

```

1  --Hygrometer V1.0 JankOP Factory
2  -- *****
3  ssid = "*****" -- your router SSID
4  pass = "*****" -- your router password
5  fahrenheit=false -- set Celsius or Fahrenheit grade
6  -- false = Celsius` true = Fahrenheit
7  -- *****
8  wifi.setmode(wifi.STATION)
9  wifi.sta.config(ssid, pass,1)
10 _,RID=node.bootreason()
11 srp=5 -- period reading of sensors [s]
12 pin=4 -- GPIO2 connect to DHT sensor
13 humi=0 -- humidity
14 temp=0 -- temperature
15 DhtState=""
16 HumiAngle=0 -- angle of pointer calculated from humidity, HumiAngle = 40+humi*2.8
17 --read DHT sensor
18 function ReadDHT()
19     status, temp, humi, temp_dec, humi_dec = dht.read(pin)
20     if status == dht.OK
21     then
22         DhtState="DHT state : OK"
23         print("Humidity : ",humi)
24         if fahrenheit==true
25         then
26             print("Temperature : ",9/5*temp+32," Fahrenheit")
27         else
28             print("Temperature : ",temp," Celsius")
29         end
30     elseif status == dht.ERROR_CHECKSUM
31     then
32         DhtState="DHT state : Checksum error"
33         print( DhtState )
34     elseif status == dht.ERROR_TIMEOUT
35     then
36         DhtState="DHT state : Timed out error"
37         print( DhtState )
38     end
39     if fahrenheit==true
40     then
41         temp=9/5*temp+32
42         unit="F"
43     else
44         unit="C"
45     end
46     humi= string.format("%.1f",tostring(humi))
47     temp= string.format("%.1f",tostring(temp))
48     HumiAngle = 40+humi*2.8
49 end
50 -- web page - hygrometer face with included parameters
51 function LoadBuff()
52     local buff2 = '<!DOCTYPE HTML><html><head><meta charset="UTF-8"><meta
53 http-equiv="refresh" content="30">\
54 <title>Hygrometer</title></head><body style="background:silver">\
55 <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
56 width="1000" height="1000">\
57 <circle cx="500" cy="500" r="265" fill="whitesmoke" stroke="#b46b41" stroke-width="16"/>\
58 <circle cx="500" cy="500" r="265" fill-opacity="0" stroke="#c68762" stroke-width="4"/>\
59 <g stroke="black" stroke-width="2" fill="none">\
60 <rect x="450" y="590" rx="5" ry="5" width="100" height="40"/>\
61 <rect x="450" y="635" rx="5" ry="5" width="100" height="40"/></g>\
62 <path fill="none" stroke="#2e8b57" stroke-width="20" d="M413,331 A190,190 35 0 1
63 657,393" />\
64 <defs>\
65 <g id="PO">\
66 <line stroke-width="5" x1="500" y1="500" x2="500" y2="700" fill="orangered"/>\
67 <rect x="492" y="455" rx="2" ry="2" width="16" height="64" fill="orangered"/>\
68 <circle cx="500" cy="500" r="2" stroke="brown" stroke-width="4" />\
69 <circle cx="500" cy="500" r="3" fill="grey"/></g>\
70 <g>\
71 <line stroke-width="4" id="M" x1="500" y1="680" x2="500" y2="700"/>\
72 <line stroke-width="6" id="L" x1="500" y1="670" x2="500" y2="700"/></g>\
73 <g id="Z">\
74 <use xlink:href="#L" transform="rotate(0,500,500)"/>\
75 <use xlink:href="#M" transform="rotate(14,500,500)"/></g></defs>\
76 <g id="T" stroke="black">\
77 <use xlink:href="#Z" transform="rotate(40,500,500)"/>\
78 <use xlink:href="#Z" transform="rotate(68,500,500)"/>\
79 <use xlink:href="#Z" transform="rotate(96,500,500)"/>\
80 <use xlink:href="#Z" transform="rotate(124,500,500)"/>\
81 <use xlink:href="#Z" transform="rotate(152,500,500)"/>\

```

```

79 <use xlink:href="#Z" transform="rotate(180,500,500)"/>\
80 <use xlink:href="#Z" transform="rotate(208,500,500)"/>\
81 <use xlink:href="#Z" transform="rotate(236,500,500)"/>\
82 <use xlink:href="#Z" transform="rotate(264,500,500)"/>\
