

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  #
4  # Installing:
5  # python.exe -m pip install --upgrade pip Collecting pip
6  # pip install -U pymodbus
7  # pip install requests
8  # pip install idna
9  # https://pymodbustcp.readthedocs.io/en/latest/quickstart/index.html#
10 #
11 #
12 # Modbus Software to read Coils and Inputs from openplcproject
13 # with Raspberry PI
14 # Data are sent to Webserver via http post request
15 #
16 #   pytohon 3 with modbus library
17 #
18 #   Josef Bernhardt Bad Abbach 6.3.2021   josef@bernhardt.de
19 #
20 from pyModbusTCP.client import ModbusClient
21 import time
22 import requests
23
24
25 s1 = "Teststring Input"
26
27 count = 0
28
29 # Webserver at Strato
30 url = 'http://www.bernhardt.de/modbus/sqlabfrage.php'
31
32 #aktion=write&file=messwerte.csv&inhalt=Test
33
34 # Modbus Server PLC Software at Raspberry PI or ESP8266 with Modbus
35 SERVER_HOST = "192.168.178.89"
36 SERVER_PORT = 502
37 SERVER_U_ID = 1
38
39 c = ModbusClient()
40
41 # uncomment this line to see debug message
42 # c.debug(True)
43
44 # define modbus server host, port and unit_id
45 c.host(SERVER_HOST)
46 c.port(SERVER_PORT)
47 c.unit_id(SERVER_U_ID)
48
49 while True:
50     # Data for File outputs.csv
51     sq = "1.99"
52     # Data for File inputs.csv
53     si = "5.55"
54
55     # open or reconnect TCP to modbus server
56     if not c.is_open():
57         if not c.open():
58             print("unable to connect to "+SERVER_HOST+": "+str(SERVER_PORT))
59
60     # if open() is ok, read coils (modbus function 0x01)
61     if c.is_open():
62         # read 8 bits at address 0, store result in regs list
63         bits = c.read_coils(0, 8)
64         # if success display registers
65         if bits:
66             #print("Coils at #0 to 7: "+str(bits))
67             #s1 = str(bits)
68             if bits[0] == 1:
69                 sq = sq + ";1"
70             else:
71                 sq = sq + ";0"
72             if bits[1] == 1:

```

```

73         sq = sq + ";1"
74     else:
75         sq = sq + ";0"
76     if bits[2] == 1:
77         sq = sq + ";1"
78     else:
79         sq = sq + ";0"
80     if bits[3] == 1:
81         sq = sq + ";1"
82     else:
83         sq = sq + ";0"
84     if bits[4] == 1:
85         sq = sq + ";1"
86     else:
87         sq = sq + ";0"
88     if bits[5] == 1:
89         sq = sq + ";1"
90     else:
91         sq = sq + ";0"
92     if bits[6] == 1:
93         sq = sq + ";1"
94     else:
95         sq = sq + ";0"
96     if bits[7] == 1:
97         sq = sq + ";1"
98     else:
99         sq = sq + ";0"
100
101     print (sq)
102
103     dataout = {'aktion':'write','file':'outputs.csv' , 'inhalt': sq }
104     try:
105         r = requests.post(url , dataout)
106         print(r.text)
107     except:
108         print("Post Error")
109
110
111     time.sleep(0.2)
112
113     # read 8 bits at address 0, store result in regs list
114     bits = c.read_discrete_inputs(0, 8)
115     # if success display registers
116     if bits:
117         #print("Inputs ad #0 to 7: "+str(bits))
118         #s1 = str(bits)
119         if bits[0] == 1:
120             si = si + ";1"
121         else:
122             si = si + ";0"
123         if bits[1] == 1:
124             si = si + ";1"
125         else:
126             si = si + ";0"
127         if bits[2] == 1:
128             si = si + ";1"
129         else:
130             si = si + ";0"
131         if bits[3] == 1:
132             si = si + ";1"
133         else:
134             si = si + ";0"
135         if bits[4] == 1:
136             si = si + ";1"
137         else:
138             si = si + ";0"
139         if bits[5] == 1:
140             si = si + ";1"
141         else:
142             si = si + ";0"
143         if bits[6] == 1:
144             si = si + ";1"

```

```
145         else:
146             si = si + ";0"
147         if bits[7] == 1:
148             si = si + ";1"
149         else:
150             si = si + ";0"
151
152     print (si)
153
154
155     # Send Data to Webserver
156     datain = {'aktion': 'write', 'file': 'inputs.csv' , 'inhalt': si}
157     try:
158         r = requests.post(url , datain)
159         print(r.text)
160     except:
161         print ("Post Error")
162
163     # Read input Values from PLC
164
165     """
166         regs = c.read_holding_registers(0, 8)
167         # if success display registers
168         if regs:
169             print("reg ad #0 to 7: "+str(regs))
170     """
171
172     # sleep 0.2 s before next polling
173     time.sleep(0.2)
174
```