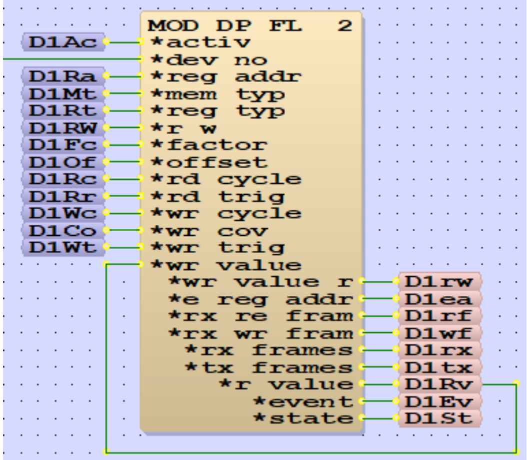
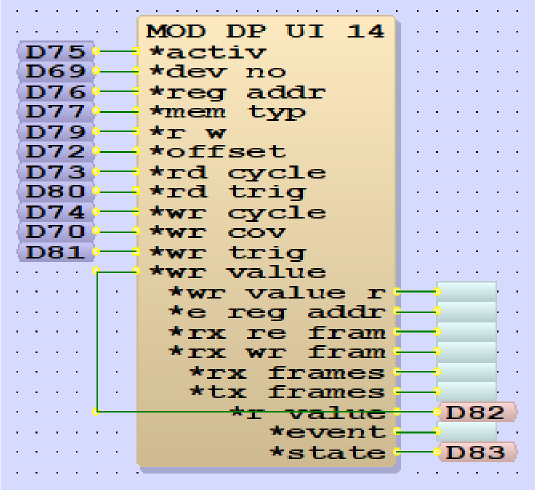
TT230502 – OFXL - Modbus Module UI and ULI

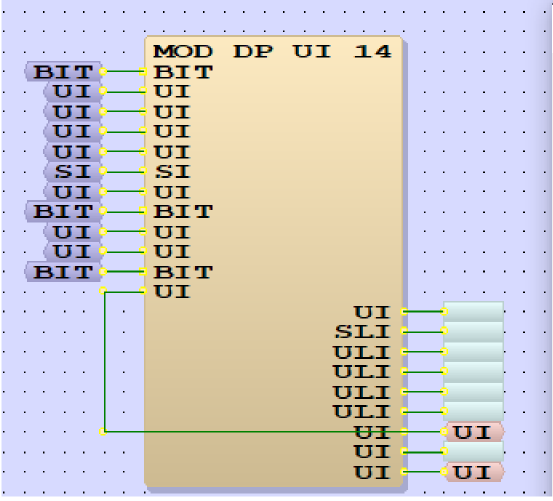
1. In this document, we will show you another 2 Modbus modules in OFXL 4. In most of the cases, the “MOD_DP_FL” module are suitable for basically all applications in Modbus communication. But there are still some reasons that you can’t use them in some applications.

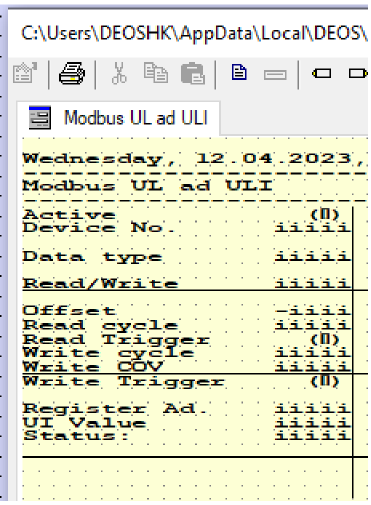


2. One of the reasons is that you need to do some bitwise read/write operation from a UI point. In this case we don’t want to convert the UI to FL as in the “MOD_DP_FL” module. So, we use the “MOD_DP_UI” module.



3. Basically, it’s almost the same as the “FL” module, but you need to change some of the types to “SI” and “UI”, like below.

