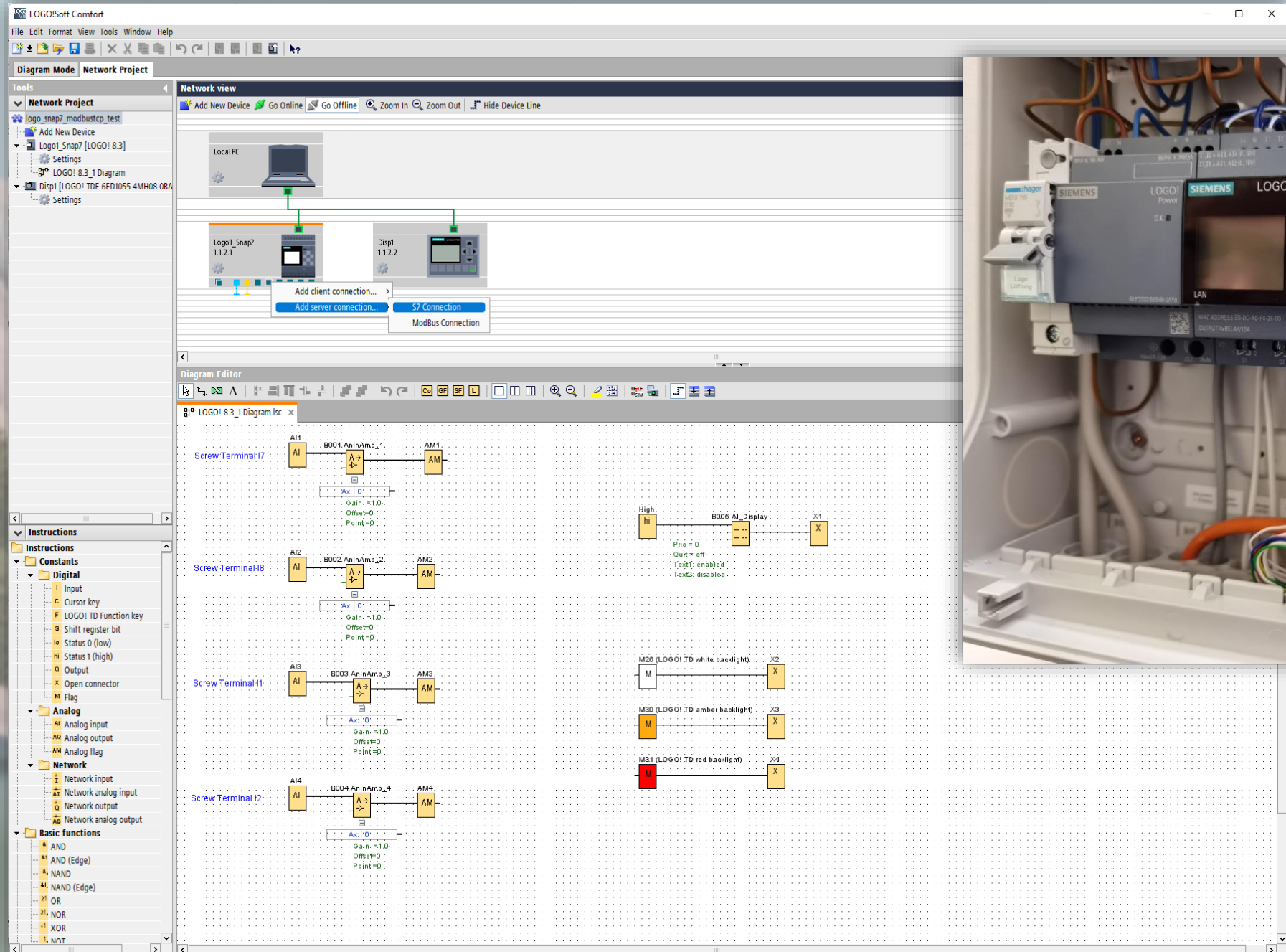


Snap7





S7 Connection

Device name: Logo1_Snap7

IP Address: 1.1.2.1

TSAP: 20.00

Client Server

Connect with an Operator Panel (OP)

Accept all connection request in server side

IP Address: . . .

