

بسم الله الرحمن الرحيم

دانشگاه امام خمینی

گروه الکترونیک

گزارش پروژه پایانی

عنوان پروژه:

طراحی و پیاده‌سازی سامانه اندازه‌گیری دما و رطوبت با نمایش
تحت وب با استفاده از ماژول ESP8266

استاد راهنما: مهندس مهدی صادقی نیا

تهیه کننده: ابوالفضل رسالتی فر

بهار 1404

فهرست مطالب

1. مقدمه	6. نتیجه گیری
2. معرفی سخت افزار	7. منابع
2.1. ماژول ESP8266 NodeMCU	
2.2. سنسور دما و رطوبت AHT10	
2.3. نمایشگر LCD 16*2 با رابط i2c	
2.4. شماتیک پروژه	
3. معرفی نرم افزار.	
3.1. زبان برنامه نویسی و کتابخانه ها	
3.2. ساختار فایل در حافظه SPIFFS	
4. تحلیل بخش های اصلی کد	
4.1. بررسی توابع SPIFFS	
4.2. بررسی توابع Server	
5. فلوچارت برنامه	

مقدمه:

در دنیای امروز، جمع‌آوری و نمایش اطلاعات محیطی مانند دما و رطوبت به یکی از نیازهای اساسی در بسیاری از پروژه‌های دانشجویی و صنعتی تبدیل شده است.

این پروژه با هدف طراحی و پیاده‌سازی یک سامانه ساده برای اندازه‌گیری و نمایش دما و رطوبت محیط اجرا شده است. در این سامانه از ماژول ESP8266 NodeMCU به عنوان هسته‌ی پردازشی، سنسور AHT10 برای اندازه‌گیری دما و رطوبت، و نمایشگر LCD 16x2 برای نمایش اطلاعات به صورت محلی استفاده شده است. همچنین یک رابط کاربری مبتنی بر وب برای مشاهده‌ی مقادیر اندازه‌گیری‌شده و مدیریت تنظیمات دستگاه طراحی گردیده است. این پروژه تلاش دارد تا با ترکیب سخت‌افزار ساده و نرم‌افزاری کارآمد، تجربه‌ای مناسب از طراحی و توسعه‌ی سیستم‌های مانیتورینگ را ارائه دهد.

هدف پروژه:

هدف از اجرای این پروژه، طراحی و ساخت یک سامانه‌ی ساده برای اندازه‌گیری و نمایش دما و رطوبت محیط می‌باشد. در این سامانه اطلاعات محیطی به صورت محلی روی یک نمایشگر LCD نمایش داده می‌شود و همچنین از طریق یک رابط کاربری تحت وب قابل مشاهده و مدیریت است. این پروژه به گونه‌ای طراحی شده که کاربر بتواند به سادگی به داده‌های محیطی دسترسی داشته باشد و در صورت نیاز، تنظیمات سیستم مانند نام کاربری و رمز عبور را تغییر دهد.

در ادامه به بررسی دقیق این پروژه خواهیم پرداخت. ابتدا بخش سخت‌افزار معرفی می‌شود و اجزای مورد استفاده توضیح داده می‌شود. سپس در بخش نرم‌افزار، نحوه‌ی برنامه‌نویسی و عملکرد سیستم تشریح خواهد شد.

سخت افزار:

1. ESP8266 NodeMCU

ماژول ESP8266 NodeMCU یک برد میکروکنترلر مبتنی بر تراشه ESP8266 است که به دلیل قابلیت اتصال به WiFi و قیمت مقرون به صرفه، برای پروژه‌های شبکه‌ای و مانیتورینگ محیطی بسیار مناسب است. این ماژول به عنوان یک پلتفرم توسعه محبوب شناخته می‌شود و با محیط Arduino سازگار است.

مشخصات:

پردازنده: Tensilica L106 32-bit با فرکانس 80 تا 160 مگاهرتز.

حافظه:

4 مگابایت حافظه فلش برای ذخیره برنامه و داده‌ها.

128 کیلوبایت RAM.

اتصالات:

پورت های GPIO (ورودی و خروجی دیجیتال).