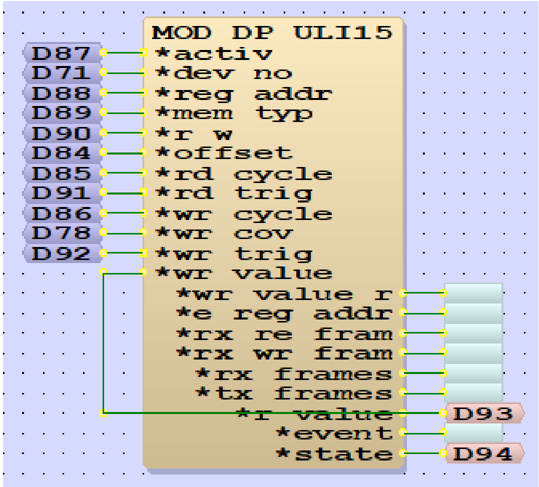



C:\Users\DEOSHK\AppData\Local\DEOS\

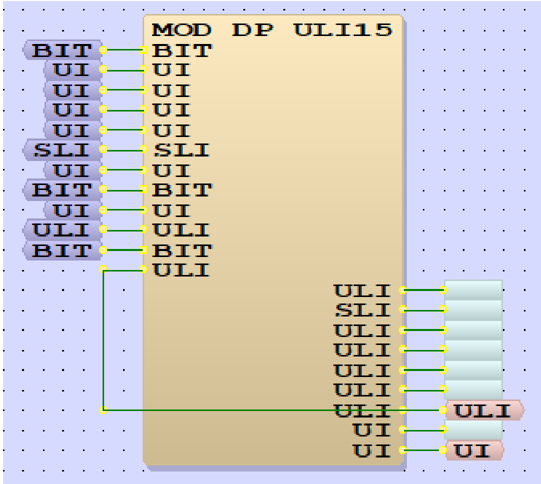
Modbus UL ad ULI

```
Wednesday, 12.04.2023,
----------------------
Modbus UL ad ULI
----------------------
Active                (▯)
Device No.          iiiii

Data type           iiiii

Read/Write          iiiii

Offset              -iiii
Read cycle          iiiii
Read Trigger          (▯)
Write cycle         iiiii
Write COV           iiiii
Write Trigger         (▯)

Register Ad.        iiiii
UI Value            iiiii
Status:             iiiii
```

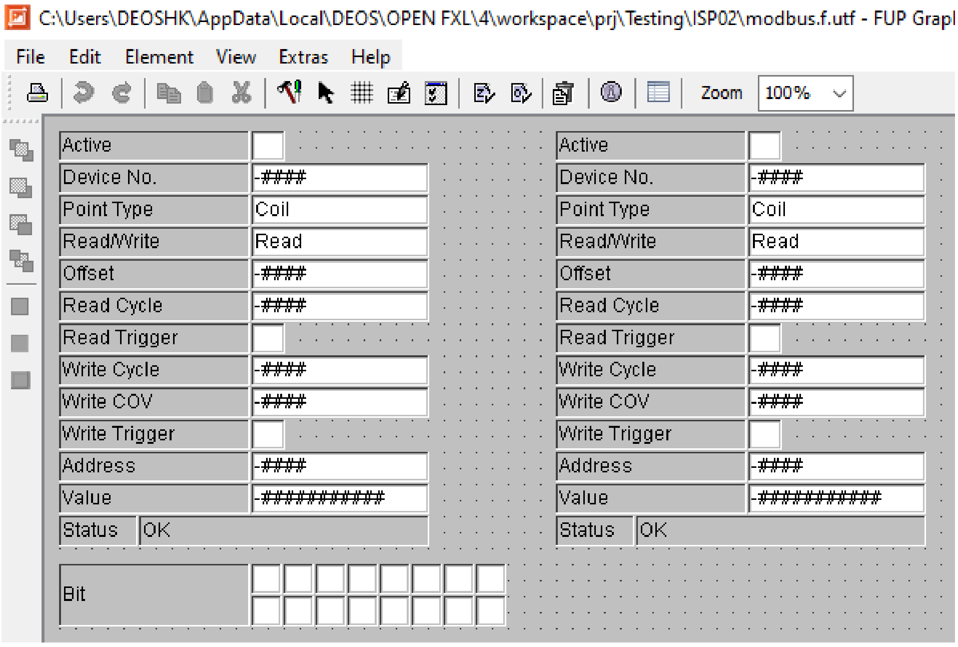
4. Another reason is that you need to read a ULI (32-bits integer) point (e.g. kWh) and you don’t want to convert it to type “FL”, because the accuracy will become lower after conversion. In this case, we use the “MOD_DP_ULI” module.



