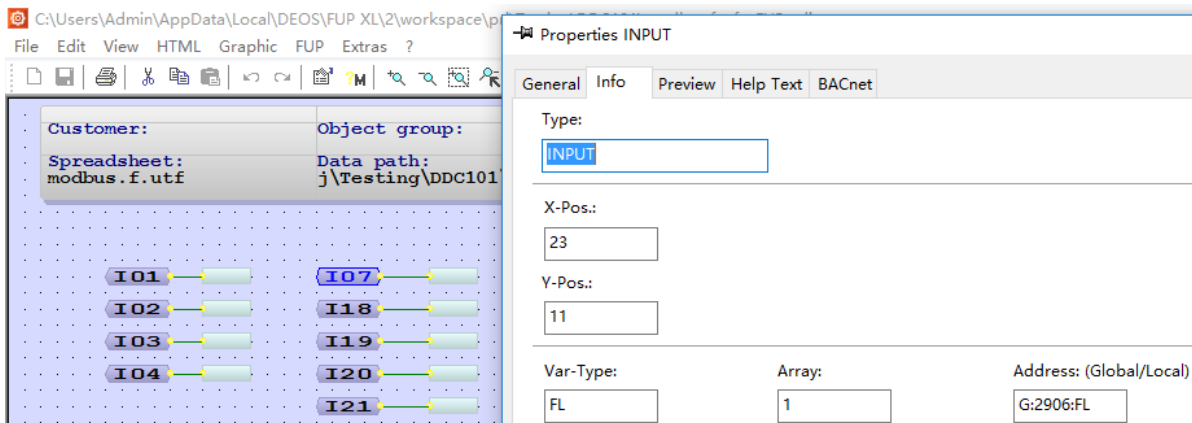
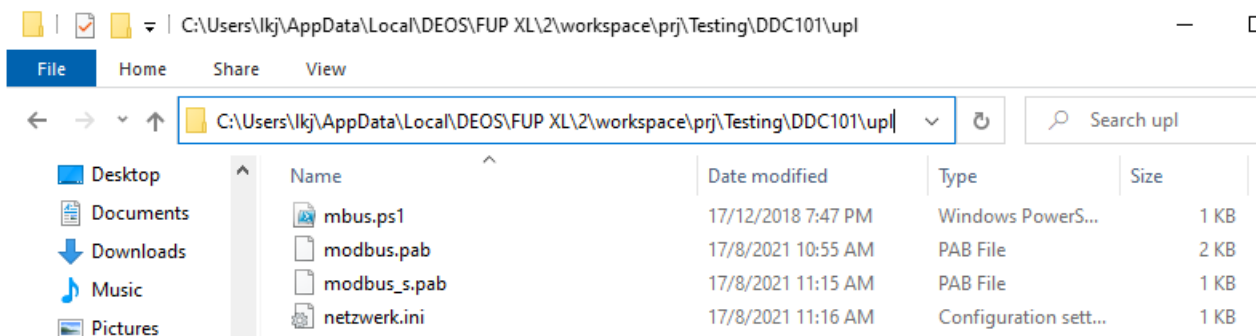