رابط‌های I2C ، SPI و UART برای ارتباط با سنسورها و ماژول‌ها.

پشتیبانی از ADC (مبدل آنالوگ به دیجیتال) تک کاناله.

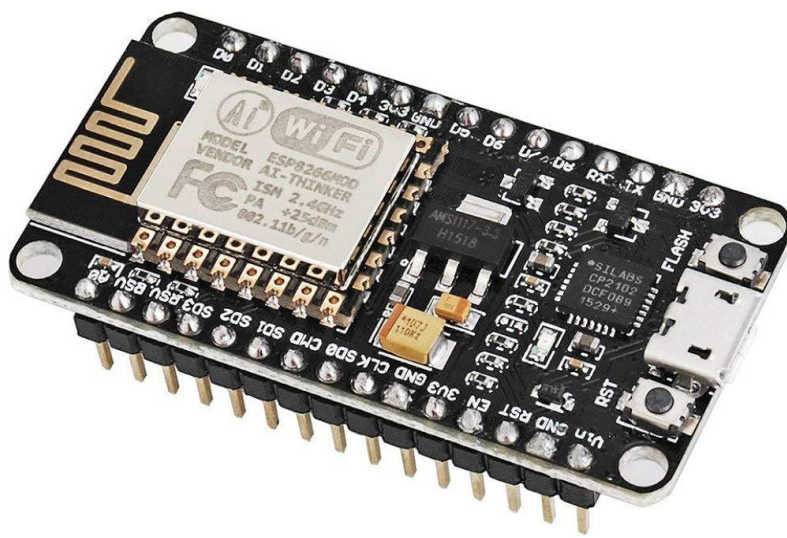
اتصال شبکه:

ماژول WiFi داخلی برای اتصال به شبکه‌های بی سیم.

قابلیت عمل به عنوان Access Point یا Station.

نقش در پروژه :

هسته مرکزی سیستم: اجرای کد پروژه، پردازش داده‌ها و مدیریت ارتباطات.
ارتباط با سخت‌افزار: مدیریت سنسور AHT10 و نمایشگر LCD از طریق رابط I2C.
میزبانی وب‌سرور: ارائه رابط کاربری وب برای نمایش داده‌های دما و رطوبت و مدیریت تنظیمات.



در این پروژه از برد NodeMCU استفاده خواهیم کرد.
این برد به خاطر ابعاد مناسب، گزینه‌ی بسیار خوبی برای نمونه‌سازی و پروتوتایپ پروژه‌ها به حساب می‌آید.
همچنین به دلیل وجود مبدل داخلی USB به Serial ، اتصال آن به کامپیوتر بسیار ساده است و به راحتی می‌توان آن را برنامه‌ریزی کرد.

2. AH10 Sensor

سنسور دما و رطوبت AHT10:

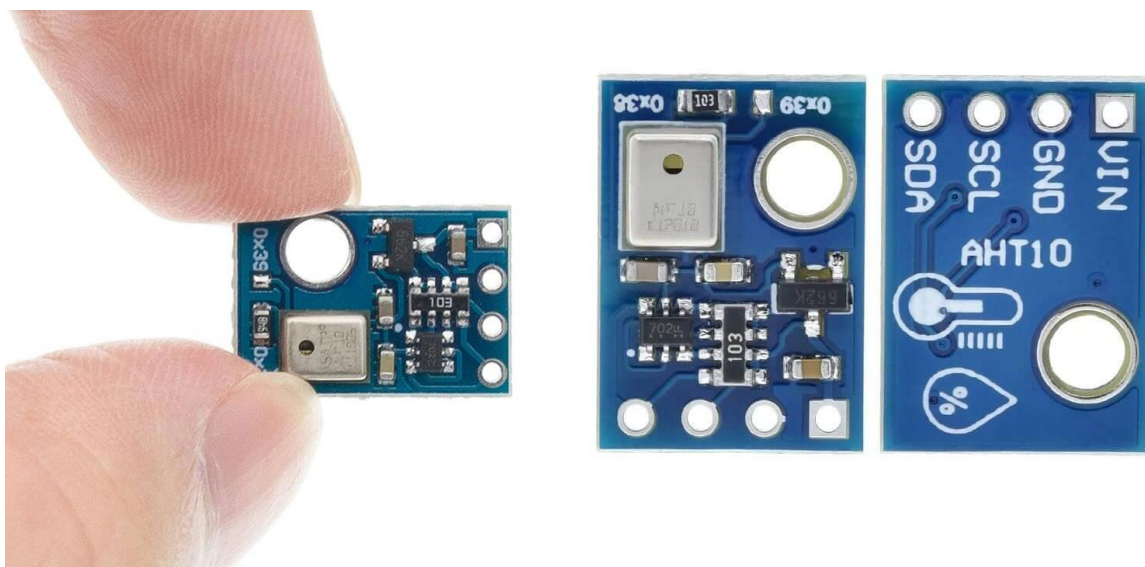
در این پروژه برای اندازه‌گیری دما و رطوبت محیط، از سنسور AHT10 استفاده شده است.

