

```
1 # tinyWeb_AP_5.py - Based on this - https://www.recantha.co.uk/blog/?p=21398
2 # should auto run as AP 'PicoW_2' and display in browser the core temp. after 14 secs from power-up.
3 # Some parts based on an idea - https://medium.com/geekculture/striking-binary-clock-with-raspberry-pi-pico-w-and-addressable-rgb-leds-ee900a6d3cde
4 # Now index page, to select option - on home page, also lets the user toggle an output GP15, in this case.
5 # Phil J, June 23 - Pico W - AP mode
6
7 import socket, machine
8 import network
9 import tinyweb
10 from machine import Pin, ADC
11 from utime import sleep
12 import gc
13
14 # machine.freq(250000000) #set system clock to 250Mhz
15
16 # Define SSID and password for the access point
17 ssid = "PicoW_2"
18 password = "ph1lj12345" #requires 9 characters
19
20 vcore = ADC(4)
21 cf = 3.3/(65553)
22 value = 0
23
24 # Define an access point, name it and then make it active
25 ap = network.WLAN(network.AP_IF)
26 ap.config(essid=ssid, password=password)
27 ap.active(True)
28
29 # Wait until it is active
30 while ap.active == False:
31     pass
32
33 print("Access point active")
34 # Print out IP information
35 print(ap.ifconfig())
36
37 # Define on-board LED & output pin 'outbit'
38 led = Pin("LED", Pin.OUT)
39 outbit = Pin(15, Pin.OUT)
40
41 # default state
42 outbit.value(False)
43
44 # Start up a tiny web server
45 app = tinyweb.webserver()
46 led.value(True)
47
48 #-----
49 # Serve a simple Core Temp.value as response when / is called
50 # and turn the LED on/off using toggle()
```

```

51 # @app.route('/')
52 # async def index(request, response):
53 #     # Start HTTP response with content-type text/html
54 #     await response.start_html()
55 #     # Send actual HTML page
56 #     await response.send('<html><body><h1>Hello, world!</h1></body></html>\n' + str(value))
57 #
58 #     led.toggle()
59 #-----
60
61 @app.route('/')
62 async def index(request, response):
63     # Start HTTP response with content-type text/html
64     await response.start_html()
65     # Send actual HTML page
66     await response.send('''
67         <!DOCTYPE html>
68         <html>
69             <head>
70                 <title>Pico W tinyweb example v5</title>
71             </head>
72             <body style="display:flex; flex-direction:column; justify-content:center; align-items:center; padding: 50px; font-family: verdana;">
73                 <h1 style="font-size: 4rem; color:blue"> 'tinyweb' on the Pico W example .... </h1>
74                 <h2 style="font-size: 3rem;">Choose an option:-</h2>
75                 <a href="/memory" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%; height:
100px; background-color: #cabbb8; color: black; border-radius: 15px;">Pico Memory usage</button></a>
76                 <a href="/btn" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%; height:
100px; background-color: #cabbb8; color: black; border-radius: 15px;">Toggle 'outbit' (GP15)</button></a>
77                 <a href="/core" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%; height:
100px; background-color: #cabbb8; color: black; border-radius: 15px;">Pico Core Temperature</button></a>
78                 <a href="/about" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%; height:
100px; background-color: #cabbb8; color: black; border-radius: 15px;">About</button></a>
79
80                 <!-- These are commented out <a href="/10-second-timer" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem;
font-family: verdana; width:100%; height: 150px; background-color: #ffe6e6; color: black; border-radius: 15px;">10 Second Timer</button></a>
81                 <a href="/30-second-timer" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana;
width:100%; height: 150px; background-color: #ffe6e6; color: black; border-radius: 15px;">30 Second Timer</button></a>
82                 <a href="/1-minute-timer" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%;
height: 150px; background-color: #ffe6e6; color: black; border-radius: 15px;">1 Minute Timer</button></a>
83                 <a href="/2-minute-timer" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%;
height: 150px; background-color: #ffe6e6; color: black; border-radius: 15px;">2 Minute Timer</button></a>
84                 <a href="/3-minute-timer" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%;
height: 150px; background-color: #ffe6e6; color: black; border-radius: 15px;">3 Minute Timer</button></a>
85                 <a href="/4-minute-timer" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%;
height: 150px; background-color: #ffe6e6; color: black; border-radius: 15px;">4 Minute Timer</button></a>
86                 <a href="/off" style="margin-bottom: 50px; width:100%;"><button style="font-size:4rem; font-family: verdana; width:100%; height:
150px; background-color: #ff6666; color: black; border-radius: 15px;">Stop</button></a>
87                 -->
88             </body>
89         </html>
90     ''')

```

```

91     print("home")
92
93
94 # Serve a simple Free Memory reading as response to when '/memory' is called
95 # and turn the LED on/off using toggle()
96 @app.route('/memory')
97 async def index(request, response):
98     await response.start_html()
99     await response.send('<h1>Free Memory={ } bytes</h1>'.format(gc.mem_free()))
100     led.toggle()
101
102 # Serve a simple Core Temp.value as response to when '/' is called
103 # and turn the LED on/off using toggle()
104 @app.route('/')
105 async def index(request, response):
106     reading = vcore.read_u16() * cf
107     res = 27 - (reading - 0.706)/0.001721
108     print(res)
109     value = round(res, 1)
110     #response = ('Core temp = ', value)
111     await response.start_html()
112     await response.send('<html><head><title>tinyWeb 2 . . </title>' )
113     await response.send('<meta http-equiv="refresh" content="10">')
114     await response.send('<html><body><h1>The Core temp. = ' + str(value) + '&deg;C</h1></body></html>' )
115     await response.send('<html><body><h2>Will refresh the value, every 10 seconds</h2></body></html>')
116     led.toggle()
117
118 # Serve to display button as response to when '/btn' is called
119 #
120 @app.route('/btn')
121 async def index(request, response):
122     await response.start_html()
123     await response.send('''
124         <!DOCTYPE html>
125         <html>
126             <body>
127
128                 <h1>The button Element</h1>
129
130                 <button type="button" onclick="alert('Toggle the outbit !')">Click Me!</button>
131
132             </body>
133         </html>
134     ''')
135     outbit.toggle()
136
137 # Serve to display button as response to when '/about' is called
138 #
139 @app.route('/about')
140 async def index(request, response):
141     await response.start_html()

```

```
142     await response.send('''
143         <!DOCTYPE html>
144         <html>
145             <body>
146                 <h1>Created by Phil J</h1>
147             </body>
148         </html>
149     ''')
150     outbit.toggle()
151
152 # Run the web server as the sole process
153 app.run(host="0.0.0.0", port=80)
154
```