

```

1
2 --Barometer Glass & Brass V1.2 with three hourly trend pressure
3 -- *****
4 ssid = "yourSSID" -- your router SSID
5 pass = "yourPassword" -- your router password
6 alti = 185 -- set correction for your altitude location in meters
7 -- *****
8 wifi.setmode(wifi.STATION)
9 wifi.sta.config(ssid, pass,1)
10 _,RID=node.bootreason()
11 myheap=0
12 srp=5      -- period reading of sensors [s]
13 sda=4      -- GPIO2 connect to SDA BMP180
14 scl=3      -- GPIO0 connect to SCL BMP180
15 press=0    -- actual pressure from BMP180
16 pressold=0 -- pressure (n-3)hour
17 temp=0     -- temperature BMP180
18 PressAngle=0 -- angle of pointer calculated from pressure  PressAngle = 30+(press-960)*3
19 TrendAngle=189.9 -- angle of "press trend" pointer
20 ctdown=3   -- hour counter for "press trend"
21 -- web page - barometer face with included parameters
22 function LoadBuff()
23 local buff2 = '<!DOCTYPE HTML><html><head><meta charset="UTF-8"><meta
24 http-equiv="refresh" content="60">\
25 <title>Barometer</title></head><body style="background:skyblue">\
26 <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
27 width="1000" height="1000">\
28 <circle cx="500" cy="500" r="265" fill="white" stroke="#b46b41" stroke-width="16"/>\
29 <circle cx="500" cy="500" r="265" fill-opacity="0" stroke="#c68762" stroke-width="4"/>\
30 <g stroke="black" stroke-width="2" fill="none">\
31 <path id="SU" d="M600 445a15 15 0 1 1 0 1h-15m45 0h15m-30 15v15m0-45v-15m11 41110
32 10m-32-32l-10-10m10 32l-11 11m33-33l10-10"/>\
33 <path id="CL" fill="white" d="M350 455h70a-15-15 0 1 0-15-30a-15-10 0 1 0-40 0a5 5 0 0
34 0-15 30"/>\
35 <path id="RA" d="M350 460l-5 10m20-10l-5 10m20-10l-5 10m20-10l-5 10"/>\
36 <rect x="450" y="590" rx="5" ry="5" width="100" height="40"/>\
37 <rect x="450" y="635" rx="5" ry="5" width="100" height="40"/></g>\
38 <defs>\
39 <g id="PO">\
40 <path d="M495 440A12 12 0 1 1 505 440L505 440 503 710 500 720 497 710z" fill="black"/>\
41 <circle cx="500" cy="500" r="10" stroke="black" stroke-width="3" fill="#Cd7F32"/>\
42 <circle cx="500" cy="500" r="5" fill="grey"/>\
43 <line stroke-width="2" id="S" x1="500" y1="725" x2="500" y2="750"/>\
44 <line stroke-width="2" id="M" x1="500" y1="720" x2="500" y2="750"/>\
45 <line stroke-width="2" id="L" x1="500" y1="705" x2="500" y2="750"/></g>\
46 <g id="T">\
47 <use xlink:href="#L" transform="rotate(30,500,500)"/>\
48 <g id="D">\
49 <use xlink:href="#S" transform="rotate(33,500,500)"/>\
50 <use xlink:href="#S" transform="rotate(36,500,500)"/>\
51 <use xlink:href="#S" transform="rotate(39,500,500)"/></g>\
52 <use xlink:href="#S" transform="rotate(42,500,500)"/>\
53 <use xlink:href="#M" transform="rotate(45,500,500)"/>\
54 <use xlink:href="#S" transform="rotate(48,500,500)"/>\
55 <use xlink:href="#D" transform="rotate(18,500,500)"/></g></defs>\
56 <g stroke="black">\
57 <use xlink:href="#D" transform="rotate(-12,500,500)"/>\
58 <g id="TT">\
59 <use xlink:href="#T"/>\
60 <use xlink:href="#T" transform="rotate(30,500,500)"/>\
61 <use xlink:href="#T" transform="rotate(60,500,500)"/>\
62 <use xlink:href="#T" transform="rotate(90,500,500)"/>\
63 <use xlink:href="#T" transform="rotate(120,500,500)"/></g>\
64 <use xlink:href="#TT" transform="rotate(150,500,500)"/>\
65 <use xlink:href="#L" transform="rotate(330,500,500)"/>\
66 <use xlink:href="#D" transform="rotate(300,500,500)"/></g>\
67 <g font-family="arial" font-size="28" font-weight="normal" font-style="italic"
68 fill="black" >\
69 <text x="330" y="600">970</text>\
70 <text x="500" y="500">980</text>\
71 <text x="330" y="400">990</text>\
72 <text x="390" y="350">1000</text>\
73 <text x="465" y="320">1010</text>\
74 <text x="540" y="350">1020</text>\
75 <text x="605" y="400">1030</text>\
76 <text x="635" y="500">1040</text>\
77 <text x="610" y="600">1050</text>\
78 <text x="470" y="740">hPa</text>\
79 <text x="470" y="620" fill="red">'..temp..'</text>\
80 <text x="560" y="620" font-size="24">C</text>\
81 <text x="455" y="665" fill="green">'..press..'</text>\

```

```

77 <text x="555" y="663" font-size="24" >hPa</text>\
78 <circle cx="560" cy="600" r="3" fill="white" stroke="black" stroke-width="2"/></g>\
79 <g stroke="black" stroke-width="2" fill="none">\
80 <use xlink:href="#SU" transform="translate(-127,-90)"/>\
81 <use xlink:href="#CL" transform="translate(123,-65)"/></g>\
82 <use xlink:href="#S" stroke="red" transform="rotate('..TrendAngle..',500,500)"/>\