این سنسور به صورت دیجیتال عمل کرده و از پروتکل ارتباطی I2C برای تبادل داده با میکروکنترلر بهره می‌برد.

ولتاژ کاری AHT10 در بازه 2.2 تا 5.5 ولت قرار دارد، بنابراین می‌توان آن را مستقیماً به بردهای رایجی مانند NodeMCU متصل کرد.

بازه اندازه‌گیری دما در این سنسور بین 40- تا 85+ درجه‌ی سانتی‌گراد و بازه اندازه‌گیری رطوبت بین 0 تا 100 درصد رطوبت نسبی است.

دقت مناسب، ابعاد کوچک، مصرف انرژی پایین و سادگی در راه‌اندازی باعث شده است AHT10 به یک انتخاب محبوب در پروژه‌های دانشجویی و صنعتی تبدیل شود.



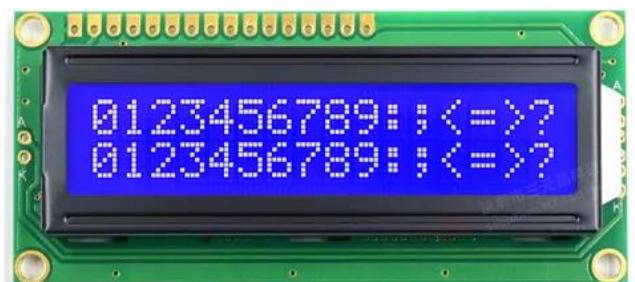
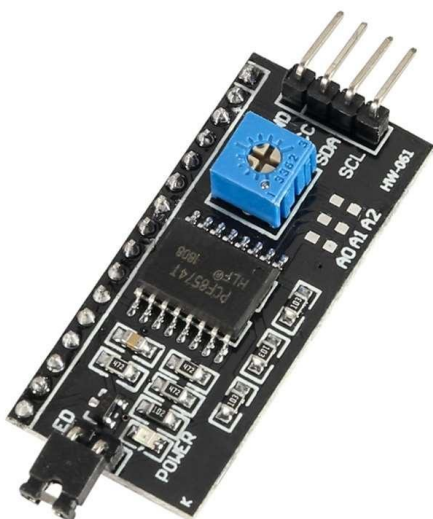
3. Character LCD Display + PCF8574 Module

نمایشگر LCD کاراکتری به همراه ماژول PCF8574:

