Cognex In-Sight & SIMATIC S7 300 PLC

Modbus Communication Manual

S7-Client & IS-Server

V1.2

The information contained in this document has been developed solely for the purpose of providing general guidance to Cognex customers who need to configure communications between an In-Sight® sensor and a SIMATIC S7-300 PLC via Modbus protocol and data contained in this document serves informational purposes only.

Information in this document does not represent a commitment on the part of Cognex and in especially is not intending to be binding upon Cognex to any particular course of business. Cognex assumes no responsibility for errors or omissions in this document. Cognex does not make any express or implied representation or warranty as to the accuracy or completeness of the information for a particular purpose. Cognex shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of this document.

I. System properties

This section shows the software, firmware versions of the equipments and the hardware elements that were used for creating this document.

- Operating Systems
  - Windows XP Service Pack 3

- In-Sight Environment
  - In-Sight Explorer version 4.5.0 (build 474)
  - Camera type: In-Sight Micro 1400-10
  - Camera Firmware version: 4.5.0 (build 233

- Siemens S7 Environment
  - STEP 7 version 5.4 + Service Pack 5
  - CPU type: 315-2 PN/DP
  - S7-OpenModbus/TCP license for PN-CPU V2.X
  - Modbus TCP Wizard V 1.1.1.1
    (that can be downloaded from http://support.automation.siemens.com/WW/view/en/31535566)

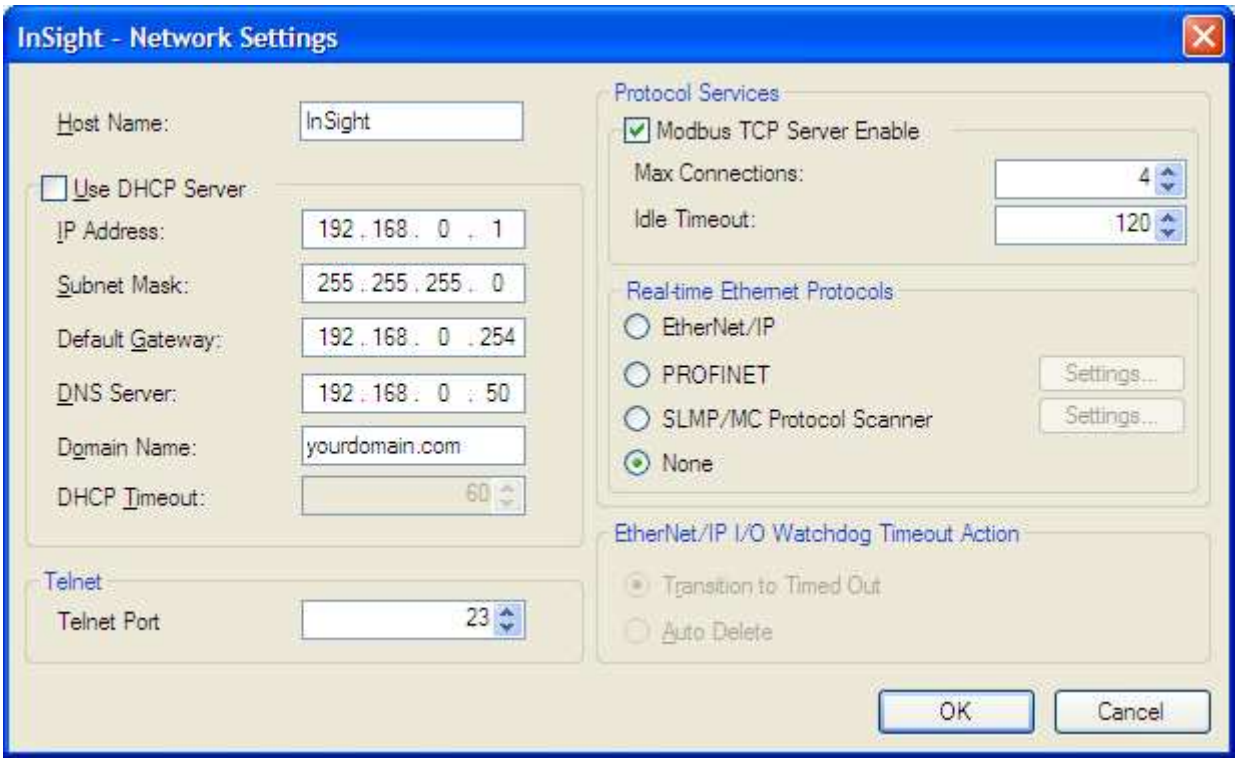
II. Enabling Modbus Communication on an In-Sight Vision System

Before Modbus a communication can be established with an In-Sight vision system, the vision system must be configured to enable Modbus, using the Network Settings of the In-Sight vision system.