83 <use xlink:href="#Z" transform="rotate(292,500,500)"/>\
84 <use xlink:href="#L" transform="rotate(320,500,500)"/></g>\
85 <g font-family="arial" font-size="36" font-weight="normal" font-style="normal"
stroke="black" fill="black" >\
86 <text x="350" y="685">0</text>\
87 <text x="270" y="590">10</text>\
88 <text x="255" y="490">20</text>\
89 <text x="290" y="395">30</text>\
90 <text x="378" y="315">40</text>\
91 <text x="480" y="295">50</text>\
92 <text x="577" y="316">60</text>\
93 <text x="660" y="380">70</text>\
94 <text x="705" y="490">80</text>\
95 <text x="690" y="595">90</text>\
96 <text x="610" y="690">100</text>\
97 <text x="450" y="720">% RH</text>\
98 <text x="470" y="620" fill="brown">'..temp..'</text>\
99 <text x="560" y="620" font-size="24">'..unit..'</text>\
100 <text x="470" y="665" fill="blue">'..humi..'</text>\
101 <circle cx="560" cy="595" r="3" fill="white" stroke="black" stroke-width="2"/></g>\
102 <g stroke="red" stroke-width="1" fill="none">\
103 <use xlink:href="#PO" transform="rotate('..HumiAngle..' ,500,500)"/></g></svg>\
104 <ul><li>node.chipid : '..node.chipid()..'</li>\
105 <li>node.bootreason : '..RID..'</li>\
106 <li>node.heap : '..node.heap()..'</li>\
107 <li>'..DhtState..'</li></ul></body></html>'
108 lenght= #buff2
109 buff1 = 'HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n'..
110 'Content-Length: '.. lenght ..'\r\n'..
111 'Cache-Control: max-age=120\r\n'..
112 'Connection: Keep-Alive\r\n\r\n'..buff2
113 buff2=nil
114 end
115 -- web server
116 function StartServer()
117   srv = net.createServer(net.TCP, 40)
118   srv:listen(80, function (conn)
119     conn:on("receive",
120       function (client, request)
121         if string.find(request, "GET / HTTP/1.1") ~= nil
122         then
123           client:send(string.sub(buff1,1, (#buff1 > 1460) and 1460 or #buff1),
124             function()
125               if #buff1>1460
126               then
127                 client:send(string.sub(buff1,1461,(#buff1 > 2920) and
128                   2920 or #buff1),
129                   function()
130                     if #buff1 > 2920
131                     then
132                       client:send(string.sub(buff1,2921,#buff1),
133                         function()
134                           end
135                         end)
136                       end)
137                     end)
138                   else
139                     client:send('HTTP/1.1 404 Not found\r\n'..
140                       'Content-Type: text/html\r\n'..
141                       'Content-Length: 25\r\n'..
142                       'Connection: close\r\n\r\n'..
143                       '<h1>Page not found !</h1>')
144                     client:close()
145                   end)
146                 end)
147             end)
148           end)
149         -- main loop
150         tmr.alarm(0,1000,1,
151           function()
152             if wifi.sta.getip()==nil
153             then
154               print("wait for IP")
155             else
156               ipa,_,_=wifi.sta.getip()
157               print("IP is ",ipa)

```

```
158         tmr.stop(0)
159         ReadDHT()
160         LoadBuff()
161         print("Server start")
162         StartServer()
163         -- periodic reading of sensors and load buffer
164         tmr.alarm(1,srp*1000,1,
165             function()
166                 ReadDHT()
167                 LoadBuff()
168             end)
169     end
170 end)
171
```