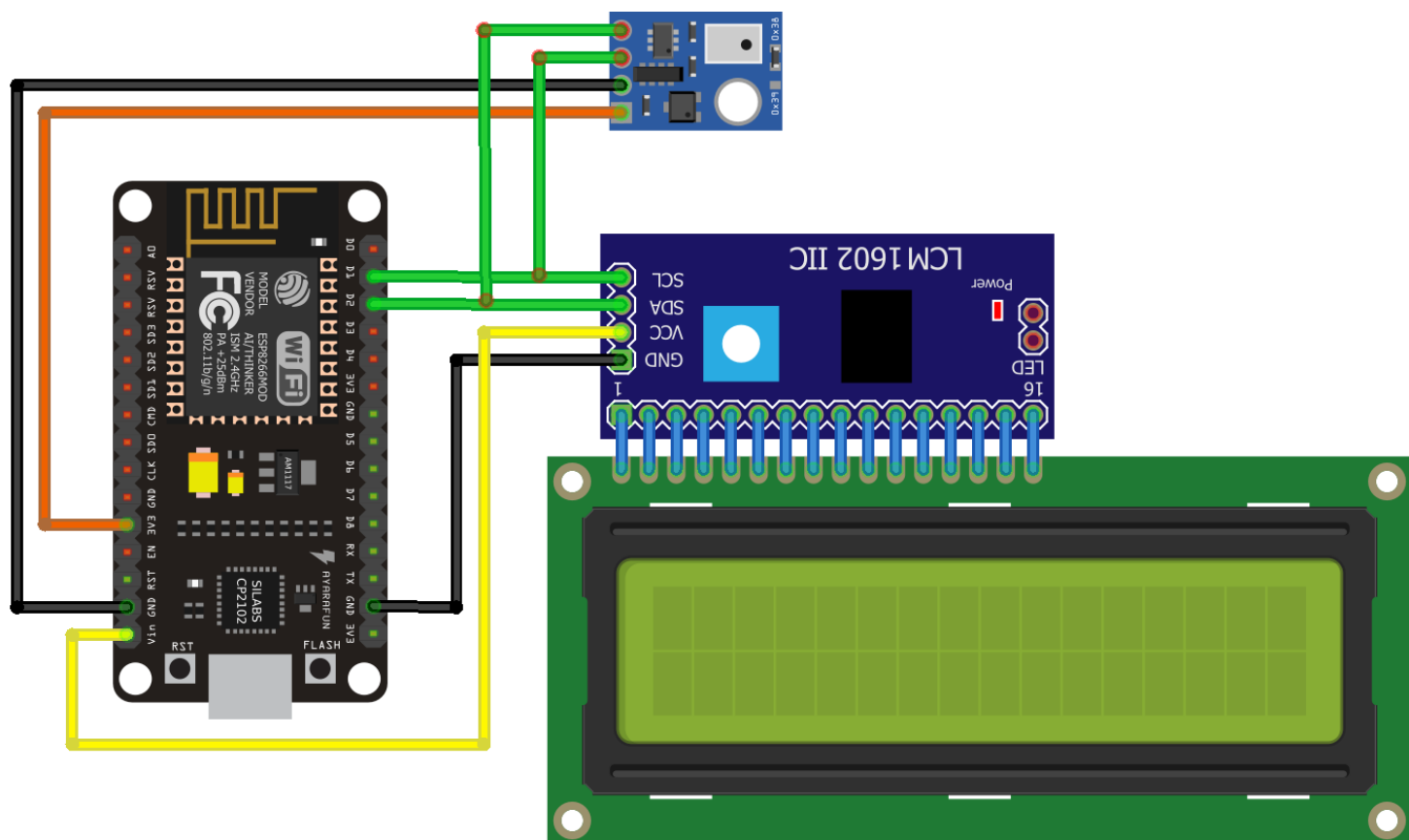
در این پروژه برای نمایش مقادیر دما و رطوبت، از یک نمایشگر LCD کاراکتری با آرایش 16×2 استفاده شده است.

این نمایشگر به همراه یک ماژول I2C مبتنی بر تراشه PCF8574 استفاده می‌شود تا ارتباط با میکروکنترلر از طریق پروتکل I2C انجام شود. استفاده از ماژول I2C تعداد پین‌های مورد نیاز برای اتصال را به‌طور قابل‌توجهی کاهش داده و مدیریت نمایشگر را در پروژه‌های الکترونیکی ساده‌تر می‌کند. این ترکیب برای نمایش اطلاعات متنی در پروژه‌های مانیتورینگ و کنترل بسیار مناسب است.

تراشه PCF8574 یک گسترش‌دهنده ورودی/خروجی (I/O Expander) که ارتباط موازی LCD را به پروتکل I2C تبدیل می‌کند.



