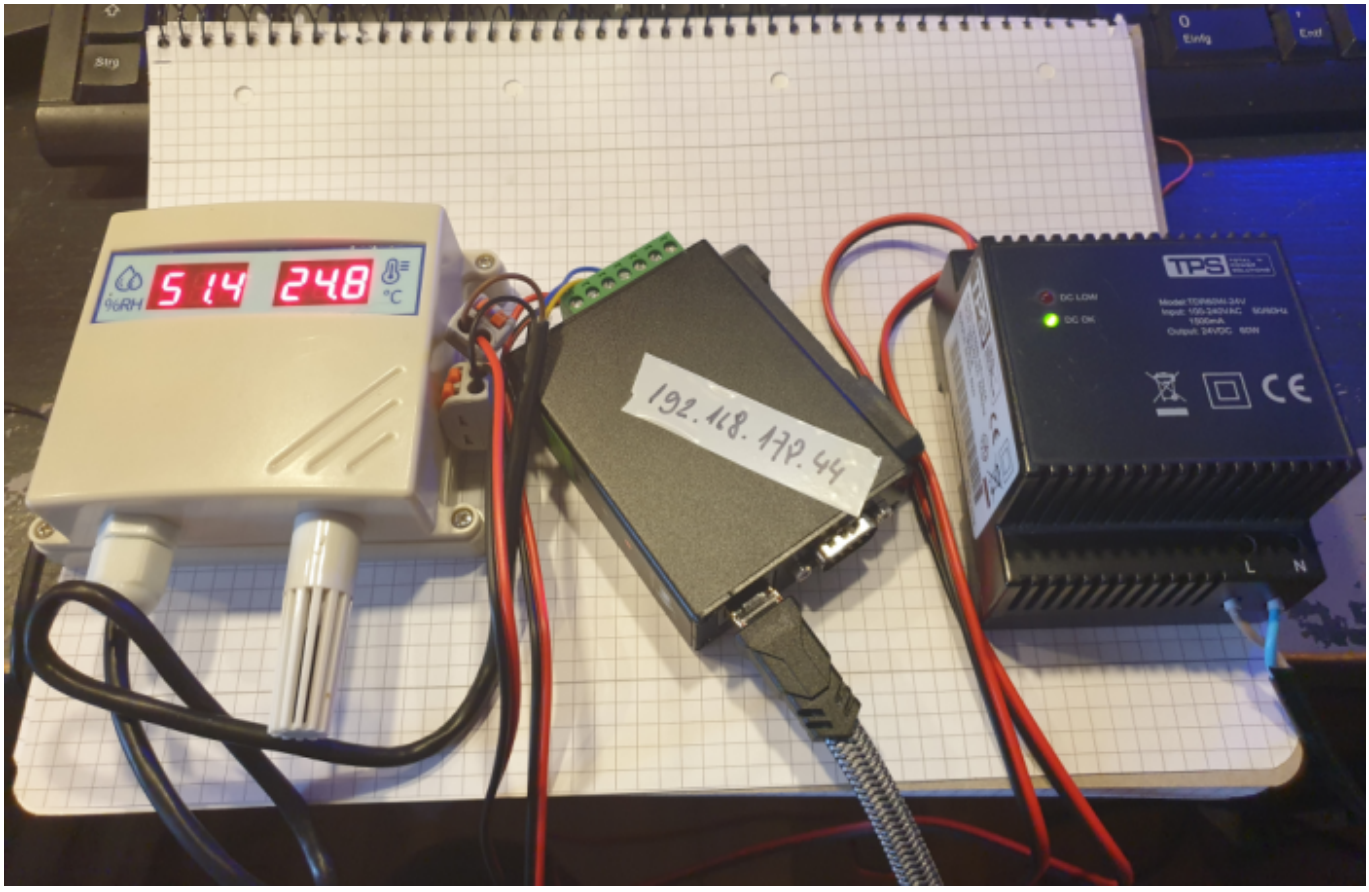
lamaPLC: S7-1500 and UICPAL Temp.humi.sensor Modbus TCP communication