5. Also, if you need to write to an ULI point, then you MUST use this module, as the FL conversion in the “MOD_DP_FL” module will write an incorrect value to the Modbus device because of the type conversion. Again, you need to change some point types to SLI and ULI.



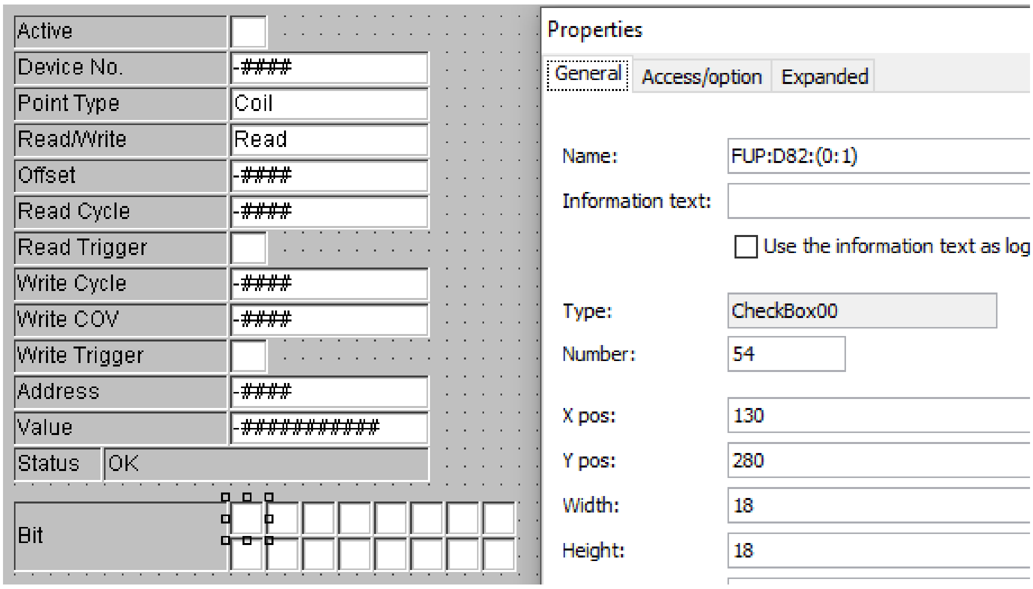
6. Finally, we build the graphic page like below. This time we add some checkbox to show how we can read/write a specific bit using the “MOD_DP_UI” module.


C:\Users\DEOSHK\AppData\Local\DEOS\OPEN FXL\4\workspace\prj\Testing\ISP02\modbus.f.utf - FUP Grapl

Active	□	Active	□
Device No.	-####	Device No.	-####
Point Type	Coil	Point Type	Coil
Read/Write	Read	Read/Write	Read
Offset	-####	Offset	-####
Read Cycle	-####	Read Cycle	-####
Read Trigger	□	Read Trigger	□
Write Cycle	-####	Write Cycle	-####
Write COV	-####	Write COV	-####
Write Trigger	□	Write Trigger	□
Address	-####	Address	-####
Value	-###########	Value	-###########
Status	OK	Status	OK

Bit □ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □

7. We create a “CheckBox00” in graphic, and link it to the UI “Input”. Then we type in “:(0:1)” behind it, meaning that we read the UI “Input” from Bit 0 for 1 bit. For the others, use “:(1:1)”, “:(2:1)” and “:(3:1)”, etc.