شماتیک و اتصالات پروژه:



fritzing

نرم افزار:

معرفی محیط Arduino IDE و دلایل استفاده:



برای توسعه نرم افزار این پروژه، از محیط برنامه نویسی **Arduino IDE** استفاده شده است. **Arduino IDE** یک محیط توسعه ی یکپارچه (IDE) ساده و کاربرپسند است که امکان برنامه نویسی سریع و آسان انواع میکروکنترلرها، از جمله ماژول های مبتنی بر ESP8266، را فراهم می کند.

دلایل انتخاب **Arduino IDE** در این پروژه عبارت اند از:

رابط کاربری ساده: مناسب برای انجام پروژه های سریع و آموزشی

کتابخانه های گسترده: وجود کتابخانه های آماده برای کار با سنسور AHT10 ،

LCD کاراکتری و ارتباط I2C

پشتیبانی از ماژول NodeMCU: پیکربندی آسان برای آپلود مستقیم کد روی ماژول

بدون نیاز به ابزار اضافی

جامعه کاربری فعال: وجود منابع زیاد برای رفع اشکال و توسعه پروژه

مراحل آماده‌سازی نرم‌افزار:

نصب Arduino IDE: ابتدا آخرین نسخه Arduino IDE از سایت رسمی دانلود و نصب می‌شود.

افزودن برد ESP8266: با افزودن لینک مربوطه در بخش Boards Manager ، پشتیبانی از ماژول NodeMCU به Arduino IDE اضافه می‌شود.

نصب کتابخانه‌های مورد نیاز:

کتابخانه‌ی مخصوص سنسور AHT10

کتابخانه‌ی LiquidCrystal_I2C برای کنترل نمایشگر کاراکتری با ماژول I2C

نوشتن برنامه: کدنویسی شامل راه‌اندازی ماژول‌ها، خواندن داده‌های دما و رطوبت، و نمایش آن‌ها بر روی LCD است.

آپلود و تست: پس از بررسی و اطمینان از صحت کد، برنامه توسط کابل USB روی ماژول NodeMCU آپلود شده و عملکرد آن تست می‌شود.

ساختار کلی برنامه:

برنامه نوشته شده شامل بخش‌های زیر است:

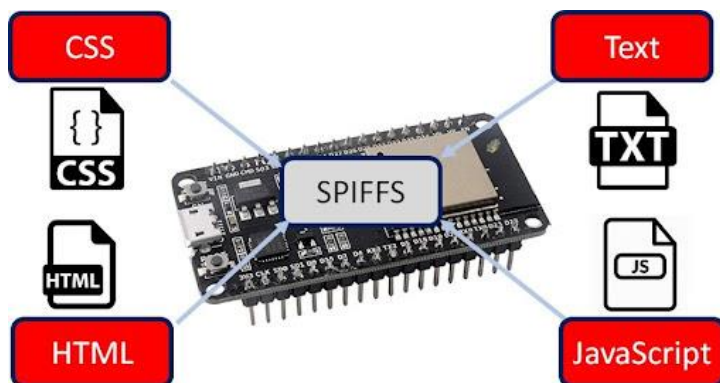
- تعریف کتابخانه‌ها و متغیرها: وارد کردن کتابخانه‌های مورد نیاز و تعریف پایه‌های ارتباطی
- راه‌اندازی اولیه: (setup) مقداردهی اولیه سنسور، LCD
- حلقه اصلی: (loop) خواندن داده‌های دما و رطوبت از سنسور و به‌روزرسانی نمایشگر LCD به صورت دوره‌ای

معرفی سیستم فایل SPIFFS:

در این پروژه از قابلیت ¹SPIFFS استفاده شده است. SPIFFS یک سیستم فایل ساده و سبک است که به منظور ذخیره‌سازی داده‌ها در حافظه‌ی فلش داخلی ماژول‌هایی مانند ESP8266 و ESP32 طراحی شده است. این سیستم فایل امکان ذخیره‌ی فایل‌های متنی، صفحات HTML، تصاویر و سایر داده‌های مورد نیاز پروژه را مستقیماً بر روی حافظه‌ی داخلی ماژول فراهم می‌کند.

