

دانشگاه صنعتی خواجه نصیرالدین طوسی

"به نام خدا"

پروژه ی مدار های الکتریکی

استاد پوریا کرمی

تهیه کننده: صبا سلیمان زاده (۴۰۳۲۱۸۰۳)

## فهرست مطالب:

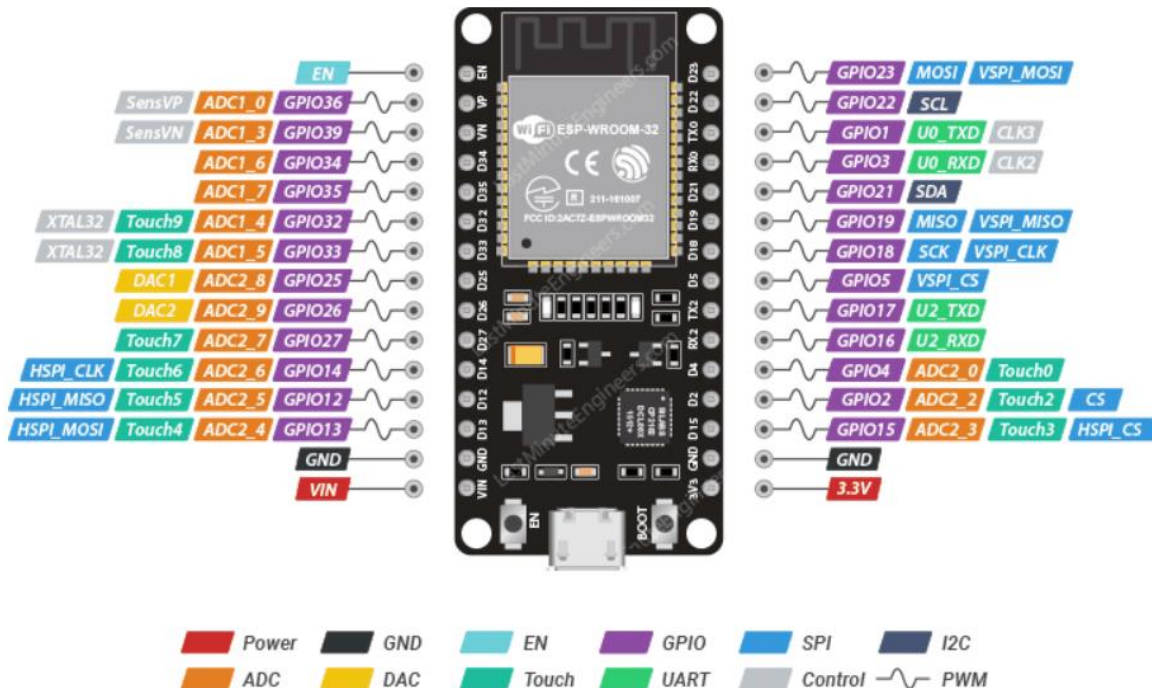
۳	برد ESP32-WROOM : .....
۵	راه اندازی وب سرور با ESP32 : .....
۶	سنسور دما و رطوبت و فشار هوا GY-BME280 : .....
۸	معرفی نمایشگر oled با رابط i2c : .....
۱۰	معرفی ماژول SD card reader : .....
۱۱	ارتباط spi : .....
۱۲	پیاده سازی نهایی پروژه در پروتئوس: .....
۱۳	قدم آخر: .....
۱۴	اهداف پروژه: .....
۱۴	کاربرد های پروژه: .....

## برد ESP32-WROOM :

برد ESP32-WROOM-32 یکی از محبوب‌ترین و پرکاربردترین ماژول‌های خانواده ESP32 است که توسط شرکت Espressif Systems طراحی شده. این ماژول به دلیل امکانات گسترده و قیمت مناسب، انتخاب اول بسیاری از توسعه‌دهندگان در پروژه‌های اینترنت اشیا (IoT)، کنترل از راه دور، و سیستم‌های هوشمند است.

در حقیقت اصلی‌ترین دلیلی که در این پروژه از ESP32 به جای Arduino استفاده شده، بخاطر وجود وایفای و بلوتوث و قابلیت ایجاد وب سرور توسط این برد بود که این برد را با سایر بردهای آردوینو متمایز می‌سازد.