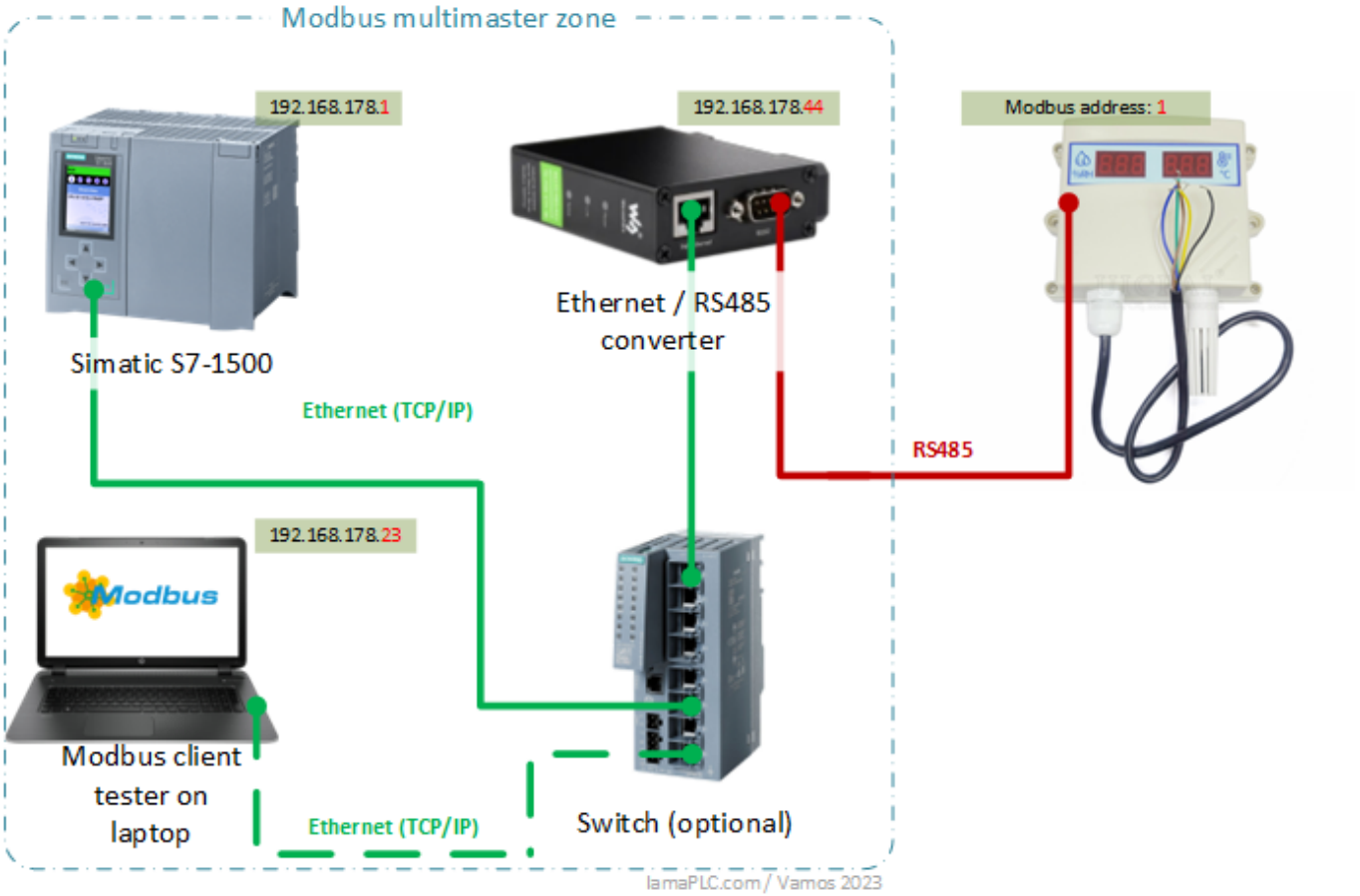
UICPAL's temperature/humidity measuring unit is a fairly cost-effective solution for controlling the air in closed spaces. The unit is also available on Aliexpress, for example, at a price well below 100 euros. The advantage is that it displays the measured values and has a Modbus communication solution. This function can be well integrated with industrial equipment, for example it can be used together with S7 PLC, the solution below describes this possibility.

A more detailed description of the UICPAL temperature / humidity meter can be found here.

Unfortunately, the unit (as far as I know) does not have a “CE” certification, which can make its application difficult in some cases.



The UICPAL unit and the S7-1500



The UICPAL unit has RTU communication, which should be connected with an ethernet/RS485 converter as a cost-effective solution. This converter is now available directly from Simatic. It is worth starting the testing with a Modbus client tester program. If UICPAL is available from here, then there will be no major problem with Simatic.