دلایل استفاده از SPIFFS در پروژه:

- عدم نیاز به حافظه‌ی خارجی: بدون نیاز به کارت حافظه یا سخت‌افزار اضافی، می‌توان فایل‌های لازم را در حافظه داخلی ذخیره کرد.
- دسترسی سریع و آسان به فایل‌ها: خواندن و نوشتن فایل‌ها از طریق دستورات ساده‌ی برنامه‌نویسی انجام می‌شود.
- حفظ داده‌ها پس از ریست: داده‌های ذخیره شده در SPIFFS حتی پس از قطع برق یا ریست شدن ماژول از بین نمی‌روند.
- امکان ذخیره تنظیمات یا قالب‌های نمایشی: در پروژه‌هایی که نیاز به نگهداری اطلاعات ثابت یا تغییرپذیر مانند داده‌های کالیبراسیون، صفحات وب یا مقادیر پیکربندی وجود دارد، SPIFFS بسیار مفید است.

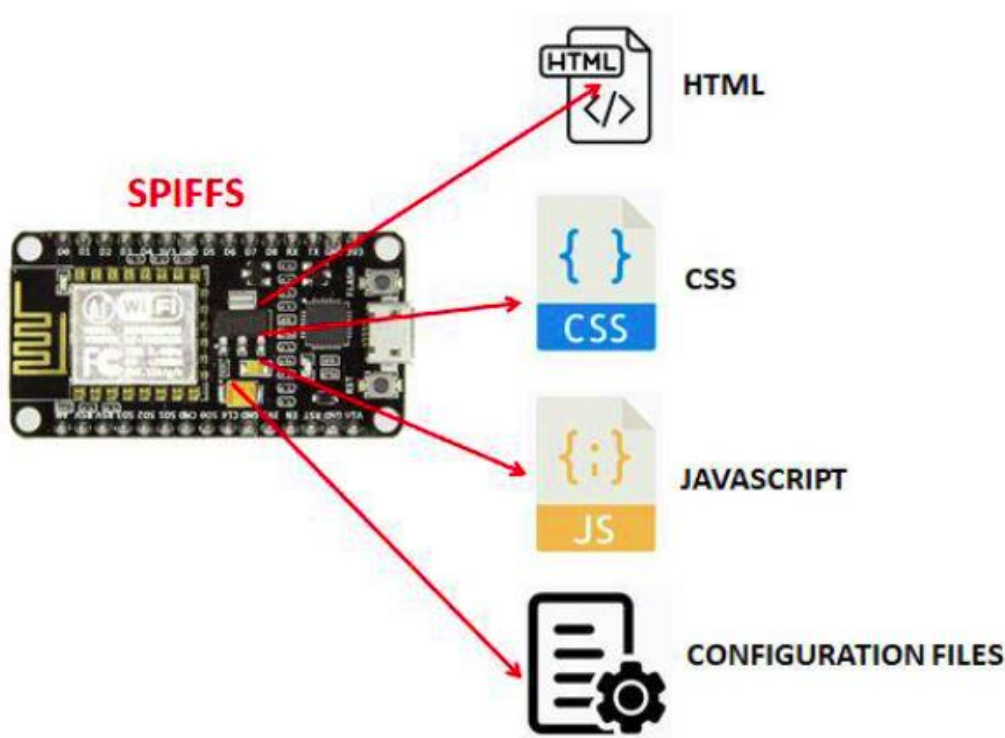


¹ Serial Peripheral Interface Flash File System

نحوه استفاده از SPIFFS در پروژه:

در این پروژه، از SPIFFS برای ذخیره فایل‌های مربوط به نمایش محتوا (فایل‌های HTML و CSS و ...) استفاده شده است. قبل از شروع کار با این سیستم فایل، مراحل زیر انجام می‌شود:

1. **فرمت و آماده‌سازی حافظه فلش: یک بار حافظه‌ی ماژول برای استفاده از SPIFFS فرمت می‌شود.**
2. **آپلود فایل‌ها: فایل‌های مورد نیاز توسط افزونه‌ی مخصوص Arduino IDE به حافظه ماژول آپلود می‌شوند.**
3. **دسترسی به فایل‌ها: برنامه‌ی نوشته شده با استفاده از توابع مخصوص، فایل‌ها را از حافظه‌ی داخلی خوانده و مورد استفاده قرار می‌دهد.**



بررسی کد پروژه:

در ابتدا کد مربوط به ESP8266 بررسی خواهد شد.