15 کانال ADC با دقت 12 بیتی با محدوده‌های قابل انتخاب	15 کانال ADC
رابط UART با کنترل جریان و پشتیبانی IrDA	واحد ارتباط UART
25 پین PWM برای کنترل مواردی مانند سرعت موتور یا روشنایی LED دارد	25 خروجی PWM
دو DAC با دقت 8 بیتی برای تولید ولتاژ آنالوگ واقعی	2 کانال DAC
سه رابط SPI و یک رابط I2C برای اتصال سنسورها و تجهیزات جانبی مختلف و همچنین دو رابط I2S برای افزودن صدا به پروژه شما	واحد SPI، I2C و I2S
9 پین با حسگر لمسی خازنی	9 تاج پد

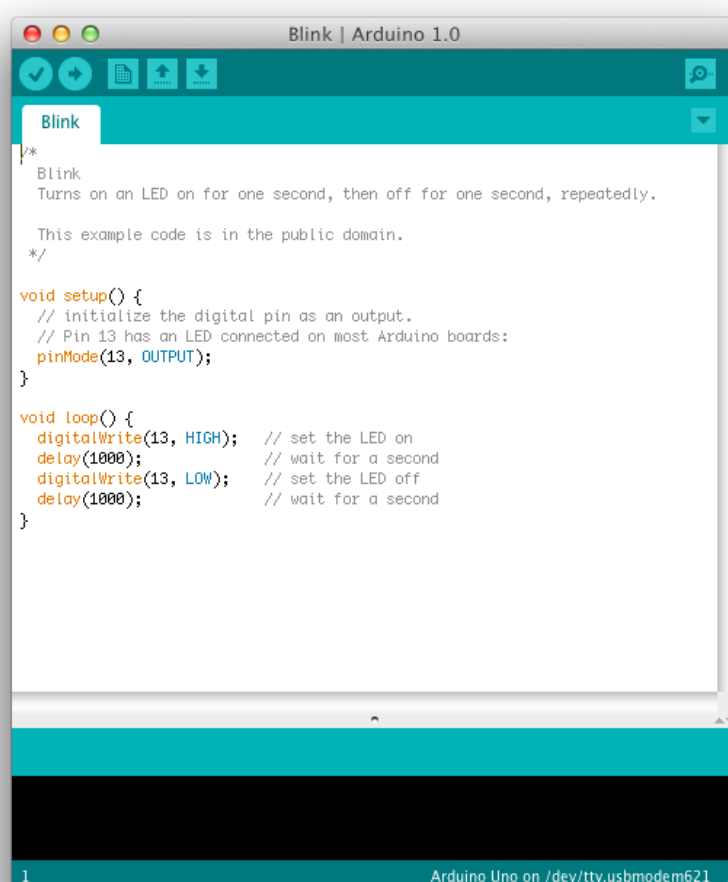


## تغذیه بردهای ESP32 :

ولتاژ کاری این میکروکنترلر 3.3 ولت است. در نتیجه برای تغذیه آن، اگر از خود میکروکنترلر استفاده می‌کنید لازم است که ولتاژ 3.3 ولت را به عنوان تغذیه ماژول تامین کنید. اما بردهای توسعه مبتنی بر میکروکنترلر ESP32 دارای یک رگولاتور ولتاژ 3.3 ولت هستند. در نتیجه در صورتی که از بردهای توسعه مبتنی بر میکروکنترلر ESP32 (مشابه بردهای معرفی شده در قبل) استفاده می‌کنید، می‌توانید تغذیه برد و همچنین برنامه‌ریزی کردن برد را از طریق پورت میکرو USB که بر روی برد قرار دارد انجام دهید. این بردها، دارای یک رگولاتور ولتاژ 3.3 ولت هستند که عمل تبدیل ولتاژ 5 ولت تامین شده از طریق USB به ولتاژ 3.3 ولت مورد نیاز میکروکنترلر را انجام می‌دهد.