Active	□
Device No.	-####
Point Type	Coil
Read/Write	Read
Offset	-####
Read Cycle	-####
Read Trigger	□
Write Cycle	-####
Write COV	-####
Write Trigger	□
Address	-####
Value	-###########
Status	OK
Bit	□□□□□□□□ □□□□□□□□

Properties

General | Access/option | Expanded

Name: FUP:D82:(0:1)

Information text:

□ Use the information text as log

Type: CheckBox00

Number: 54

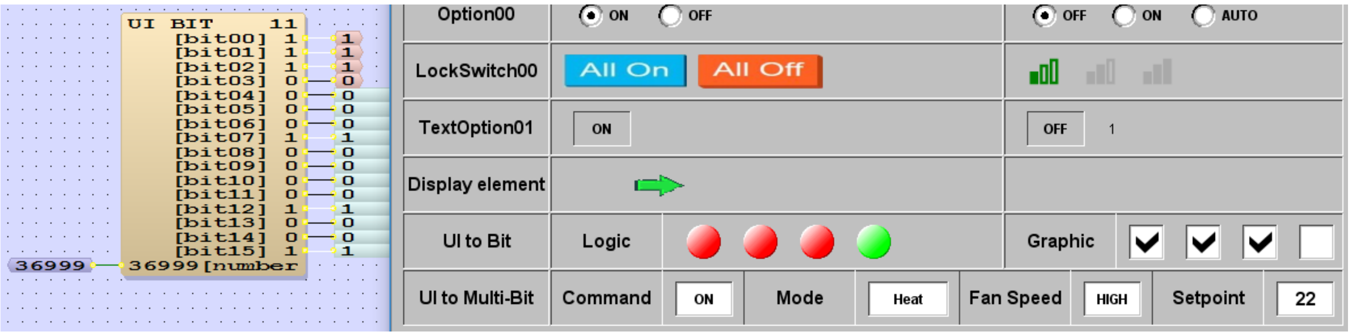
X pos: 130

Y pos: 280

Width: 18

Height: 18

8. Please refer to TT190703 for more information and also other ways to do bitwise operations in FUP and graphic page for UI point.



```
UI BIT          11
       [bit00] 1 — 1
       [bit01] 1 — 1
       [bit02] 1 — 1
       [bit03] 0 — 0
       [bit04] 0 — 0
       [bit05] 0 — 0
       [bit06] 0 — 0
       [bit07] 1 — 1
       [bit08] 0 — 0
       [bit09] 0 — 0
       [bit10] 0 — 0
       [bit11] 0 — 0
       [bit12] 1 — 1
       [bit13] 0 — 0
       [bit14] 0 — 0
       [bit15] 1 — 1
36999 — 36999[number
```