TSAP: 30.00

Client Server

Data transfer

ID	Address	Length (Bytes)	Direction	Address	Length (Bytes)
1	VB				

Server

Client

```
File Edit Format Run Options Window Help
1 import snap7
2 import time
3
4 plc = snap7.logo.Logo()
5 plc.connect("1.1.2.1", 0x3000, 0x2000)
6
7 if plc.get_connected():
8     print("Connected to LOGO!8")
9     signals = (True, False, False)
10    colors = ("Backlight color: white", "Backlight color: red", "Backlight color: amber")
11    s_splits = 1
12    c_splits = 1
13    while(True):
14        signals = (signals[len(signals) - s_splits:len(signals)] + signals[0:len(signals) - s_splits])
15        colors = (colors[len(colors) - c_splits:len(colors)] + colors[0:len(colors) - c_splits])
16        #print(signals)
17        plc.write("V1107.1", signals[0]) # enable white backlight
18        plc.write("V1107.5", signals[1]) # enable amber backlight
19        plc.write("V1107.6", signals[2]) # enable red backlight
20        print(colors[0])
21        time.sleep(3)
22
23 else:
24     pass
25
26 plc.disconnect()
27 plc.destroy()
28
```


ID	Block	Parameter	Type	Address
1	B001 AnInAmp_1 [Analog Amplifier]	Ax, amplified	Word	0
2	B002 AnInAmp_2 [Analog Amplifier]	Ax, amplified	Word	2
3	B003 AnInAmp_3 [Analog Amplifier]	Ax, amplified	Word	4
4	B004 AnInAmp_4 [Analog Amplifier]	Ax, amplified	Word	6

```

snap7logo_test.py - C:\Users\johanneskinzig\Documents\logo_snap7_modbustcp\snap7logo_test.py (3.9.7)
File Edit Format Run Options Window Help
1 import snap7
2 import time
3
4 plc = snap7.logo.Logo()
5 plc.connect("1.1.2.1", 0x3000, 0x2000)
6
7 if plc.get_connected():
8     print("connected")
9
10 # read analog inputs and AM
11 print("AI1: " + str(plc.read("VW1032"))) # read from AI1
12 print("AI1: " + str(plc.read("VW0"))) # read from B001.AnInAmp_1 - via parameter VM mapping
13 print("AI1: " + str(plc.read("VW118"))) # read from AM1
14
15 print("AI2: " + str(plc.read("VW1034"))) # read from AI2
16 print("AI2: " + str(plc.read("VW2"))) # read from B002.AnInAmp_2 - via parameter VM mapping
17 print("AI2: " + str(plc.read("VW120"))) # read from AM2
18
19 # read digital inputs - bitwise access
20 print("DI1: " + str(plc.read("V1024.0")))
21 print("DI2: " + str(plc.read("V1024.1")))
22 print("DI3: " + str(plc.read("V1024.2")))
23 print("DI4: " + str(plc.read("V1024.3")))
24
25 # write analog outputs and read from to check
26 plc.write("VW1072", 8000)
27 print("AO1: " + str(plc.read("VW1072")))
28
29 # set digital outputs - bitwise access
30 plc.write("V1064.0", True) # write DO 1 - Q1
31 plc.write("V1064.1", True) # write DO 2 - Q2
32 plc.write("V1064.2", True) # write DO 3 - Q3
33 plc.write("V1064.3", True) # write DO 4 - Q4
34
35 # read digital output states - bitwise access
36 print("DO1: " + str(plc.read("V1064.0")))
37 print("DO2: " + str(plc.read("V1064.1")))
38 print("DO3: " + str(plc.read("V1064.2")))
39 print("DO4: " + str(plc.read("V1064.3")))
40
41 time.sleep(2)
42
43 # reset digital outputs
44 plc.write("V1064.0", False) # write DO 1 - Q1
45 plc.write("V1064.1", False) # write DO 2 - Q2
46 plc.write("V1064.2", False) # write DO 3 - Q3
47 plc.write("V1064.3", False) # write DO 4 - Q4
48
49 else:
50     pass
51
52 plc.disconnect()
53 plc.destroy()
54

```