



149 lines - 52 Removals

```
1 import os
2 import time
3 import ipaddress
4 import wifi
5 import socketpool
6 import board
7 import microcontroller
8 import terminalio
9 from digitalio import DigitalInOut, Direction
10 from adafruit_httpserver.server import HTTPServer
11 from adafruit_httpserver.request import HTTPRequest
12 from adafruit_httpserver.response import HTTPResponse
13 from adafruit_httpserver.methods import HTTPMethod
14 from adafruit_httpserver.mime_type import MIMEType
15
16 # onboard LED setup
17 led = DigitalInOut(board.LED)
18 led.direction = Direction.OUTPUT
19 led.value = False
20
21 # function to convert celcius to fahrenheit
22 def c_to_f(temp):
23     temp_f = (temp * 9/5) + 32
24     return temp_f
25
26 # connect to network
27 print()
28 print("Connecting to WiFi")
29
30 # connect to your SSID
31 wifi.radio.connect(os.getenv('CIRCUITPY_WIFI_SSID'),
32                   os.getenv('CIRCUITPY_WIFI_PASSWORD'))
33 print("Connected to WiFi")
34 pool = socketpool.SocketPool(wifi.radio)
35 server = HTTPServer(pool, "/static")
36
37 # variables for HTML
38
39 unit = "F"
40
41 # font for HTML
42 font_family = "monospace"
43
```

147 lines + 51 Additions

```
1 import os
2 import time
3 import ipaddress
4 import wifi
5 import socketpool
6 import board
7 import microcontroller
8 import terminalio
9 from digitalio import DigitalInOut, Direction
10 from adafruit_httpserver.server import HTTPServer
11 from adafruit_httpserver.request import HTTPRequest
12 from adafruit_httpserver.response import HTTPResponse
13 from adafruit_httpserver.methods import HTTPMethod
14 from adafruit_httpserver.mime_type import MIMEType
15
16 # onboard LED setup
17 led = DigitalInOut(board.LED)
18 led.direction = Direction.OUTPUT
19 led.value = False
20
21 # function to convert celcius to fahrenheit
22 def c_to_f(temp):
23     temp_f = (temp * 9/5) + 32
24     return temp_f
25
26 # connect to network
27 print()
28 print("Connecting to WiFi")
29
30 # set static IP address
31 ipv4 = ipaddress.IPv4Address("192.168.1.42")
32 netmask = ipaddress.IPv4Address("255.255.255.0")
33 gateway = ipaddress.IPv4Address("192.168.1.1")
34 wifi.radio.set_ipv4_address(ipv4=ipv4, netmask=netmask, gateway=gateway)
35 # connect to your SSID
36 wifi.radio.connect(os.getenv('CIRCUITPY_WIFI_SSID'),
37                   os.getenv('CIRCUITPY_WIFI_PASSWORD'))
38 print("Connected to WiFi")
39 pool = socketpool.SocketPool(wifi.radio)
40 server = HTTPServer(pool, "/static")
41
42 # variables for HTML
43 temp_test = str(c_to_f(microcontroller.cpu.temperature))
44
45 unit = "F"
46 i = 0
47 istr = str(i)
48 # font for HTML
49 font_family = "monospace"
50
```

```

42 # the HTML script
43 # setup as an f string
44 # this way, can insert string variables from code.p
y directly
45 # of note, use {{ and }} if something from html *ac
tually* needs to be in brackets
46 # i.e. CSS style formatting
47 def webpage():
48     temp = str(c_to_f(microcontroller.cpu.temperatur
e))
49
50     html = f"""
51     <!DOCTYPE html>
52     <html>
53     <head>
54     <meta http-equiv="Content-type" content="text/ht
ml; charset=utf-8">
55     <meta name="viewport" content="width=device-widt
h, initial-scale=1">
56     <script src="https://ajax.googleapis.com/ajax/li
bs/jquery/3.6.4/jquery.min.js"></script>
57     <style>
58     html{{font-family: {font_family}; background-col
or: lightgrey;
59     display:inline-block; margin: 0px auto; text-ali
gn: center;}}
60     h1{{color: deeppink; width: 200; word-wrap: br
eak-word; padding: 2vh; font-size: 20px;}}
61     p{{font-size: 15px; width: 200; word-wrap: bre
ak-word;}}
62     .button{{font-family: {font_family}; display: i
nline-block;
63     background-color: black; border: none;
64     border-radius: 4px; color: white; padding: 5px
5px;
65     text-decoration: none; font-size: 30px; margi
n: 2px; cursor: pointer;}}
66     p.dotted {{margin: auto;
67     width: 75%; font-size: 15px; text-align: cente
r;}}
68     </style>
69     </head>
70     <body>
71     <title>Pico W HTTP Server</title>
72     <h1>Pico W HTTP Server</h1>
73     <br>
74     <p class="dotted">This is a Pico W running an HT
TP server with CircuitPython.</p>
75     <br>
76     <p class="dotted">The current ambient temperatur
e near the Pico W is
77     <span style="color: deeppink;"><span id="temp">
{temp}</span>°<span>{unit}</span></p><br>
78     <h1>Control the LED on the Pico W with these but
tons:</h1><br>
79     <p><button class="button" name="LED" value="ON"
type="submit">LED ON</button></a></p>
80     <p><button class="button" name="LED" value="OFF"
type="submit">LED OFF</button></a></p>