Option00: ● ON ○ OFF | ● OFF ○ ON ○ AUTO

LockSwitch00: All On | All Off

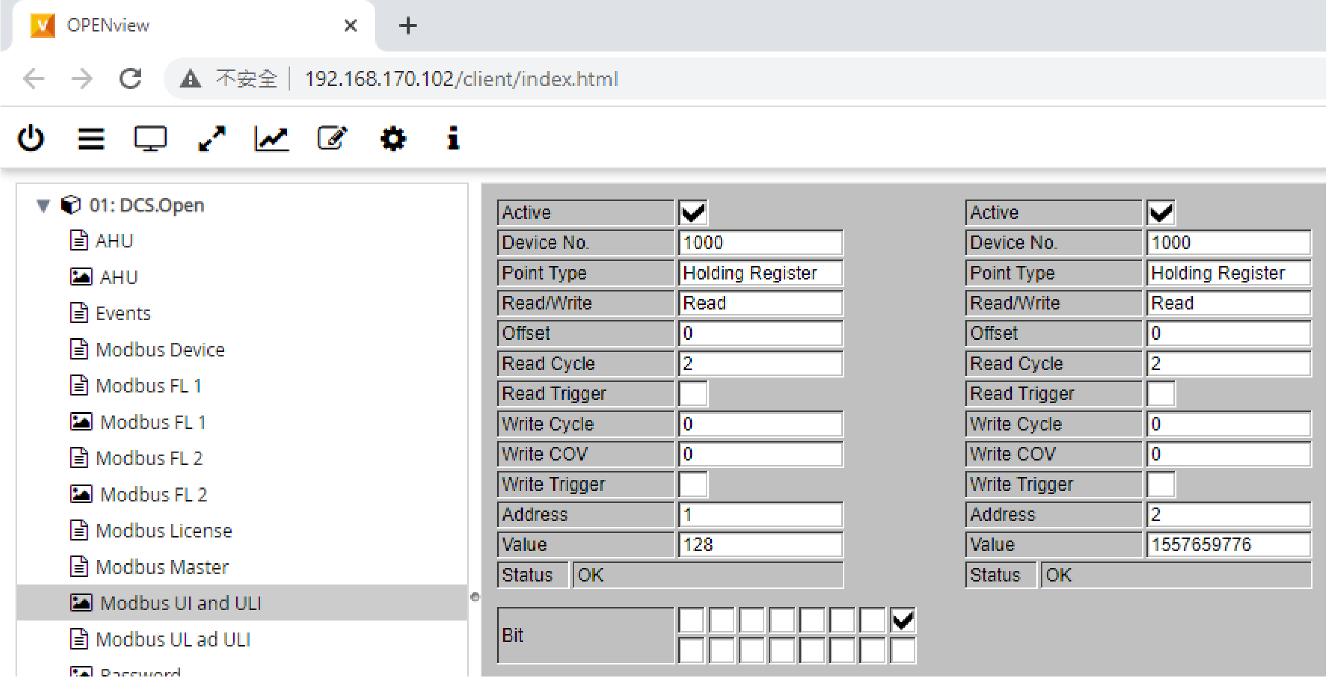
TextOption01: ON | OFF 1

Display element

UI to Bit: Logic ● ● ● ● | Graphic ☑ ☑ ☑ □

UI to Multi-Bit: Command ON | Mode Heat | Fan Speed HIGH | Setpoint 22

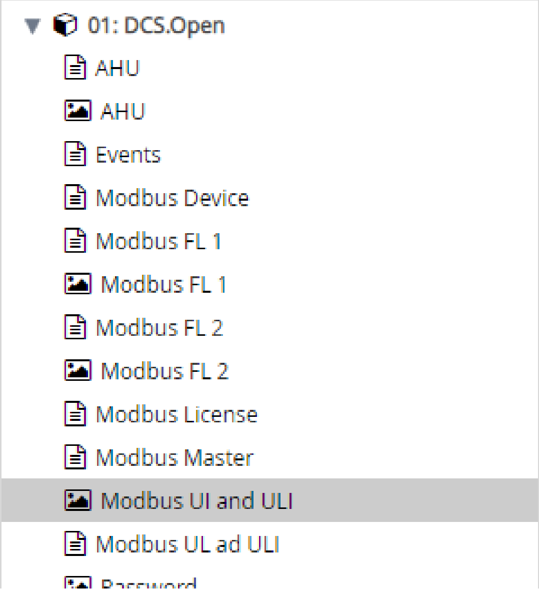
9. Now create and load the new program to your controller. First, set the correct “Device No.” and the addresses for the UI and ULI points, and you should see the correct values from the Modbus device, like below. The specific bits will change to 1 based on the UI value.