## محیط‌های برنامه‌نویسی:

یکی از دلایل محبوبیت میکروکنترلر ESP32 قابلیت برنامه‌نویسی ساده آن است. این برنامه‌نویسی می‌تواند در بسیاری از محیط‌ها گوناگون انجام شود که در این پروژه ما با IDE Arduino برنامه نویسی می‌کنیم.



```
Blink | Arduino 1.0

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);            // wait for a second
}
```

1 Arduino Uno on /dev/tty.usbmodem621

## راه اندازی وب سرور با ESP32 :

هدف از این پروژه، راه اندازی یک وب سرور داخلی با استفاده از برد ESP32 است که پس از اتصال به شبکه Wi-Fi، آی پی اختصاصی دریافت کرده و سپس داده های محیطی شامل دما، رطوبت و فشار هوا را از طریق سنسور (BME280) دریافت کرده و در قالب یک صفحه HTML در مرورگر نمایش دهد.

برد ESP32 دارای قابلیت Wi-Fi داخلی است که می توان از آن برای اتصال به مودم و دریافت آی پی استفاده کرد. با استفاده از کتابخانه WebServer، می توان یک وب سرور راه اندازی کرده و داده هایی مانند دما، رطوبت و فشار را که از طریق سنسور دریافت می شوند، در قالب یک رابط گرافیکی HTML برای کاربر نمایش داد.

### مود های کاری ESP32 :

برد ESP32 به عنوان یکی از محبوب ترین ماژول های اینترنت اشیا (IoT) ، قابلیت برقراری ارتباط بی سیم از طریق Wi-Fi را در حالت های مختلفی فراهم می کند. این حالت ها بسته به نیاز پروژه و نوع ارتباط مورد نظر با شبکه، انتخاب و پیاده سازی می شوند. در این بخش از گزارش، به بررسی مود های کاری مختلف ESP32 در ارتباط Wi-Fi می پردازیم.

ESP32 از سه مود اصلی برای ارتباط Wi-Fi پشتیبانی می کند:

#### ۱) حالت Station Mode (STA) :

در این حالت، ESP32 همانند یک دستگاه عادی مانند موبایل یا لپ تاپ به یک مودم (Access Point) متصل می شود. پس از اتصال موفق، یک آدرس IP از مودم دریافت می کند و از طریق آن می تواند داده ها را ارسال یا دریافت کند.

دستور مربوطه در برنامه :

```
WiFi.mode(WIFI_STA); // ایستگاه مود تنظیم
WiFi.begin(ssid, password);
```

#### ۲) حالت Access Point Mode (AP) :

در این حالت، خود ESP32 نقش یک مودم یا Access Point را ایفا می کند. یعنی شبکه ای وای فای ایجاد کرده و سایر دستگاه ها (مانند موبایل یا لپ تاپ) می توانند مستقیماً به آن متصل شوند.

```
WiFi.mode(WIFI_AP); // تنظیم حالت نقطه دسترسی
WiFi.softAP("ESP32-AccessPoint", "12345678"); // نام شبکه و رمز
```

#### ۳) حالت Access Point & Station :

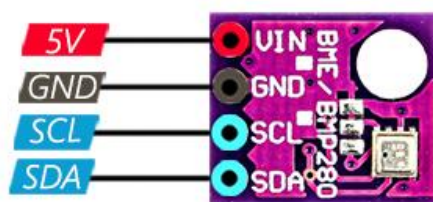
ESP32 به عنوان Access Point عمل می کند. در عین حال به عنوان Station، به Access Point دیگر نیز متصل است.

## سنسور دما و رطوبت و فشار هوا GY-BME280 :

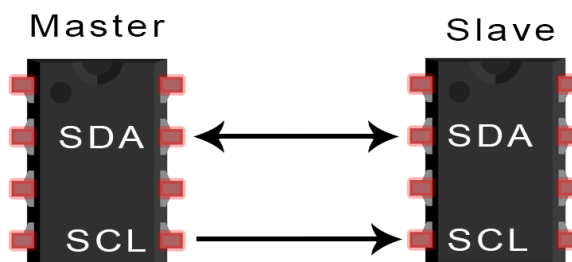
ماژول سنسور BME280 یک ماژول اندازه گیری دما، رطوبت و فشار هوای بسیار دقیق است. ویژگی‌های این ماژول در زیر آورده شده است:

- بازه اندازه گیری دما -40 تا +80 درجه سلسیوس. با دقت اندازه گیری 1 درجه سلسیوس در بازه ی 0 تا +65 درجه و 5 درجه سلسیوس در خارج از این بازه.
- بازه اندازه گیری رطوبت هوا در محدوده 0 تا 100% با دقت اندازه گیری 3%
- بازه اندازه گیری فشار هوا 330hPa تا 1100hPa با دقت اندازه گیری 1hPa

همچنین این ماژول ارتفاع از سطح دریا را به صورت غیرمستقیم با استفاده از فشار هوا، محاسبه می کند. پروتکل ارتباطی این سنسور می تواند SPI یا I2C باشد. (در این پروژه ما از ارتباط i2c این ماژول استفاده کرده ایم).