The steps for setting up communication:

1. Switching on UICPAL, wiring with the ethernet/RS485 converter (A TX+, B RX-). The converter is described here. UICPAL is described here.
2. Connecting the laptop to the ethernet/RS485 converter
3. ethernet/RS485 converter settings:
4. Setting the IP address (in the example: 192.168.178.44)
5. Setting the RS485 communication parameters: 8n1, speed: 9600 baud, addr: 1
6. Testing communication from the laptop. If the unit is not available, review the parameters and possibly replace the Modbus wires
7. If the Modbus communication works from the laptop; disconnect.
8. Uploading the PLC program, testing the PLC communication.

The S7 programm for communication

```
#ston(IN:=NOT(#stonb),
      PT:=t#2s);

#stonb := #ston.Q;

IF #stonb THEN
    IF #start THEN
        #start := FALSE;
    ELSE
        #start := TRUE;
    END_IF;
    "mbRead".liveBeat := "mbRead".liveBeat + 1;
END_IF;

//#start := FALSE;

"MB_CLIENT_DB".MB_Unit_ID := 1;

"MB_CLIENT_DB"(REQ := #start,
               DISCONNECT := NOT(#start),
               MB_MODE := 0,
               MB_DATA_ADDR := 40001,
               MB_DATA_LEN := 10,
               DONE=>#done,
               BUSY=>#busy,
               ERROR=>#error,
               STATUS=>#status,
               MB_DATA_PTR := P#DB6.DBX2.0 WORD 10,
               CONNECT := "ModbusComm".comm_client);

#temp := "mbRead".data[1] / 10.0;

#hum := "mbRead".data[0] / 10.0;
```

#stonb	FALSE
#stonb	FALSE
Result	FALSE
Result	TRUE
#start	FALSE
#start	TRUE
"mbRead".liveBeat	16#82
"MB_CLIENT_DB".MB_...	16#01
#start	FALSE
#start	FALSE
#done	FALSE
#busy	FALSE
#error	FALSE
#status	16#7000
#temp	25.2
#hum	50.3

The sensor communicates slowly (9600 baud), time must be allowed for transmission. The program runs with a 2-second clock, the data is updated every 4 seconds.

It is important that the address of the sensor must also be set in the S7 program (*“MB_CLIENT_DB”.MB_Unit_ID := 1;*)!
The program uses Modbus TCP communication, where the address is irrelevant. This communication, however, is Modbus RTU (after the ethernet/RS485 converter), where the address must always be entered.

mbRead

	Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...
1	Static					☐	☐	☐	☐
2	liveBeat	Byte	0.0	16#0	16#67	☐	☑	☑	☑
3	data	Array[0..9] of Word	2.0			☐	☑	☑	☑
4	data[0]	Word	2.0	16#0	16#01F8	☐	☑	☑	☑
5	data[1]	Word	4.0	16#0	16#00FD	☐	☑	☑	☑
6	data[2]	Word	6.0	16#0	16#0000	☐	☑	☑	☑
7	data[3]	Word	8.0	16#0	16#0000	☐	☑	☑	☑
8	data[4]	Word	10.0	16#0	16#0000	☐	☑	☑	☑
9	data[5]	Word	12.0	16#0	16#0000	☐	☑	☑	☑
10	data[6]	Word	14.0	16#0	16#0000	☐	☑	☑	☑
11	data[7]	Word	16.0	16#0	16#0000	☐	☑	☑	☑
12	data[8]	Word	18.0	16#0	16#0000	☐	☑	☑	☑
13	data[9]	Word	20.0	16#0	16#0201	☐	☑	☑	☑

The SLC code contains all FBs, FCs and DBs required for the operation. The code was developed with TIA version 16. I cannot take responsibility for its use.

This code ensures communication for this equipment only. Good luck with your application! The code contains no restrictions, restrictions, 100% free!

If you need a modified or different code, find me here!

If you just want to throw me a coffee tip, you can do it here:

If you liked these codes you could buy me some coffee over paypal!

If you are a manufacturer and would like an example program similar to your product, you can find me here!

The S7 program can be downloaded here, it must be exported to the TIA portal, you can find a description of this here.

S7-UICPAL Modbus communication download

The program includes the following parts:

- testFB (FB): calling the program
- mbRead (DB6!): the data is read here (if the address is not DB6, the addressing must be changed in testFB at *MB_DATA_PTR*)
- ModbusComm (DB): Modbus communication settings, the address of the converter must be set here.

Important: mbRead cannot be an optimized DB due to direct addressing, you can find information about this setting here.

If the status stays in the 700x range and changes continuously, then communication is working (Otherwise, you can find more information about status codes here).

bus, communication, s7, simatic, s7 1500, s7 1200, scl, uicpal, temperature, humidity, modbus, example, download, tia portal

This page has been accessed for: Today: 3, Until now: 112