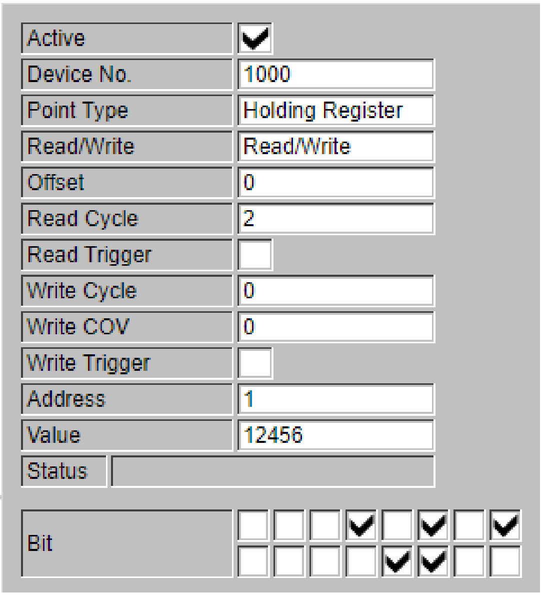


- 01: DCS.Open
  - AHU
  - AHU
  - Events
  - Modbus Device
  - Modbus FL 1
  - Modbus FL 1
  - Modbus FL 2
  - Modbus FL 2
  - Modbus License
  - Modbus Master
  - Modbus UI and ULI
  - Modbus UL ad ULI
  - Password

Active	☑	Active	☑
Device No.	1000	Device No.	1000
Point Type	Holding Register	Point Type	Holding Register
Read/Write	Read	Read/Write	Read
Offset	0	Offset	0
Read Cycle	2	Read Cycle	2
Read Trigger	□	Read Trigger	□
Write Cycle	0	Write Cycle	0
Write COV	0	Write COV	0
Write Trigger	□	Write Trigger	□
Address	1	Address	2
Value	128	Value	1557659776
Status	OK	Status	OK
Bit	□□□□□□□☑ □□□□□□□□		

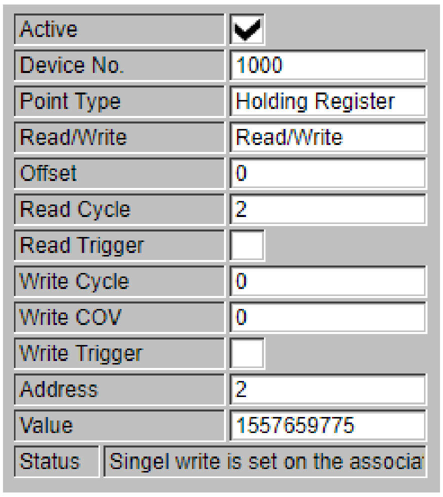
10. Set the UI to “Read/Write” and then you can change the value directly and see the corresponding bits enable/disable. You can also control each bit individually to 0 or 1 by clicking the corresponding checkbox.





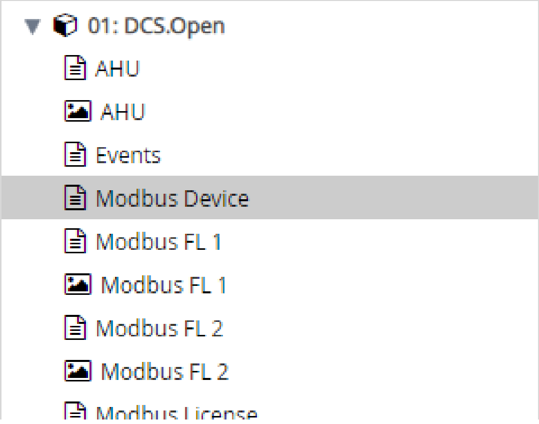
Active	☑
Device No.	1000
Point Type	Holding Register
Read/Write	Read/Write
Offset	0
Read Cycle	2
Read Trigger	□
Write Cycle	0
Write COV	0
Write Trigger	□
Address	1
Value	12456
Status	
Bit	□ □ □ ☑ □ ☑ □ ☑ □ □ □ □ ☑ ☑ □ □

