

```

1  // -// 12. SPIFF Webszerver, SPIFFS v2.pdf
2  // -// ESP8266 - Fájlok listázása
   https://main.elearning.uni-obuda.hu/pluginfile.php/464853/mod\_resource/content/1/SPIFFS\_v2.pdf
3
4
5  #include <ESP8266WiFi.h>
6  #include <FS.h>
7  void setup() {
8      Serial.begin(115200);
9      // Fájrendszer csatolása
10     SPIFFS.begin();
11     //gyökérkönyvát megnyitása
12     //SPIFFS esetén csak a gyökérkönyvtár van!
13     Dir dir = SPIFFS.openDir ("/");
14     //dir.next() = true amíg van fájl
15     while (dir.next()) {
16         Serial.println(dir.fileName());
17         Serial.println(dir.fileSize());
18     }
19 }
20 void loop() {
21 }
22
23 // -// ESP8266 - Fájl megnyitása és olvasása
24 #include "FS.h"
25
26 void setup() {
27     Serial.begin(115200);
28
29     if(!SPIFFS.begin()){
30         Serial.println("An Error has occurred while mounting SPIFFS");
31         return;
32     }
33
34     File file = SPIFFS.open("/test_example.txt", "r");
35     if(!file){
36         Serial.println("Failed to open file for reading");
37         return
38     }
39
40     Serial.println();
41     Serial.println("File Content:");
42     while(file.available()){
43         Serial.write(file.read());
44     }
45     file.close();
46 }
47
48 void loop() {
49 }
50
51 // -// Aszinkron webszerver példa LED vezérlés külön html,css,js fájlok SPIFFS-sel
52 // script.js:
53 async function setLed(state) {
54     try {
55         const response = await fetch("/led", {
56             method: "POST",
57             headers: {
58                 "Content-Type": "application/x-www-form-urlencoded"
59             },
60             // A query paraméter a POST body-ba kerül
61             body: `state=${state}`
62         });
63
64         if (!response.ok) {
65             throw new Error(`Hiba: ${response.status} - ${response.statusText}`);
66         }
67
68         const data = await response.text();
69         // Kiírjuk a válasz szöveget
70         console.log("Szerver válasza:", data);
71     } catch (error) {

```

```

72         // Hibakezelés
73         console.error("Hiba történt:", error);
74     }
75 }
76
77 // styles.css
78 body {
79     font-family: Arial, sans-serif;
80     background-color: #f0f0f0;
81     color: #f0f0f0;
82     margin: 0;
83     padding: 0;
84     display: flex;
85     flex-direction: column;
86     justify-content: center;
87     align-items: center;
88     text-align: center;
89 }
90
91 //index.html
92 <!DOCTYPE html>
93 <html lang="hu">
94
95 <head>
96     <meta charset="UTF-8">
97     <meta name="viewport" content="width=device-width, initial-scale=1.0">
98     <title>LED vezérlés</title>
99     <!-- CSS FÁJL HIVATKOZÁS-->
100     <link rel="stylesheet" href="styles.css">
101     <!-- defer -> DOM betöltése után indul el-->
102     <script src="script.js" defer></script>
103 </head>
104
105 <body>
106     <header>
107         
108     </header>
109     <h1>LED vezérlés</h1>
110     <button onclick="setLed('on')">LED BE</button>
111     <button onclick="setLed('off')">LED KI</button>
112 </body>
113
114 </html>
115
116 // Aszinkron webserverver kód
117 #include <ESP8266WiFi.h>
118 #include <ESPAsyncWebServer.h>
119 #include <FS.h>
120
121 const char* ssid = "BIR_WIFI";
122 const char* password = "OeBir2019";
123
124 AsyncWebServer server(80);
125
126 // LED BE kapcsolása
127 void handleLed(AsyncWebServerRequest* request) {
128     // Ha a paraméter 'stata' létezik a bodyban
129     if (request->hasParam("state", true)) {
130         String state = request->arg("state");
131         // LED vezérlés
132         if (state == "on") {
133             digitalWrite(LED_BUILTIN, LOW);
134         } else if (state == "off") {
135             digitalWrite(LED_BUILTIN, HIGH);
136         }
137         // Válasz küldése
138         request->send(200, "text/plain", "LED állapot: " + state);
139     } else {
140         // Ha nincs 'state' paraméter, hibát küldünk
141         request->send(400, "text/plain", "Hibás kérés");
142     }
143 }
144

```

```
145 void notFound(AsyncWebServerRequest* request) {
146     String message = "A keresett oldal nem található!";
147     request->send(404, "text/plain", message);
148 }
149
150 void setup() {
151     Serial.begin(115200);
152     while (!Serial){
153         delay(100);
154     }
155
156     pinMode(LED_BUILTIN, OUTPUT);
157     Serial.println();
158
159     WiFi.begin(ssid, password);
160     while (WiFi.status() != WL_CONNECTED) {
161         delay(500);
162         Serial.print(".");
163     }
164     Serial.println("\nCsatlakozva! IP: " + WiFi.localIP().toString());
165
166     //SPIFFS inicializálása
167     if (!SPIFFS.begin()) {
168         Serial.println("SPIFFS inicializálás sikertelen!");
169         return;
170     }
171
172     //p11. server.serveStatic("/", SPIFFS, "/").setDefaultFile("index.html");
173     //param1: gyökér URL esetén
174     //param2: a szerver a SPIFFS fájlrendszerből tölti be az adatokat.
175     //param3: SPIFFS fájlrendszerben lévő fájlok gyökérkönyvtárát jelöli
176     server.serveStatic("/", SPIFFS, "/").setDefaultFile("index.html");
177     server.on("/led", HTTP_POST, handleLed);
178     server.onNotFound(notFound);
179     server.begin();
180 }
181
182 void loop() {}
183
184
185
186
187
188
189
190
```