```

```

50 # the HTML script
51 # setup as an f string
52 # this way, can insert string variables from code.p
y directly
53 # of note, use {{ and }} if something from html *ac
tually* needs to be in brackets
54 # i.e. CSS style formatting
55 def webpage():
56     html = f"""
57     <!DOCTYPE html>
58     <html>
59     <head>
60     <meta http-equiv="Content-type" content="text/ht
ml; charset=utf-8">
61     <meta name="viewport" content="width=device-widt
h, initial-scale=1">
62     <style>
63     html{{font-family: {font_family}; background-col
or: lightgrey;
64     display:inline-block; margin: 0px auto; text-ali
gn: center;}}
65     h1{{color: deeppink; width: 200; word-wrap: br
eak-word; padding: 2vh; font-size: 20px;}}
66     p{{font-size: 15px; width: 200; word-wrap: bre
ak-word;}}
67     .button{{font-family: {font_family}; display: i
nline-block;
68     background-color: black; border: none;
69     border-radius: 4px; color: white; padding: 5px
5px;
70     text-decoration: none; font-size: 30px; margi
n: 2px; cursor: pointer;}}
71     p.dotted {{margin: auto;
72     width: 75%; font-size: 15px; text-align: cente
r;}}
73     </style>
74     </head>
75     <body>
76     <title>Pico W HTTP Server</title>
77     <h1>Pico W HTTP Server</h1>
78     <br>
79     <p class="dotted">This is a Pico W running an HT
TP server with CircuitPython.</p>
80     <br>
81     <p class="dotted">The current ambient temperatur
e near the Pico W is
82     <span style="color: deeppink;">{temp_test}°<uni
t}</span></p><br>
83     <span style="color: black;">i={istr}</span></p><br>
84     <h1>Control the LED on the Pico W with these but
tons:</h1><br>
85     <form accept-charset="utf-8" method="POST">
86     <button class="button" name="LED ON" value="ON"
type="submit">LED ON</button></a></p></form>

```

```

81     </body>
82     <script>
83         // LED buttons handler
84         $(".button").click((el) => {{ $.post("/led",
85         {{[el.target.name]: el.target.value}} }});
86         // 1 sec temperature refresher
87         setInterval(() => {{
88             $.get("/temp", (data) => {{
89                 $("#temp").text(data);
90             }});
91         }}, 1000);
92     </script>
93     </html>
94     """
95     return html
96 # route default static IP
97 @server.route("/")
98 def base(request: HTTPRequest):
99     # serve the HTML f string
100     # with content type text/html
101     with HTTPResponse(request, content_type=MIMETyp
102     e.TYPE_HTML) as response:
103         response.send(f"{webpage()}")
104 # change led state
105 @server.route("/led", method=HTTPMethod.POST)
106 def buttonpress(request: HTTPRequest):
107     print("button request")
108     # get query params (doesn't handle all cases, u
109     se with caution
110     query = {x[0] : x[1] for x in [x.split("=") for
111     x in request.body.decode("utf8").split("&")]}}
112     # if the led on button was pressed
113     if "LED" in query:
114         # turn on or off the onboard LED
115         led.value = (query["LED"] == "ON")
116         # Acknowledge
117         with HTTPResponse(request, content_type=MIMETyp
118         e.TYPE_HTML) as response:
119             response.send()
120 # get temperature
121 @server.route("/temp")
122 def temp(request: HTTPRequest):

```

```

87     <p><form accept-charset="utf-8" method="POST">
88     <button class="button" name="LED OFF" value="OF
89     F" type="submit">LED OFF</button></a></p></form>
90     </body></html>
91     """
92     return html
93 # route default static IP
94 @server.route("/")
95 def base(request: HTTPRequest):
96     # serve the HTML f string
97     # with content type text/html
98     with HTTPResponse(request, content_type=MIMETyp
99     e.TYPE_HTML) as response:
100         response.send(f"{webpage()}")
101 # if a button is pressed on the site
102 @server.route("/", method=HTTPMethod.POST)
103 def buttonpress(request: HTTPRequest):
104     # get the raw text
105     raw_text = request.raw_request.decode("utf8")
106     print("raw text = ")
107     print(raw_text)
108     # if the led on button was pressed
109     if "ON" in raw_text:
110         # turn on the onboard LED
111         led.value = True
112     # if the led off button was pressed
113     if "OFF" in raw_text:
114         # turn the onboard LED off
115         led.value = False
116     # reload site
117     with HTTPResponse(request, content_type=MIMETyp
118     e.TYPE_HTML) as response:
119         response.send(f"{webpage()}")
120 print("starting server..")

```