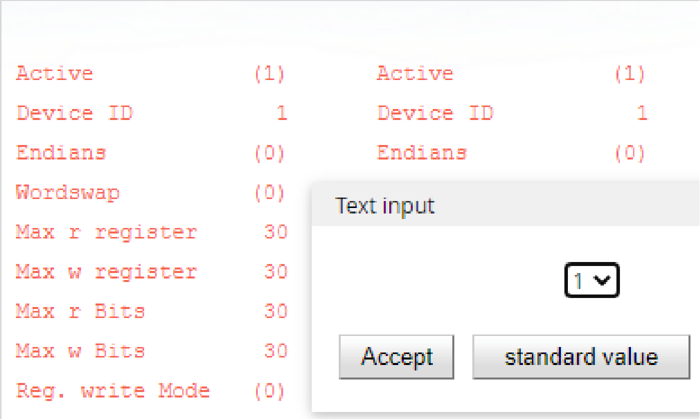
11. Now if change the ULI point to “Read/Write”, you may get an error in “Status” saying that “Single write is set...”.



Active	☑
Device No.	1000
Point Type	Holding Register
Read/Write	Read/Write
Offset	0
Read Cycle	2
Read Trigger	□
Write Cycle	0
Write COV	0
Write Trigger	□
Address	2
Value	1557659775
Status	Singel write is set on the associa

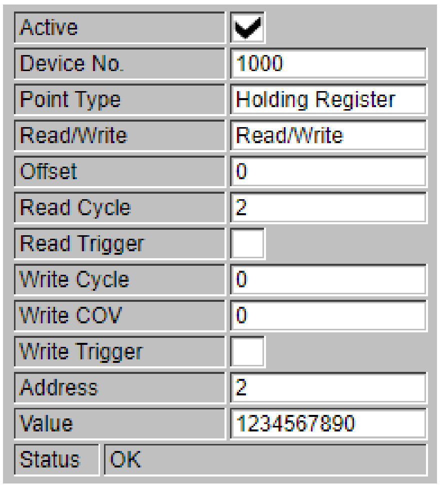
12. To fix it, go to the “Modbus Device” page and set “Reg. Write Mode” to 1.





