

کلید هوشمند با قابلیت دریافت فرمان صوتی

گزارش فاز دوم

گروه 9

اعضای گروه:

● سید علی مرعشیان سرائی - 97102441

● محمدجواد حمزه - 971015539

● آرین اعتمادی حقیقی - 97110003

بخش اول: کنترل کردن NodeMCU به صورت بی سیم در شبکه محلی

به طور کلی، دو رویکرد برای کنترل کردن بی سیم NodeMCU (و در نتیجه کنترل کردن لامپ های هوشمند پروژه ما) وجود دارد:

1. از طریق شبکه محلی: در این روش هم خود NodeMCU و هم دستگاهی که با آن می خواهیم کنترل کردن را انجام دهیم (مانند موبایل یا کامپیوتر شخصی)، به یک مودم وصل می شوند و از آن جا با هم ارتباط برقرار می کنند.

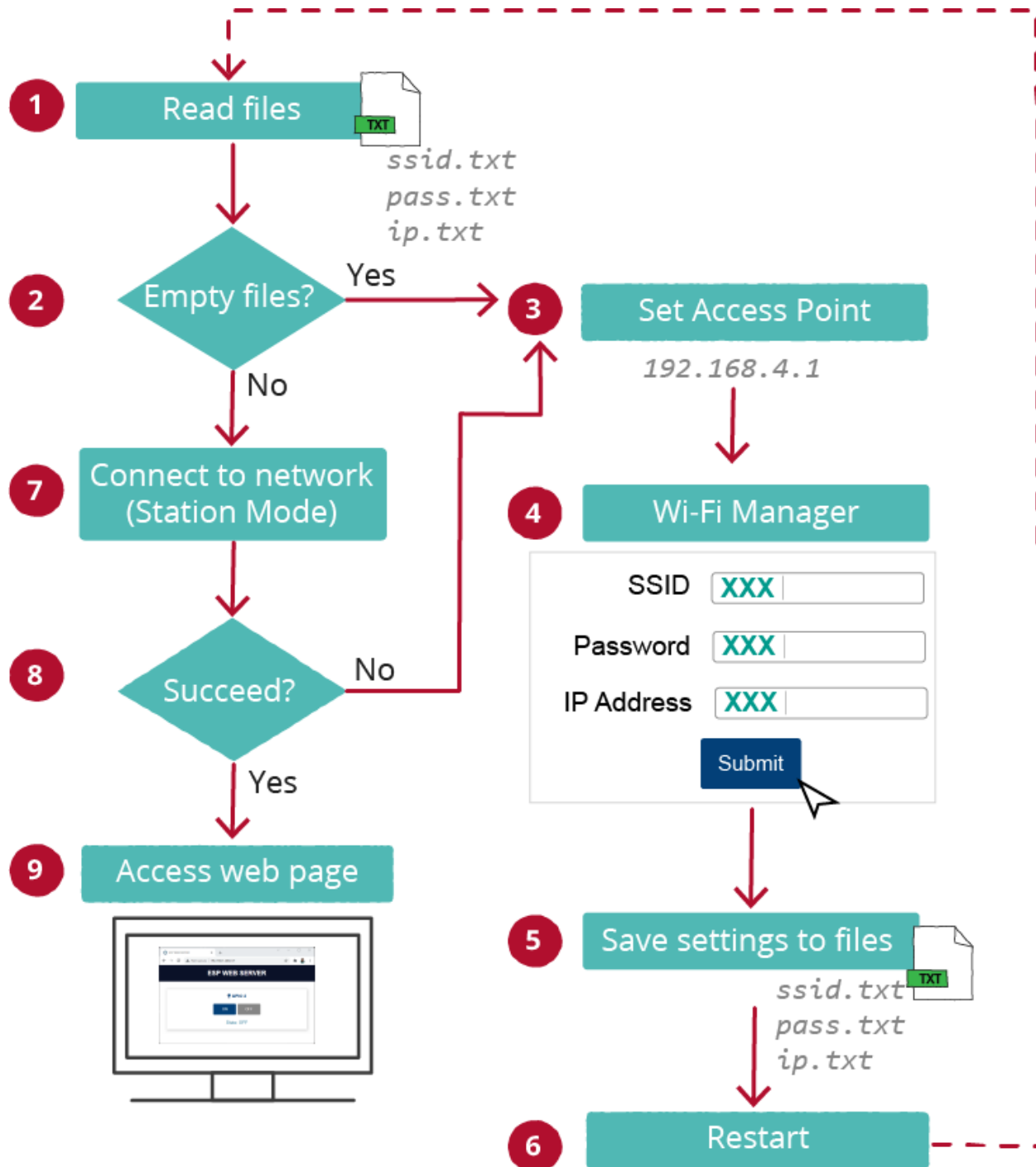
2. از طریق اینترنت و در نتیجه از هر جای جهان: در این روش NodeMCU به اینترنت وصل می شود. با تغییر تنظیمات Forwarding مودم، می توان IP Address و شماره پورتی را مشخص کرد تا مورد NodeMCU را بتوان از هر جایی در اینترنت پیدا کرد و به آن متصل شد.