\* همانطور که میدانیم در ارتباط i2c از دوسیم برای انتقال داده بین دستگاه ها استفاده می کند:



- SDA (Serial Data) خطی برای Master و Slave برای ارسال و دریافت داده ها.
- SCL (Serial Clock) خطی که سیگنال ساعت را حمل می کند.

I2C یک پروتکل ارتباطی سریال است ، بنابراین داده ها نوبتی در امتداد یک سیم (خط SDA) منتقل می شوند.

مانند SPI ، ارتباط I2C نیز همگام است ، بنابراین خروجی بیت ها با نمونه برداری از بیت ها توسط یک سیگنال ساعت مشترک (Clock) بین Master و Slave هماهنگ می شوند. سیگنال ساعت همیشه توسط Master کنترل می شود.

```
#include <Wire.h>
```

با اضافه کردن این کتابخانه رابط i2c روی برد ESP32 فعال میشود. این کتابخانه به صورت پیشفرض SDA را پایه 21 و SCK را پایه 22 در نظر میگیرد.

```
Wire.begin(21, 22); // SDA = GPIO21, SCL = GPIO22
```

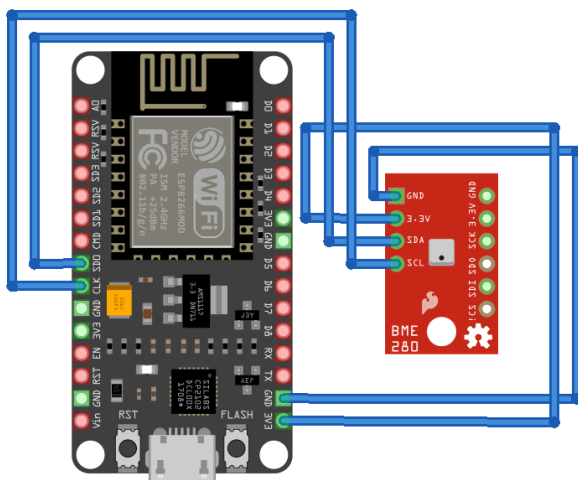
\*البته با این خط کد میتوان پایه های پیشفرض را به پایه های دلخواه تغییر داد.

```
#include <Adafruit_BME280.h>
```

\*با اضافه کردن این کتابخانه از Adafruit، از سینتکس های مربوط به این ماژول میتوانیم استفاده کنیم.

در ادامه اتصالات لازم را برای راه اندازی سنسور با ESP32

مشاهده مینمایید.



fritzing

```
float temp = bme.readTemperature();  
float pressure = bme.readPressure() / 100.0F;  
float humidity = bme.readHumidity();  
float altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
```

همچنین لازم به ذکر است که برای خواندن مقادیر ذکر شده از سنسور، از این دستورات در برنامه نویسی استفاده شده است.

## معرفی نمایشگر oled با رابط i2c :

نمایشگر های OLED ، نمایشگر هایی با کنتراست و رزولوشن بالا می باشند، از این رو قابلیت خوانایی زیادی را برای کاربر فراهم می کنند. این نمایشگرها نیاز به نور پس زمینه Backlight ندارند و پیکسل ها خودشان نور افشانی می کنند و در مقایسه با LCD های کریستال مایع، باریکتر و شفاف تر بوده و کنتراست بالاتر و مصرف توان کمتری دارند. چیپ درایور این ماژول SSD1306 است که توانایی ارتباط I2C را برای این ماژول فراهم می آورد.

SSD1306 یک ماژول درایور برای صفحه نمایش سیستم های نقطه ماتریسی می باشد و برای صفحه نمایش نوع کاتد مشترک استفاده می شود و در تجهیزات قابل حمل مانند تلفن موبایل، MP3 Player، ماشین حساب و کاربردهای مختلف دیگر به کار گرفته می شود.