در این کد، ابتدا ماژول **ESP8266** به عنوان یک **Access Point** راه اندازی می شود و یک وب سرور محلی ایجاد می کند. داده های صفحات وب از حافظه داخلی (SPIFFS) خوانده شده و به کاربران ارائه می شود. کد شامل یک **سامانه احراز هویت** است که اطلاعات ورود (نام کاربری و رمز عبور) را بررسی می کند و در صورت تأیید، کاربر را به داشبورد هدایت می کند. در داشبورد، داده های خوانده شده از سنسور **AHT10** (دما و رطوبت) نمایش داده می شود. همچنین امکان تغییر نام کاربری و رمز عبور از طریق صفحه تنظیمات فراهم شده و این اطلاعات در حافظه ذخیره می گردد. نمایشگر **LCD 16x2** نیز وضعیت اتصال و اطلاعات سرور را نمایش می دهد.

تعریف کتابخانه ها و متغیرهای اولیه

در ابتدای کد، چندین کتابخانه ی مهم اضافه شده اند که هر کدام وظیفه ی مشخصی دارند:

```
1. #include <ESP8266WiFi.h>
2. #include <ESP8266WebServer.h>
3. #include <Adafruit_AHTX0.h>
4. #include <LiquidCrystal_I2C.h>
5. #include "FS.h"
```

ESP8266WiFi.h: این کتابخانه امکان راه اندازی ماژول Wi-Fi داخلی ESP8266 و مدیریت اتصال شبکه را فراهم می کند.

ESP8266WebServer.h: برای ایجاد و مدیریت یک وب سرور ساده روی ماژول استفاده شده است.

Adafruit_AHTX0.h: جهت ارتباط و خواندن داده‌های سنسور دما و رطوبت مدل AHT20/AHT10 به کار می‌رود.

LiquidCrystal_I2C.h: برای راه‌اندازی نمایشگر LCD کاراکتری با استفاده از ماژول I2C به کار گرفته شده است.

FS.h: برای کار با فایل‌های ذخیره‌شده در سیستم فایل داخلی ESP8266 یعنی (SPIFFS) استفاده می‌شود.

پس از معرفی کتابخانه‌ها، متغیرهای زیر تعریف شده‌اند:

```
IPAddress local_ip(192,168,1,1);
IPAddress gateway(192,168,1,1);
IPAddress subnet(255,255,255,0);

Adafruit_AHTX0 aht;

ESP8266WebServer server(80);
LiquidCrystal_I2C lcd(0x3F, 16, 2);

String savedUsername = "admin";
String savedPassword = "1234";
bool isAuthenticated = false;

float temperatureVal = 0.00;
float humidityVal = 0.00;
```

local_ip ، **gateway** ، **subnet**: برای تعیین IP شبکه‌ی محلی ماژول در حالت Access Point.

aht: یک شی از کلاس Adafruit_AHTX0 برای کنترل سنسور دما و رطوبت.

server: شی اصلی وب‌سرور که روی پورت ۸۰ (HTTP) راه‌اندازی می‌شود.

lcd: شی مربوط به نمایشگر LCD کاراکتری با آدرس I2C 0x3F و اندازه ۲×۱۶ کاراکتر.

savedUsername و **savedPassword**: برای نگهداری اطلاعات کاربری جاری (نام کاربری و رمز عبور).

`isAuthenticated`: یک متغیر بولین که نشان می‌دهد کاربر احراز هویت شده است یا خیر.

`humidityVal` و `temperatureVal`: متغیرهای شناور برای نگهداری آخرین مقادیر خوانده شده از سنسور.

