Application Note
No. 4

Product: Softing Modbus TCP OPC Server

Keywords: Modbus OPC configuration

Problem:

How to configure the Softing Modbus OPC-Server to get access to a Modbus device?
This configuration is done with an alias file. Wile creating this file one has to take care about several things.
This how-to will show how to configure the Aliasfile for a Modbus I/O Controller.

Solution:

For a correct configuration of the alias file it is important to find the correct Modbus address from the documentation of the used Modbus device.

Difference between COIL ←→ REGISTER:

There are two options to access data on a Modbus device,

- COIL: a coil represents one Bit on the Modbus device. This may be one single digital input or output. In this sample we will use 2 digital inputs and 2 digital outputs on the Modbus controller.

- REGISTER: a register represents a complex data type which may consist of one or more bytes. This may be a collection of 8 in/outputs or also 2 bytes for an analogue value. In this sample we will use the first byte on the controller which represents the first 8 digital inputs/outputs (the same we are using for the coil).

In Modbus the first digit of the address is used to specify the mode of access (Coil/Register, Input/Output). The help file of the Softing Modbus server shows this (Help -> Operation → OPC Namespace → Syntax Items)

<Modbus/TCP Address>

The Modbus/TCP Address of the value that should be accessed.

Modbus/TCP defines 4 address areas:

- Digital (Coil) Output - 0
- Digital (Coil) Input - 1
- Register Input - 3
- Register Output - 4

The Modbus/TCP address starts with the address area (0, 1, 3 or 4). After that stands an offset within the address area. This offset starts at 1 and ends at 65536.

Finding the offset in the device documentation:

To create a correct configuration line we have to find the correct offset. This can be found in the documentation of the used Modbus device.

Note:

Some devices represent the same data multiple times in the address space (like in this controller ➔ register write 0-255 and 512-767, ➔ bit write 0-511 and 512-1023).
One of these areas may be used for special features of the Modbus controller, please check the device description.

The documentation of the used controller shows:

Bit access, reading:

Modbus Address		Memory area	Description
Decimal	Hex		
0 ... 511	0x0000 … 0x01FF	Physical Output Area	The first 512 inputs
512 … 1023	0x0200 … 0x03FF	Physical Input Area	The first 512 outputs
1024 …	…..	…..	…..

→ The offsets for reading coil access:
Inputs: 0-511
Outputs: 512-1023

Bit access, writing:

Modbus Address		Memory ares	Description
Decimal	Hex		
0 ... 511	0x0000 … 0x01FF	Physical Output Area	The first 512 outputs
512 … 1023	0x0200 … 0x03FF	Physical Output Area	The first 512 outputs (special)

→ The offsets for writing coil access:
Outputs: 0-511
Outputs with special functions: 512-1023

Register access, reading: :

Modbus Address		Memory area	Description
Decimal	Hex		
0 ... 255	0x0000 … 0x00FF	Physical Input Area	Inputs as bytes
256 … 511	0x0100 … 0x01FF	Something else…	
512 … 767	0x0200 ... 0x02FF	Physical Output Area	Outputs as bytes
768 …..	…..	…..	…..

→ The offsets for reading register access:
Inputs: 0-255
Outputs: 512-767

Register access writing:

Modbus Address		Memory area	Description
Decimal	Hex		
0 ... 255	0x0000 … 0x00FF	Physical Output Area	Outputs as bytes
256 … 511	0x0100 … 0x01FF	Something else…	
512 … 767	0x0200 ... 0x02FF	Physical Output Area	Outputs as bytes (special)
768 …..	…..	…..	…..

→ The offsets for writing register access:
Outputs: 0-255
Outputs with special functions: 512-767

Creation of the alias file:

All important information's are now available and we can create the alias file for this device.
The alias file is a simple text file (ending ".txt").
The alias file has to be formatted correctly. The entries are separated by "tab stops". To reduce the possibility of errors the alias file should be created and changed using the Excel template (alias.xls). After editing this file has to be "saved as" as "Text file, tab stop separated".
If it is not possible to use Excel the template alias.txt should be opened and edited. The best way is to copy/paste and then modify existing lines.

Note:

The Modbus OPC server uses offsets starting from 1. Many of the device documentations (like the one we used) start counting at 0. In this case you will have to add 1 to the calculated offset.

One line in the alias file consists of the alias name and the Modbus address. The address always has to be 6 digits long:

The first digit is the access mode (Coil/Register/Input/Output ➔ 0,1,3,4)
The next 5 digits are the offset.

Sample 1: writing output bits:
Writing, Coil: mode = 0
Output offset from documentation: 512 → + 1 (as documentation starts at 0) = 513
➔ The correct Modbus address is 000513 for Bit 1, 000514 for Bit 2 and so on...

Sample 2, reading input register:
Reading, register = mode = 3
Input offset from documentation: 0 → + 1 (as documentation starts at 0) = 1
➔ The correct Modbus address is 300001 for register 1, 300002 for register 2 and so on…

A configuration may now look like this:

Alias name	Modbus address	Comment
Input_Register_1	300001	First input byte -> inputs 1-8
Output_Register_1	400513	First output byte -> outputs 1-8
Input_Coil_1	100001	First input bit -> input 1
Input_Coil_2	100002	Second input bit-> input 2
Output_Coil_1	000513	First output bit -> output 1
Output_Coil_2	000514	Second output bit -> output 2

→ The Alias file now looks like this:

```
# AliasFile
# TAB<AliasName>TAB<ItemSyntax>TAB<Comments>...
        Input_Register_1        300001 First Input Byte -> Inputs 1-8
        Output_Register_1       400513 First Output Byte -> Outputs 1-8
        Input_Coil_1    100001 Fist Input Bit -> Input 1
        Input_Coil_2    100002 Second Input Bit -> Input 2
        Output_Coil_1   000513 First Output Bit -> Output 1
        Output_Coil_2   000514 First Output Bit -> Output 2
```

Note:

It may happen that the OPC-Client doesn't recognize the data types of register values correctly. In this case it is possible to define the data type of a register in the alias file. To do so, add a ":" behind the Modbus address, then define the data type. For example to read a real value:

```
Input_Register_1        300001:REAL   Fist Input
```

Please refer to the online help for more information:
➔ Operation ➔ OPC-Namespace ➔ Syntax items