این ماژول هم از ارتباط i2c استفاده میکند .

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

\*کتابخانه های مربوط به نمایشگر oled

در نمایشگر های oled نیز میتوان آیکن های گرافیکی را نیز نمایش داد. در ادامه با چگونگی انجام این کار آشنا میشویم:

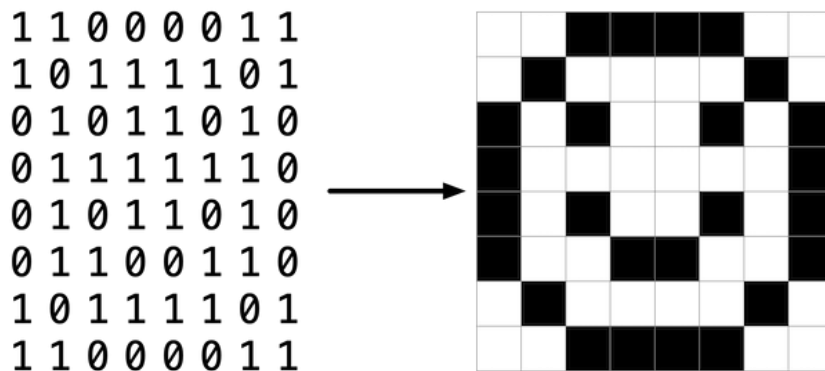
```
const unsigned char icon_temp[] PROGMEM = {
  0b00001100, 0b00000000,
  0b00011110, 0b00000000,
  0b00111111, 0b00000000,
  0b01111111, 0b10000000,
  0b01111111, 0b10000000,
  0b01111111, 0b10000000,
  0b01111111, 0b10000000,
  0b01111111, 0b10000000,
  0b00111111, 0b00000000,
  0b00111111, 0b00000000,
  0b00111111, 0b00000000,
  0b00011110, 0b00000000,
  0b00001100, 0b00000000,
  0b00001100, 0b00000000,
  0b00001100, 0b00000000,
```



```
0b00001100, 0b00000000,  
0b00001100, 0b00000000  
};
```

\*تیکه خط کد بالا برای نمایش آیکون دماسنج روی LCD است.

برای نمایش یک آیکون (مثلاً نماد دما) روی نمایشگرهای OLED در پروژه‌های آردوینو، از تصویرهای سیاه و سفید به صورت آرایه‌ای از بیت‌ها (bitmap) استفاده می‌شود. مراحل کار شامل آماده‌سازی تصویر، تبدیل آن به آرایه باینری، و سپس نمایش آن با استفاده از کتابخانه‌هاست.



در ابتدا، تصویری که می‌خواهیم نمایش دهیم (مثلاً آیکون دماسنج)، باید به صورت تصویر سیاه و سفید و با ابعاد مشخص مثلاً (16x16) پیکسل تهیه شود.

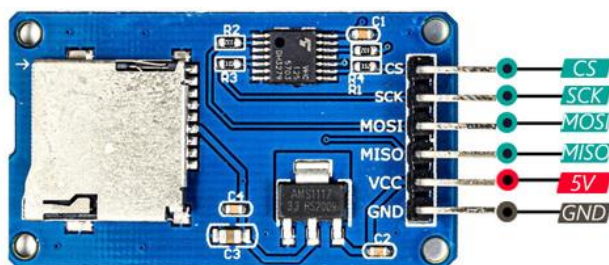
رنگ سیاه → روشن بودن پیکسل (بیت ۱)

رنگ سفید → خاموش بودن پیکسل (بیت ۰)

نمایشگرهای OLED فقط پیکسل را روشن یا خاموش می‌کنند، پس تصویر باید ۱ بیتی (black & white) باشد.