83 <use xlink:href="#PO" transform="rotate('..PressAngle..',500,500)"/>\
84 </svg>Reset ID : '..RID..' Heap : '..myheap..'</body></html>'
85 lenght= #buff2
86 buff1 = 'HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n'..
87 'Content-Length: '.. lenght ..'\r\n'..
88 'Cache-Control: max-age=120\r\n'..
89 'Connection: Keep-Alive\r\n\r\n'..buff2
90 buff2=nil
91 collectgarbage()
92 end
93 -- read sensor, load buffer
94 function ReadBMP180()
95 bmp085.init(sda,scl)
96 press=string.format("%.1f",tostring(bmp085.pressure()/((1-alti/44330)^5.255)/100))
97 temp=string.format("%.1f",tostring(bmp085.temperature()/10))
98 PressAngle=string.format("%.1f",tostring(30+(press-960)*3))
99 myheap=node.heap()
100 LoadBuff()
101 --print("Lenght of http page: ",#buff1)
102 collectgarbage()
103 end
104 -- web server
105 function StartServer()
106 srv = net.createServer(net.TCP, 120)
107 srv:listen(80, function (conn)
108     conn:on("receive",
109         function (client, request)
110             if string.find(request, "GET / HTTP/1.1") ~= nil
111                 then
112                     client:send(string.sub(buff1,1, (#buff1 > 1460) and 1460 or #buff1),
113                         function()
114                             if #buff1>1460
115                                 then
116                                     client:send(string.sub(buff1,1461,(#buff1 > 2920) and 2920
117                                     or #buff1),
118                                         function()
119                                             if #buff1 > 2920
120                                                 then
121                                                     client:send(string.sub(buff1,2921,#buff1),
122                                                         function()
123                                                             collectgarbage()
124                                                             end)
125                                                         end)
126                                                     end)
127                                                 end)
128                                         collectgarbage()
129                                         else
130                                             client:send('HTTP/1.1 404 Not found\r\n'..
131                                                 'Content-Type: text/html\r\n'..
132                                                 'Content-Length: 25\r\n'..
133                                                 'Connection: close\r\n\r\n'..
134                                                 '<h1>Page not found !</h1>')
135                                             client:close()
136                                         end)
137                                     end)
138                                 end)
139                             end)
140                         -- main loop
141                         tmr.alarm(0,1000,1,
142                             function()
143                                 if wifi.sta.getip()==nil
144                                     then
145                                         print("wait for IP")
146                                     else
147                                         ipa,_,_=wifi.sta.getip()
148                                         print("IP is ",ipa)
149                                         tmr.stop(0)
150                                         ReadBMP180()
151                                         print("Server start")
152                                         StartServer()
153                                         -- periodic reading of sensors and load buffer
154                                         tmr.alarm(1,srp*1000,1,
155                                             function()
156                                                 ReadBMP180()

```

```
157         print("My heap :",myheap)
158     end)
159     pressold=press
160     TrendAngle=string.format("%.1f",tostring(30+(pressold-960)*3))
161     -- three hourly periodic trending
162     tmr.alarm(0,3600000,1,
163     function()
164         ctdown=ctdown-1
165         if ctdown < 1
166         then
167             TrendAngle=string.format("%.1f",tostring(30+(pressold-960)*3))
168             pressold=press
169             ctdown=3
170         end
171     end)
172 end
173 end)
174
```