TT210904 – FUP - Modbus Slave Virtual Memory

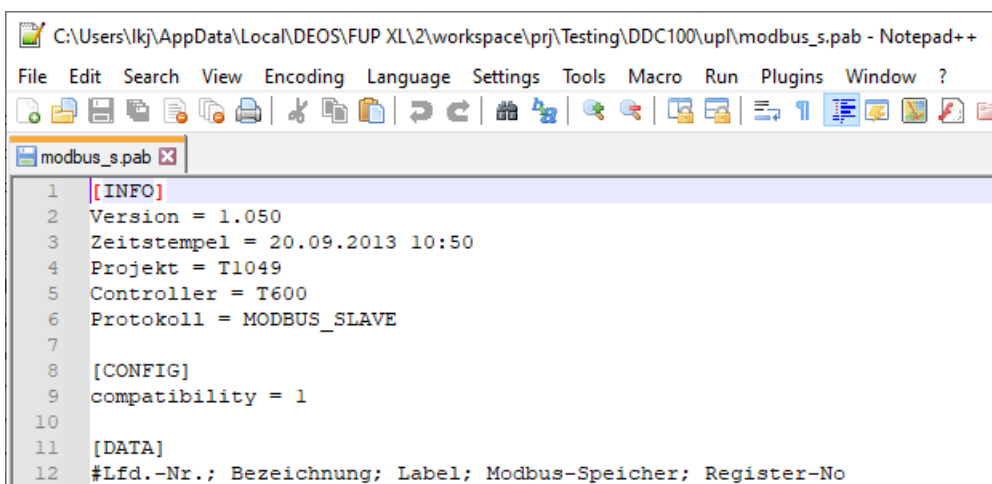
1. To setup Modbus slave function in OPEN controller, you can refer to TT190803. In this document, we will show you how to set the Modbus register address manually.
2. By default, the Modbus register address is set automatically in FUP. This is the easiest way to setup Modbus slave function, but sometimes it's not very good because the address is set "randomly". Also, 3rd party Modbus master may accidentally overwrite the settings in OPEN controller because all data are exposed with read/write by default.



3. To resolve the issue, all OPEN controllers, which are configured as Modbus slave can be addressed indirectly via "virtual memory". This allows you to set only the required "Input" and "Display" as Modbus registers and assign the address manually according to your needs.
4. To do this, you have to edit the "modbus_s.pab" file in the controller manually.
5. The file is located at "C:\Users\xxx\AppData\Local\DEOS\FUP XL\2\workspace\prj\yyy\zzz\upl", where "xxx" is your Windows user name, "yyy" and "zzz" is your FUP project and controller name. Below is an example for your reference.



6. To edit the file, start "Notepad++". (Do not use notepad or any other text editor to change it).



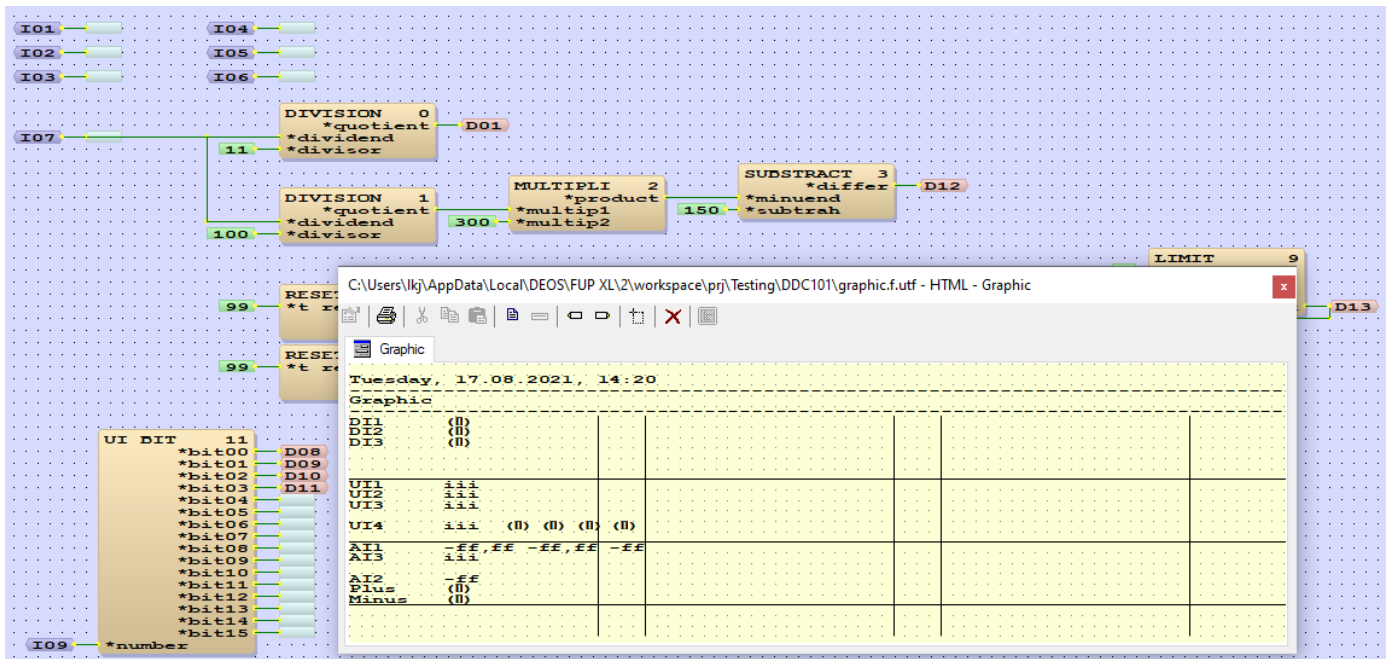
7. First of all, change the “compatibility” from 1 to 0 (under [CONFIG]). The tell the system that you want to set the Modbus address manually (i.e. virtual memory is to be used)

```

modbus_s.pab x
1  [INFO]
2  Version = 1.050
3  Zeitstempel = 20.09.2013 10:50
4  Projekt = T1049
5  Controller = T600
6  Protokoll = MODBUS_SLAVE
7
8  [CONFIG]
9  compatibility = 0
10
11 [DATA]
12 #Lfd.-Nr.; Bezeichnung; Label; Modbus-Speicher; Register-No