## معرفی ماژول SD card reader :

معمولا ذخیره کردن اطلاعات، یکی از اصلی ترین بخش های یک پروژه است. باتوجه به نوع اطلاعات و حجم آن ها، راه ها و وسایل مختلفی برای ذخیره ی آن ها وجود دارد. یکی از متداول ترین وسایل ذخیره ی اطلاعات، کارت های حافظه ی SD و micro SD است که در وسایلی مثل گوشی های تلفن همراه، مینی کامپیوتر ها و ... استفاده می شود. ماژول های SD و micro SD به شما اجازه می دهند تا با کارت حافظه ارتباط برقرار کنید و روی آنها اطلاعاتی را بنویسید و یا از آن ها اطلاعات بخوانید. رابط کاربری این ماژول ها SPI است برای راه اندازی این ماژول ها با برد های آردوینو از کتابخانه ی SD استفاده می شود. این کتابخانه به صورت پیش فرض روی نرم افزار آردوینو وجود دارد.



VCC = تغذیه ماژول - 5 ولت

GND = زمین

CS = پایه ChipSelect

MISO = پایه فرستنده - معادل پین Tx پورت سریال

MOSI = پایه گیرنده - معادل پین Rx پورت سریال

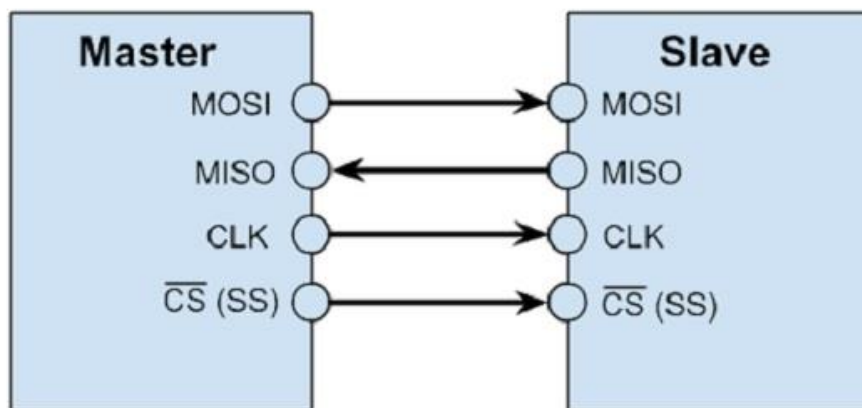
SCK = پایه کلاک ماژول در این پروژه ما از این ماژول برای ذخیره سازی دیتاها استفاده میکنیم. فایل ذخیره سازی شده را

بعدا وارد excel میکنیم و نمودار دیتاها رسم میشود.

Description	Command
Initializes the SD library and card. Enter the pin connected to the SS pin as a function's argument.	SD.begin(#sspin)
Tests whether a file or directory exists on the SD card.	SD.exists(filename)
Opens a file on the SD card in reading or writing mode. (If you leave the mode section blank, the file will open in reading mode by default) If the file is opened for writing, it will be created a file with this name if it doesn't already exist.	SD.open(filepath, mode)
Close the file and ensure that any data written to it is physically saved to the SD card.	file.close()*
Remove a file from the SD card.	SD.remove(filename)
Create a directory on the SD card	SD.mkdir(filename)
Remove a directory from the SD card.	SD.rmdir(filename)
Returns the file name	file.name()*
Print data to the file	file.print(data)
Print data, followed by a carriage return and newline	file.println(data)
Read from the file.	file.read()
Check if there are any bytes available for reading from the file.	file.available()

## ارتباط spi :

همانطور که توضیح داده شد ماژول از رابط spi استفاده میکند.



گذرگاه SPI با ۴ سیگنال عمل می کند:

**SCLK (Serial Clock):** کلاک سریال است که

یکی از سیگنال های خروجی دستگاه master به آن اختصاص داده می شود.

**MOSI (Master Out Slave In):** سیگنال خروجی داده از دستگاه master است.

MOSI در یک دستگاه master به MOSI در یک دستگاه slave متصل می شود.