در این فاز، پیاده سازی روش اول را انجام دادیم.

ابتدا دیدی کلی از نحوه راه اندازی اتصال مورد به شبکه داخلی و مراحل آن را ترسیم می کنیم:

- مورد به دنبال فایل هایی حاوی شناسه و رمز عبور مورد نیاز برای اتصال به مودم در حافظه خود می گردد. در ابتدا، هیچ کدام از این فایل ها را بر روی خود ندارد. پس صفحه ای را بالا می آوریم تا کاربر این مشخصات را وارد کند. مورد این مشخصات را در خود ذخیره کرده و Restart می شود. سپس، دوباره به دنبال فایل ها می گردد و این بار با پیدا کردن آن ها به مودم وصل می شود. پس از این می توان از مورد مانند یک سرور استفاده کرد و صفحه html اصلی پروژه (برای کنترل لامپ ها و برنامه ریزی آن ها) را روی آن قرار داد. در ادامه، کاربران باید با استفاده از دستگاه موبایل یا کامپیوتر خود به همان شبکه مودم وصل شوند، و آدرسی که مربوط به مورد است را در مرورگر خود وارد کرده، و به این صفحه html وارد شوند. کاربران از آن جا قادر خواهند بود که لامپ ها را کنترل کنند.

● در شکل زیر می توانید مراحل گفته شده را به ترتیب مشاهده کنید:



تصویر 1: مراحل دسترسی به NodeMCU از طریق شبکه محلی مودم

نحوه پیاده سازی:

پیاده سازی از طریق سه فایل زیر انجام می شود:

- **wifimanager.html**: صفحه ای است که بار اول کاربر مشخصات اتصال به مودم را در آن وارد

می کند. این صفحه هنگامی که بورد در حالت **access point** است، بالا می آید.

- **index.html**: صفحه ای است که پس از بار اول به بعد، بر روی خود بورد بالا می آید تا کاربر با

اتصال به بورد از طریق این صفحه کنترل کردن بورد (مانند روشن و خاموش کردن لامپ ها) را انجام

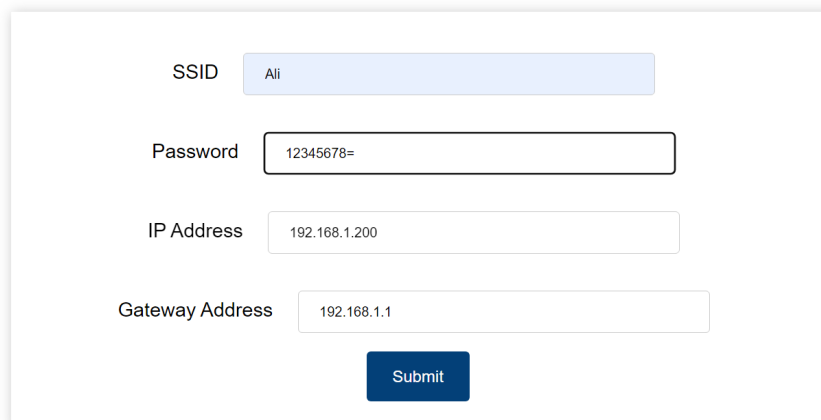
دهد.

- **style.css**: حاوی جزئیات نحوه چینش و ظاهر اجزای صفحات **html** است.

کدهای مربوط به هر سه فایل را می توانید بر روی گیتهاب در بخش کدهای مربوط به فاز 2 مشاهده کنید.

در ادامه، تصاویر دو صفحه **html** گفته شده را مشاهده می کنید:

ESP Wi-Fi Manager

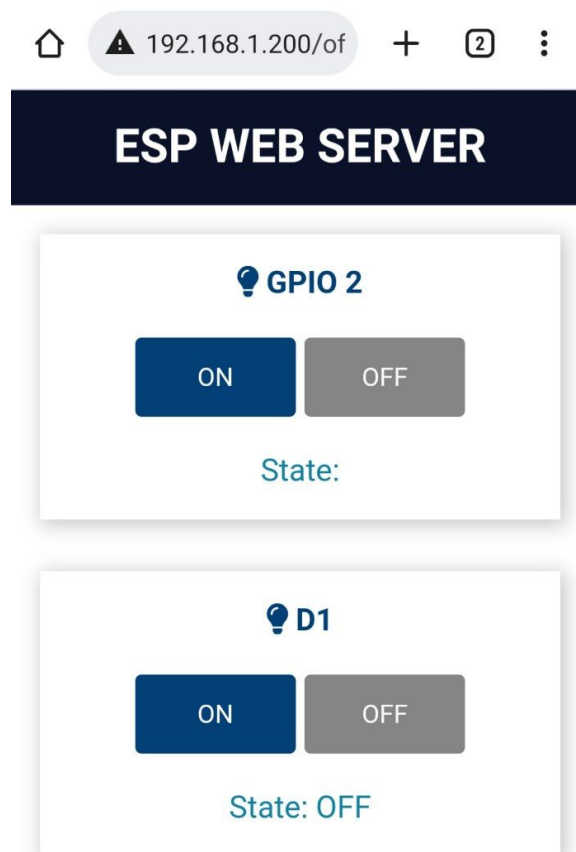


The image shows a web form titled "ESP Wi-Fi Manager". It contains four input fields with labels to their left: "SSID" with the value "Ali", "Password" with the value "12345678=", "IP Address" with the value "192.168.1.200", and "Gateway Address" with the value "192.168.1.1". Below these fields is a blue "Submit" button.