```

8. In this example, we’re going to map the points in the below FUP page (graphic.f) as Modbus registers.



9. Now you can set the Modbus register address under [DATA]. The first point to map is “I01” which has point type “BIT” (digital input or output) and have the name “DI1” in HTML page. Add the following information at the end, like this.

```

11 [DATA]
12 #Lfd.-Nr.; Bezeichnung; Label; Modbus-Speicher; Register-No
13 0;DI1;graphic.f:I01;COIL;0

```

10. The format is explained below. They are separated by semicolon “;”.

- The first “0” means this is the first point in the Modbus slave configuration
- “DI1” is the name which is for reference only (so we use the name in HTML page)
- “graphic.f:I01” is the FUP page name and the “Input” name (or “Display” name)
- “COIL” is the Modbus register type
- The last “0” is the Modbus register address we want to use

11. Now add the second point “I02” as Modbus address “1”. Please note that our Modbus address start from “0”, “1”, “2”, etc.

```

11 [DATA]
12 #Lfd.-Nr.; Bezeichnung; Label; Modbus-Speicher; Register-No
13 0;DI1;graphic.f:I01;COIL;0
14 1;DI2;graphic.f:I02;COIL;1

```

12. For the 3rd point "I03", we use the Modbus type "DISCRETE_INPUT" and set it to address "0", because we want it to be read only.

```
11 [DATA]
12 #Lfd.-Nr.; Bezeichnung; Label; Modbus-Speicher; Register-No
13 0;DI1;graphic.f:I01;COIL;0
14 1;DI2;graphic.f:I02;COIL;1
15 2;DI3;graphic.f:I03;DISCRETE_INPUT;0
```

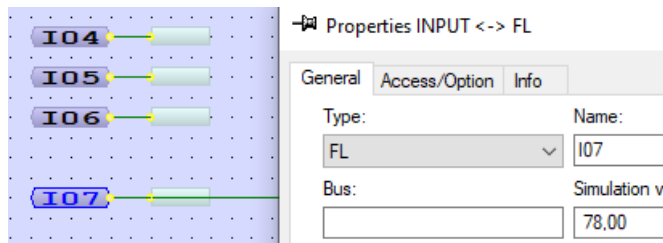
13. For analog points (type UI or SI), we use Modbus type "HOLDING_REGISTER" for read/write. So we add 3 lines below for the point "I04" to "I06", with address from 0 to 2.

```
11 [DATA]
12 #Lfd.-Nr.; Bezeichnung; Label; Modbus-Speicher; Register-No
13 0;DI1;graphic.f:I01;COIL;0
14 1;DI2;graphic.f:I02;COIL;1
15 2;DI3;graphic.f:I03;DISCRETE_INPUT;0
16 3;UI1;graphic.f:I04;HOLDING_REGISTER;0
17 4;UI2;graphic.f:I05;HOLDING_REGISTER;1
18 5;UI3;graphic.f:I06;HOLDING_REGISTER;2
```