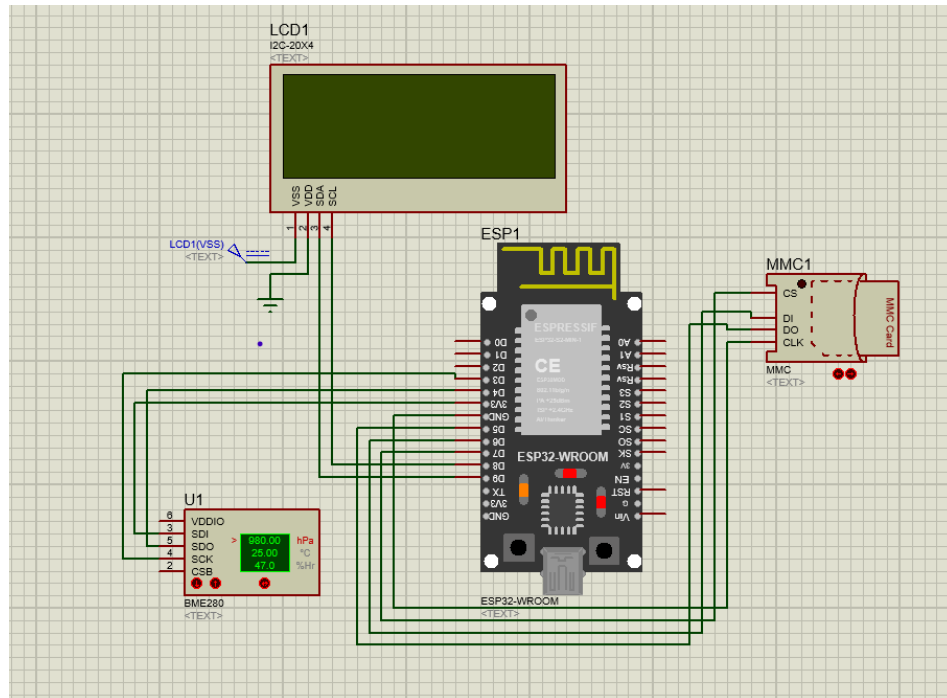
**MISO (Master In Slave Out):** سیگنال خروجی داده از دستگاه slave است.

MISO در یک دستگاه master به MISO در یک دستگاه slave متصل می شود.

**SS (Slave Select):** سیگنال خروجی از دستگاه master است.

Slave Select عملکرد مشابه تراشه را دارد و برای مفهوم آدرس دهی استفاده می شود.

## پیاده سازی نهایی پروژه در پروتئوس:



\* کتابخانه های استفاده شده در پروژه به صورت پیشفرض در پروتئوس موجود نمیباشند.

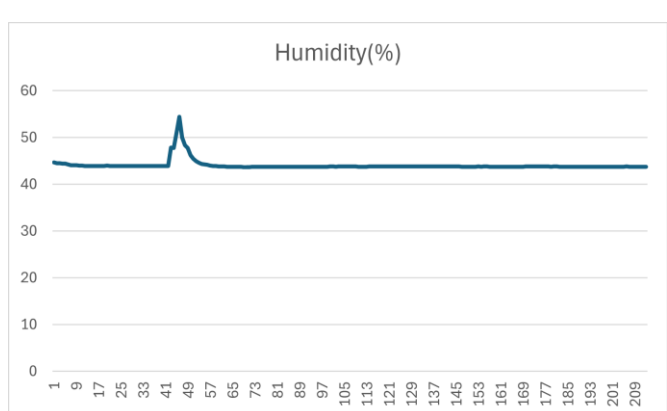
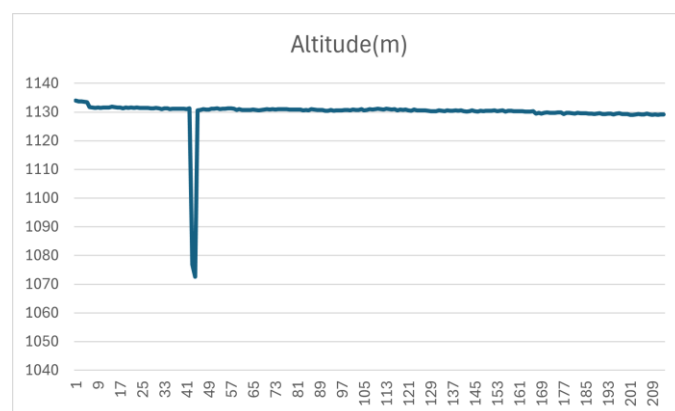
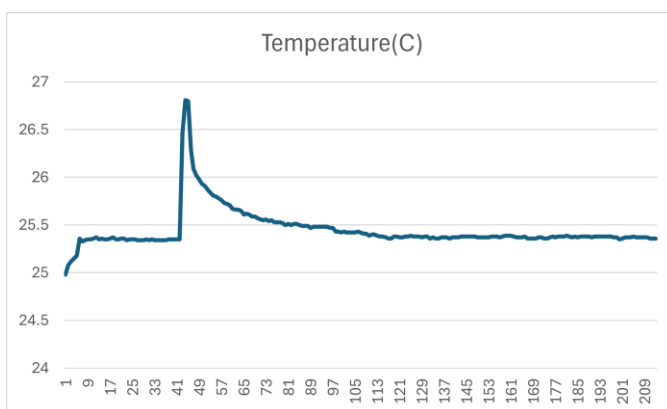
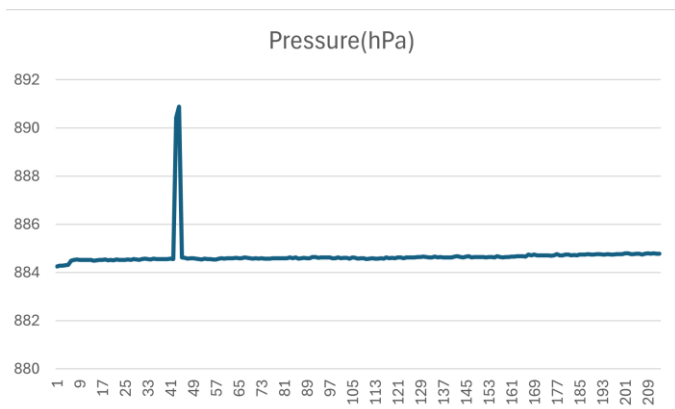
```
File dataFile = SD.open("/data.csv", FILE_APPEND);
if (dataFile) {
    dataFile.printf("%lu,%.2f,%.2f,%.2f,%.2f\n", millis(),
temperature, humidity, pressure, altitude);
    dataFile.close();
} else {
    Serial.println("Error opening data.csv");
}
```

