuki0204/BME280-I2C-ESP32>

? Arduino JSON: <https://github.com/bblanchon/ArduinoJson>

Let's see the code now.

At first, we have to set the SSID and the password of our WiFi network. Next, we have to enter the free APIKEY from [openweathermap.org](https://openweathermap.org) website. To create your own API key, you have to sign up on the website. Getting current weather data and forecast is free, but the website offers more options if you are willing to pay some money. Next, we have to find the id of our location. Find your location and copy the ID which can be found at the URL of your location.

Then enter your city's id in the CityID variable. Also, enter the altitude of your town in this variable. This value is needed for accurate barometric pressure readings from the sensor.

```
const char* ssid = "yourSSID";
const char* password = "yourPassword";
String CityID = "253394"; //Sparta, Greece
String APIKEY = "yourAPIkey";
#define ALTITUDE 216.0 // Altitude in Sparta, Greece
```

Now we are ready to move on.

At first, we initialize the sensor, and we connect to the WiFi Network. Then we request weather data from the server.

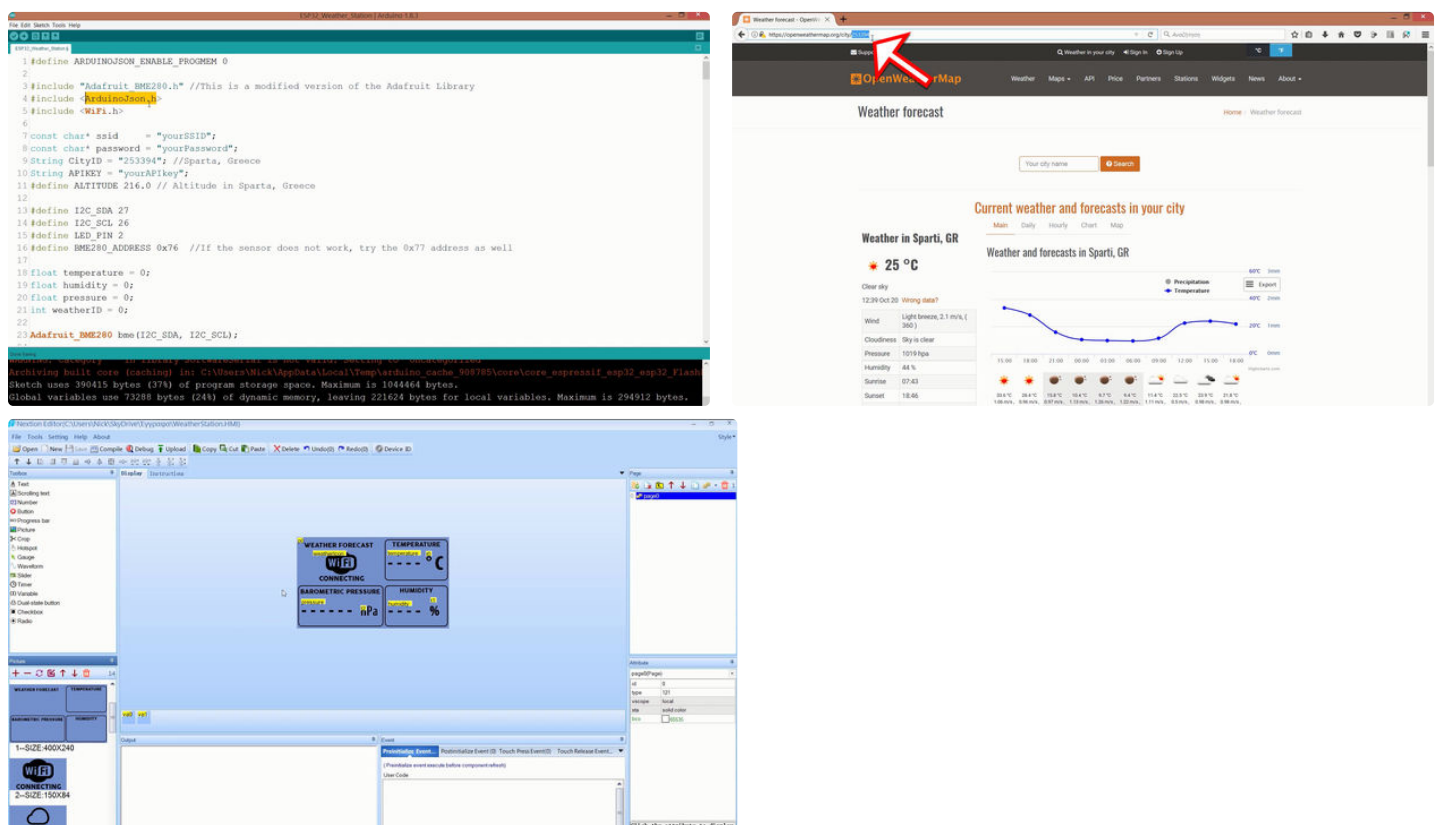
We get a reply with the weather data in JSON format. Before sending the data to the JSON library, I manually delete some characters that were causing me problems. Then the JSON library takes over, and we can easily save the data that we need in variables. After we have kept the data in variables, all we have to do is to display them on the screen and wait an hour before requesting new data from the server. The only information I present is the weather forecast, but you can display more information if you wish. It all here saved in variables. Then we read the temperature, the humidity and the barometric pressure from the sensor and we send the data to the Nextion display.

To update the display, we simply send some commands to the serial port like this:

```
void showConnectingIcon()
{
  Serial.println();
  String command = "weatherIcon.pic=3";
  Serial.print(command);
  endNextionCommand();
}
```

The Nextion GUI consists of a background, some textboxes and a picture that changes depending on the weather forecast. Please watch Nextion display tutorial for more information. You can quickly design your own GUI if you wish and display more things on it.

**As always you can find the code of the project attached to this Instructable!**



[http://www.instructable...](http://www.instructable.com/ESP32-WiFi-Weather-Station-With-a-BME280-Sensor)

Download (<https://cdn.instructables.com/ORIG/FOE/LC5H/JA36PDKK/FOELC5HJA36PDKK.zip>)

(<https://cdn.instructables.com/ORIG/FOE/LC5H/JA36PDKK/FOELC5HJA36PDKK.zip>)

## Step 7: Final Thoughts & Improvements

As you can see, an experienced maker today can build exciting projects in just a few hours with a few lines of code and only three parts! A project like this would have been impossible to make even two years ago!

Of course, this is just the beginning of the project. I would like to add many features to it, like graphs, touch functionality that is now missing, maybe a bigger display and of course a beautiful looking 3D printed enclosure. I will also design a better looking

GUI and icons. I have some very fresh ideas to implement!

I would love to hear your opinion about today's project. What kind of features do you want me to add to the project? Do you like how it looks? How do you want to see it evolve? Please post your ideas in the comments section below; I love reading your thoughts!