توابع SPIFFS:

1. `SPIFFS.begin()`

برای mount کردن سیستم فایل SPIFFS روی حافظه فلش. این خط باید در `setup()` فراخوانی بشه تا فایل‌های ذخیره‌شده در حافظه قابل دسترسی باشن.

```
1. if (!SPIFFS.begin()) {  
2.     Serial.println("SPIFFS Mount Failed!");  
3. }
```

1. `SPIFFS.open(path, mode)`

باز کردن یک فایل در مسیر مشخص و با حالت مشخص. دو حالت معمول:

r : فقط خواندن (read-only)

w : فقط نوشتن (write-only, overwrites)

```
1. File file = SPIFFS.open("/config.txt", "r");
```

1. file.readStringUntil('\n')

خوندن محتوای فایل تا رسیدن به `\n` (انتهای خط).

```
1. savedUsername = file.readStringUntil('\n');
```

1. file.available()

بررسی می‌کند که آیا هنوز داده‌ای برای خواندن باقی مانده یا نه.

```
1. while (file.available()) {  
2.     html += (char)file.read();  
3. }
```

1. file.read()

خواندن بایت به بایت از فایل.

1. file.close()

بستن فایل بعد از خواندن یا نوشتن. خیلی مهمه چون باعث آزادسازی منابع میشه.

توابع Web Server:

1. `server.on(path, method, handler)`

برای تعیین کاری که سرور باید هنگام دریافت درخواست در مسیر خاص انجام دهد.

1. `server.send(code, type, content)`

برای ارسال پاسخ HTTP به مرورگر.

1. `server.arg(name)`

گرفتن مقدار آرگومان POST یا GET که از فرم یا URL ارسال شده.

1. `server.begin()`

شروع به کار وب سرور.

1. `server.handleClient()`

این تابع باید در حلقه `loop()` فراخوانی شود تا سرور درخواست‌های دریافتی را پردازش کند.

فلوچارت برنامه:

شروع Setup:

- فعال سازی سریال
- راه اندازی نمایشگر
- راه اندازی WiFi Access Point
- راه اندازی SPIFFS
- راه اندازی سنسور AHT10
- بارگذاری نام کاربری و رمز عبور
- تعریف مسیر های سرور
- شروع وب سرور

حلقه Loop:

- بررسی درخواست های HTTP
- خواندن دما و رطوبت از سنسور
- ذخیره در متغیر ها
- نمایش روی نمایشگر

نتیجه گیری:

در این پروژه، یک سامانه کنترل و نظارت از راه دور طراحی و پیاده‌سازی شد که با استفاده از ماژول ESP8266 به‌عنوان هسته‌ی پردازشی و ارتباطی، امکان اندازه‌گیری دما و رطوبت محیط، نمایش اطلاعات روی LCD و همچنین مشاهده اطلاعات در یک وب‌سرور داخلی فراهم شد. همچنین، سیستم به نحوی طراحی شد که امکان ورود با نام کاربری و رمز عبور، تغییر رمز عبور از طریق صفحه تنظیمات و حفاظت از صفحات داخلی با احراز هویت نیز در آن گنجانده شده است.

از جمله مهم‌ترین دستاوردهای این پروژه می‌توان به موارد زیر اشاره کرد:

- استفاده از سنسور دیجیتال AHTX0 برای دریافت دقیق مقادیر دما و رطوبت
- نمایش اطلاعات روی نمایشگر LCD با رابط I2C
- طراحی رابط کاربری مبتنی بر HTML جهت نمایش داده‌ها از طریق Wi-Fi
- ذخیره‌سازی اطلاعات کاربری در حافظه‌ی SPIFFS داخلی ماژول
- پیاده‌سازی مکانیزم احراز هویت و محافظت از صفحات داخلی سرور
- امکان توسعه آتی برای کنترل رله‌ها یا تجهیزات دیگر

دریافت فایل‌های پروژه:

کدهای آردوینو و فایل‌های HTML مربوط به این پروژه در مخزن گیت‌هاب قرار داده شده‌اند. برای مشاهده و دانلود آن‌ها، به لینک زیر مراجعه کنید:

[گیت‌هاب من](#)

منابع:

<https://www.arduino.cc>

<https://randomnerdtutorials.com>

<https://github.com>

<https://openai.com/chatgpt>