تصویر 2: وارد کردن مشخصات اتصال به مودم در صفحه **wifimanager.html**؛ شامل نام **WiFi**، رمز عبور آن، **IP Address** ای که می خواهیم بورد در شبکه داخلی داشته باشد، و آدرس درگاه.



تصویر 3: اعلام *Restart* شدن دستگاه، پس از ورود مشخصات در *wifimanager.html*



تصویر 4: پس از اتصال *NodeMCU* به مودم، کاربر به آدرس نوشته شده در بالای تصویر رفته، با ورود اتصال برقرار کرده، و صفحه *index.html* را مشاهده می کند.

همان طور که در تصویر 4 مشاهده می کنید، دو دکمه برای کنترل وضعیت کلیدها تعبیه شده است. کاربر با تغییر وضعیت دکمه ها می تواند لامپ ها را روشن و خاموش کند. توجه کنید که این صفحه که کنترل لامپ ها از آن انجام می شود، هم با استفاده از گوشی همراه و هم با استفاده از کامپیوتر شخصی قابل دسترسی است. همچنین، نحوه اتصالات سیم ها و قراردهی برد بر روی بردبرد و ال ای دی (ها)، مشابه فاز یک است و اینجا آن را دوباره تکرار نمی کنیم.

کد مربوط به سخت افزار هم در گیتهاب در بخش کدهای فاز 2 با نام "ESP_Web_Server.ino" قرار داده شده است.

توجه کنید که در نظر داریم در فاز بعدی، اتصال برد به اینترنت و کنترل آن از یک سرور را پیاده سازی کنیم. به همین منظور، در این فاز بخشی از کارهای مربوط به راه اندازی یک سرور بیرونی را نیز انجام داده ایم.

بخش دوم: فرم برنامه زمان بندی لامپ ها

در این بخش فرمی به زبان html ایجاد کردیم که در آن کاربر زمان مدنظر خود را وارد می کند، لامپ مورد نظر خود (اینکه می خواهد وضعیت لامپ 1 را تغییر دهد یا لامپ 2) را مشخص می کند و در نهایت وضعیت مدنظر برای آن لامپ (روشن یا خاموش) را مشخص می کند و دکمه ثبت را می زند تا زمان بندی مدنظر ذخیره شود و سرور NodeMCU پردازش های لازم برای انجام شدن زمان بندی را انجام دهد. در این فرم با عنوان ESP Scheduling Program که تحت فایل Scheduling Program.html ایجاد شده است، در فیلد Year سال مدنظر، در فیلد Month ماه مدنظر، در فیلد Day روز مدنظر، در فیلد Hour ساعت مدنظر و در فیلد Minute دقیقه مدنظر توسط کاربر وارد می شود.

پس از تعیین شدن زمان، کاربر در فیلد Lamp 1/Lamp 2 لامپی که می‌خواهد وضعیتش را تعیین کند مشخص می‌کند و در فیلد ON/OFF وضعیت آن لامپ (اینکه در زمان مشخص شده روشن باشد یا خاموش) را مشخص می‌کند و در نهایت دکمه Submit را فشار می‌دهد تا زمان‌بندی تعیین شده ذخیره شود و سرور NodeMCU پردازش‌های لازم را انجام دهد. فیلدهای این فرم در تصویر 5 قابل مشاهده است:

Scheduling

Year

Month

Day

Hour

Minute

Lamp 1/Lamp 2

ON/OFF

Submit

تصویر 5: فیلدهای فرم مربوط به برنامه زمان‌بندی.

بخش سوم: خواندن زمان درست از اینترنت

برای پیاده سازی زمانبندی بر روی بورد، نیاز است که بورد بتواند زمان واقعی دقیق را در خود داشته باشد که برای این کار نیاز به اتصال به اینترنت است. به همین منظور، برنامه ای نوشتیم که به هنگام اتصال بورد به اینترنت، زمان را بخواند و ذخیره کند. سپس، مقدار زمانی که باید بگذرد تا تغییر بعدی در روشن/خاموش شدن لامپ ها انجام شود را به میلی ثانیه محاسبه و ذخیره کند.

```
ESP_Web_Server
    restart = true;
    request->send(200, "text/plain", "Done. ESP will restart, connect to your router and go to IP address: " + ip);
  });
  server.begin();
}

void loop() {
  if (restart){
    delay(5000);

    timeClient.update();
    Serial.print(daysOfTheWeek[timeClient.getDay()]);
    Serial.print(", ");
    Serial.print(timeClient.getHours());
    Serial.print(":");
    Serial.print(timeClient.getMinutes());
    Serial.print(":");
    Serial.println(timeClient.getSeconds());

    ESP.restart();|
  }
}
```

تصویر 6: برنامه نوشته شده برای خواندن زمان درست از اینترنت