ما در این پروژه با کد نوشته شده در بالا دیتا های گرفته شده از سمت BME280 را با فرمت CSV ذخیره سازی میکنیم.

دیتاهای ذخیره شده در اس دی کارت به صورت فایل EXEL قابل خواندن از سوی کاربر میباشد که میتوان نمودار تغییرات این دیتاها را نیز رسم نمود. برای تست، نمودار دیتاهای ذخیره شده برای 15 دقیقه در ادامه آورده شده است.

\* لازم به ذکر است که فرمت کارت SD باید روی FAT32 تنظیم شده باشد.

قدم آخر:



\*جداول رسم شده در EXEL

## اطلاعات محیطی

°C دما: 25.32

% رطوبت: 43.60

hPa فشار: 885.08

m ارتفاع: 1126.34

بهر روز رسانی دستی



نمایش دیتاها روی LCD oled

نمایش دیتاها روی وب سرور

## اهداف پروژه:

- دستگاه طراحی شده یک سیستم هوشمند اندازه‌گیری و پایش محیطی است که قابلیت‌های زیر را دارد:
- ماژول **ESP32**: مغز اصلی سیستم، مسئول خواندن داده‌ها، ذخیره‌سازی و نمایش آن‌ها روی وب‌سرور.
- سنسور **BME280**: برای اندازه‌گیری دقیق دما (Temperature) / رطوبت (Humidity) / فشار هوا (Pressure) / ارتفاع تقریبی از سطح دریا (Altitude)
- ماژول **SD Card**: برای ذخیره‌سازی بلندمدت داده‌ها به صورت فایل CSV.
- **WiFi Server** داخلی: برای نمایش داده‌ها به صورت زنده روی یک صفحه وب.

## کاربردهای پروژه:

۱. پایش شرایط محیطی در گلخانه‌ها، اتاق‌های سرور یا آزمایشگاه‌ها
۲. مانیتورینگ از راه دور با دسترسی به داده‌ها از طریق WiFi
۳. ذخیره‌سازی داده‌ها برای تحلیل‌های آماری و ترسیم نمودارهای بلندمدت
۴. آموزش و پژوهش در زمینه اینترنت اشیا (IoT)، سنسورها و میکروکنترلرها
۵. پایه‌ای برای توسعه سیستم‌های هوشمند مانند هشدار تغییرات شدید محیطی