14. Now add the last 2 holding registers for point "I07" and "I09" with address 3 and 5.

```
11 [DATA]
12 #Lfd.-Nr.; Bezeichnung; Label; Modbus-Speicher; Register-No
13 0;DI1;graphic.f:I01;COIL;0
14 1;DI2;graphic.f:I02;COIL;1
15 2;DI3;graphic.f:I03;DISCRETE_INPUT;0
16 3;UI1;graphic.f:I04;HOLDING_REGISTER;0
17 4;UI2;graphic.f:I05;HOLDING_REGISTER;1
18 5;UI3;graphic.f:I06;HOLDING_REGISTER;2
19 6;AI1;graphic.f:I07;HOLDING_REGISTER;3
20 7;UI4;graphic.f:I09;HOLDING_REGISTER;5
```

15. Please note that the point "I07" has type "FL" which is a 32-bits floating point number. This point use 2 Modbus register addresses (3 and 4), so you can see from above that address "4" is skipped. The same apply for types "ULI" and "SLI".



16. Finally, we add "D13" as "INPUT_REGISTER" because we want it to be read only, with address "0". So, this is the completed configuration. Save it and close the program.

```
modbus_s.pab x
1 [INFO]
2 Version = 1.050
3 Zeitstempel = 20.09.2013 10:50
4 Projekt = T1049
5 Controller = T600
6 Protokoll = MODBUS_SLAVE
7
8 [CONFIG]
9 compatibility = 0
10
11 [DATA]
12 #Lfd.-Nr.; Bezeichnung; Label; Modbus-Speicher; Register-No
13 0;DI1;graphic.f:I01;COIL;0
14 1;DI2;graphic.f:I02;COIL;1
15 2;DI3;graphic.f:I03;DISCRETE_INPUT;0
16 3;UI1;graphic.f:I04;HOLDING_REGISTER;0
17 4;UI2;graphic.f:I05;HOLDING_REGISTER;1
18 5;UI3;graphic.f:I06;HOLDING_REGISTER;2
19 6;AI1;graphic.f:I07;HOLDING_REGISTER;3
20 7;UI4;graphic.f:I09;HOLDING_REGISTER;5
21 8;AI2;graphic.f:D13;INPUT_REGISTER;0
```

17. Now you can upload it to the controller for testing. You don't need to compile the controller again.
18. Note: Please make sure you make backup of the "modbus_s.pab" file after making changes, and put it in another directory. This is VERY important as a backup.
19. Make sure you've enabled the Modbus slave communication in controller. You can then test it with 3rd party Modbus program (e.g. Modbus Poll).

The screenshot shows two windows. The left window is 'OPENview' displaying a web interface for a Modbus slave. The right window is 'Modbus Poll - [Mbpoll1]' showing a table of data points.

OPENview Data:

Variable	Value
DI1	(0)
DI2	(0)
DI3	(1)
UI1	2
UI2	0
UI3	2
UI4	38016 (0) (0) (0) (0)
AI1	25.68 2.33 -73

Modbus Poll Data Table:

Address	Alias	Value
0		2
1		0
2		2
3		25.68
4		--
5		38016
6		0
7		0
8		0
9		0

20. If you made some mistake in the "modbus_s.pab" file, and some points are not working, you can check the error at http://xxx/tmp/modbus_s.log, where "xxx" is your controller IP address. In the below example, the error is because of the overlapping of the Modbus register address "4".

```

Checking for Register Type Definition:
Checking for Register Type Definition:

Checking for duplicates:

Checking for overlap:
Register number = 4 ; Element size = 2 ; Memory type = HOLDING_REGISTER ; OVERLAP

```