

Mikrocomputertechnik-Praktikum

Webbasierte Steuerung einer Klimaanlage

WiSe 24/25

Sebastian Pasinski, Benedikt Schnörr

Inhalt

Aufgaben.....	3
Übertragung der finalen Weboberfläche.....	3
Kommunikation zwischen Browser und ESP.....	3
Client.....	3
Server.....	4
Ausblick.....	5
Quellen.....	5

Aufgaben

Im letzten Bericht konnte erfolgreich von einem Endgerät auf eine einfache Weboberfläche des Servers zugegriffen werden. Diese Weboberfläche wurde nun durch die geforderte html-Datei ersetzt. Außerdem wurde die Kommunikation zwischen dem Server auf dem ESP-Mikrocontroller und dem Client hergestellt. So werden Daten von der html-Webseite gesendet, vom Mikrocontroller empfangen und können im nächsten Schritt zum Senden der Befehle an die Klimaanlage genutzt werden.

Übertragung der finalen Weboberfläche

Das Übertragen der im letzten Bericht ausgearbeiteten Weboberfläche wird hier von dem Tool SPIFFS übernommen. [1] So wird auf dem ESP32 eine Dateistruktur implementiert, die bei einem http-Request des Clients angesprochen werden kann. In diesem Fall besteht die Dateistruktur nur aus einer simplen Datei "index.html", die über die Arduino IDE auf den ESP hochgeladen werden kann.

```
// Define a route to serve the HTML page
server.on("/", HTTP_GET, [](AsyncWebServerRequest* request) {
    Serial.println("ESP32 Web Server: New request received");
    // Send html page to client
    request->send(SPIFFS, "/index.html", "text/html");
});
```

Kommunikation zwischen Browser und ESP

Eine Schnittstelle für die Datenübertragung zwischen den Akteuren bietet das XMLHttpRequest-Objekt. Dieses ermöglicht es, Informationen aus der URL der Seite, zu entnehmen. [2]

Client

Der Client empfängt das vom Server gesendete html-Dokument mit der Oberfläche zur Auswahl der An- und Ausschaltzeiten für die Klimaanlage. Dort werden die Uhrzeiten und Wochentage gewählt und dann mit dem Betätigen des Knopfes an den Server versendet. Für das onclick-Event eines Knopfes wird eine Funktion aufgerufen, welche das XMLHttpRequest-Objekt erzeugt und eine Variable "data" erzeugt.

Die "data"-Variable enthält folgende Informationen:

- Namen des Befehls
- Uhrzeit
- 7 Bit, die angeben welcher Wochentag gewählt wurde ("1" - Checkbox ausgefüllt, "0" - Checkbox leer)

Im folgenden Code-Abschnitt ist die onclick-Funktion für den Button zum Einschalten der Klimaanlage dargestellt. Dabei wird die URL um die Zeichenkette "setOn" mit der Uhrzeit und den 7 Bit für die Wochentage erweitert.

```

function onclickKlimaAn(){
    var xhttp = new XMLHttpRequest();
    var data = "/setOn?";
    var InputTimeOn = document.getElementById("InputZeitKlimaAn").value;
    // Zeit eintragen
    data += "time=" + InputTimeOn;
    // Wochentage als Abfolge von 7 Bits eintragen
    data += "&week=" + changingBoolToBit("Monday") +
changingBoolToBit("Tuesday") + changingBoolToBit("Wednesday") +
changingBoolToBit("Thursday") + changingBoolToBit("Friday") +
changingBoolToBit("Saturday") + changingBoolToBit("Sunday");
    // Sende Informationen über GET-Protokoll an Server
    xhttp.open("GET", data);
    xhttp.send();
}

```

Server

Das Programm des ESP32 wird erweitert, um auf weitere URLs zu reagieren. Dazu gehören die im letzten Abschnitt erwähnten Erweiterungen der URL, wie z.B. "setOn". Beim Aufrufen dieser URLs durch den Client wird wieder ein http-Request gesendet, welches der ESP empfängt.

Außerdem wird ein Struct erstellt, welches die übermittelten Informationen wie Uhrzeit und Wochentage zur späteren Verwendung speichert. Das Struct ist hier mit dem Empfangen des setOn-Befehls dargestellt.

```

// Create object for saving times and weekdays
struct CommandParam {
    String time;
    String weekdays;
};
CommandParam paramOn;
// Listen to url extension "setOn"
server.on("/setOn", HTTP_GET, [] (AsyncWebServerRequest *request){
    // Save times and weekdays in CommandParam object
    paramOn.time = request->getParam("time")->value();
    paramOn.weekdays = request->getParam("week")->value();
    Serial.print("on request set: " + paramOn.time + ", " +
paramOn.weekdays);
    // Send default html-file for every url extension
    request->send(SPIFFS, "/index.html", "text/html");
});

```

Ausblick

Nachdem der Server und Oberfläche für den Client aufgesetzt wurden und die Kommunikation zwischen den beiden Parteien funktioniert, muss als nächstes die Übertragung der IR-Signale an die Klimaanlage implementiert werden. Dazu gehört eine einfache Schaltung mit IR-LED und Vorwiderstand und der passende Code für das NEC-Protokoll zur Steuerung der Klimaanlage.

Quellen

[1]:

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/storage/spiffs.html>,

letzter Zugriff am 04.11.24.

[2]: <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest#specifications>,

letzter Zugriff am 03.11.24.

[3]:

<https://github.com/me-no-dev/ESPAsyncWebServer?tab=readme-ov-file#handlers-and-how-do-they-work>, letzter Zugriff am 03.11.24.