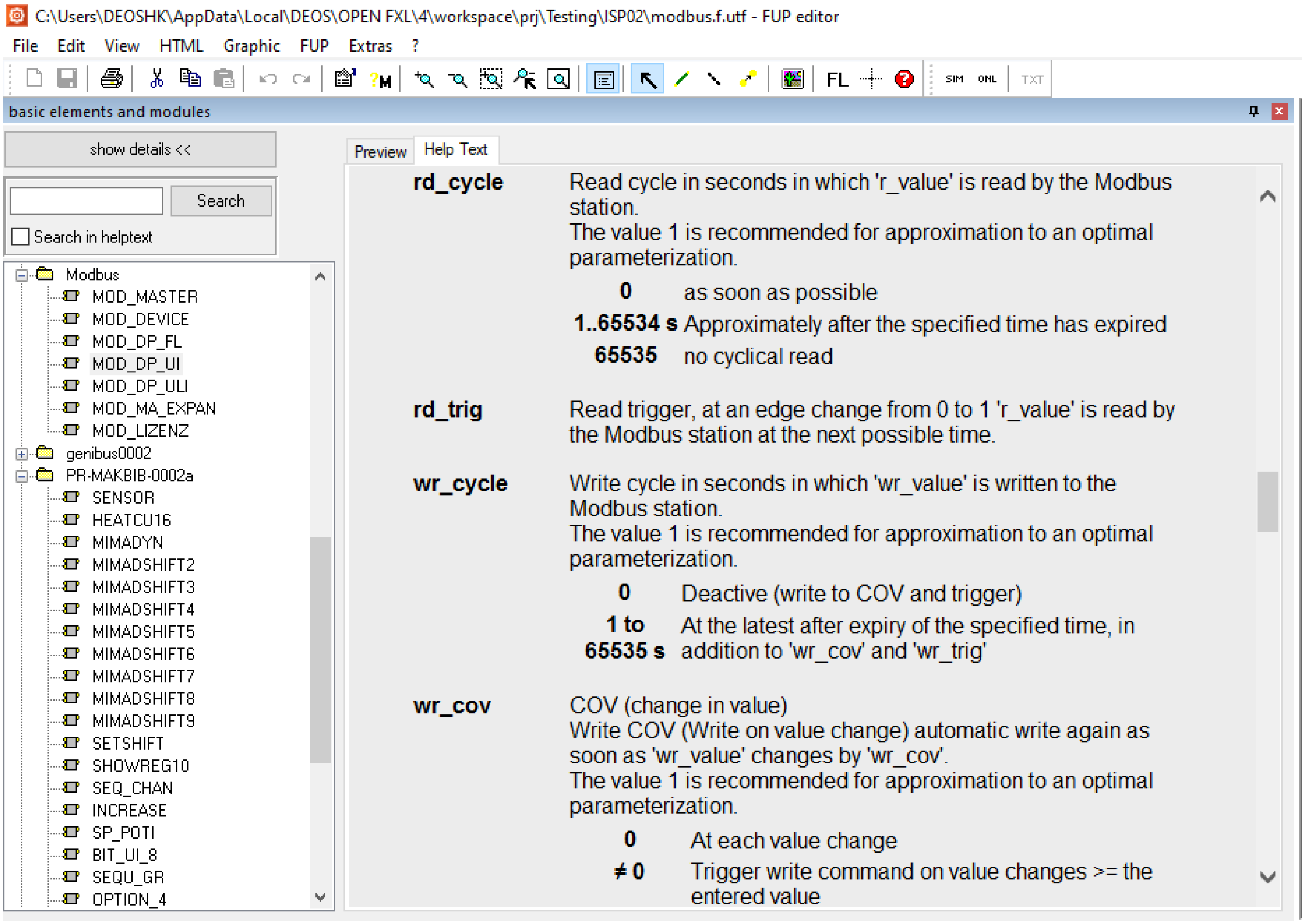
Active	(1)	Active	(1)
Device ID	1	Device ID	1
Endians	(0)	Endians	(0)
Wordswap	(0)		
Max r register	30		
Max w register	30		
Max r Bits	30		
Max w Bits	30		
Reg. write Mode	(0)		

13. After that you should be able to write to the ULI (32-bits Integer) point successfully.



Active	☑
Device No.	1000
Point Type	Holding Register
Read/Write	Read/Write
Offset	0
Read Cycle	2
Read Trigger	□
Write Cycle	0
Write COV	0
Write Trigger	□
Address	2
Value	1234567890
Status	OK

14. You can always refer to the “Help Text” for more information regarding the modules. These new Modbus modules provides much more flexibility to many of the Modbus settings and also can be changed online easily.



rd_cycle Read cycle in seconds in which 'r_value' is read by the Modbus station.
The value 1 is recommended for approximation to an optimal parameterization.

0	as soon as possible
1..65534 s	Approximately after the specified time has expired
65535	no cyclical read

rd_trig Read trigger, at an edge change from 0 to 1 'r_value' is read by the Modbus station at the next possible time.

wr_cycle Write cycle in seconds in which 'wr_value' is written to the Modbus station.
The value 1 is recommended for approximation to an optimal parameterization.

0	Deactive (write to COV and trigger)
1 to 65535 s	At the latest after expiry of the specified time, in addition to 'wr_cov' and 'wr_trig'

wr_cov COV (change in value)
Write COV (Write on value change) automatic write again as soon as 'wr_value' changes by 'wr_cov'.
The value 1 is recommended for approximation to an optimal parameterization.

0	At each value change
≠ 0	Trigger write command on value changes >= the entered value