1. Open In-Sight Explorer and connect to an In-Sight vision system.

2. From the Sensor menu, open the Network Settings dialog.

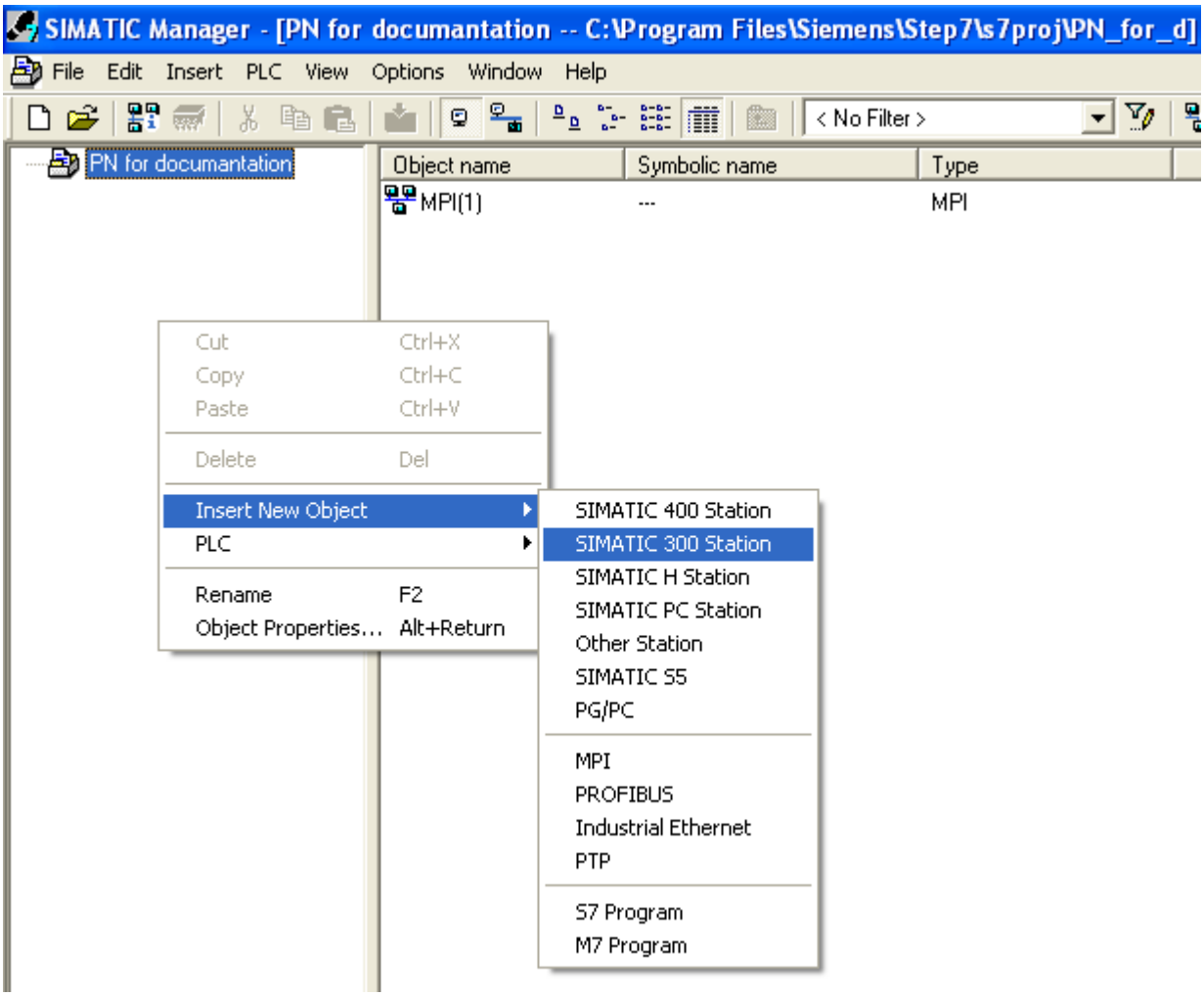
3. In the Protocol Services section of the dialog, select Modbus protocol and press OK.



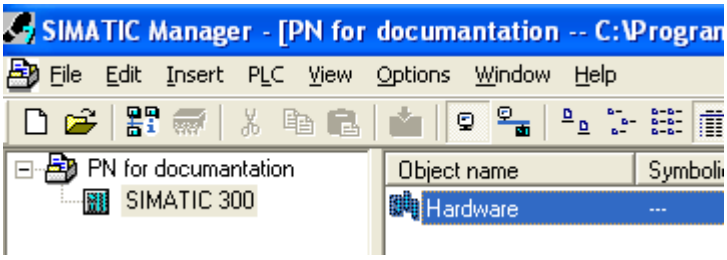
4. Restart the In-Sight vision system, and the Modbus service will be enabled upon completion of the power cycle.

III. Creating the HW configuration and parameter the connection in the PLC

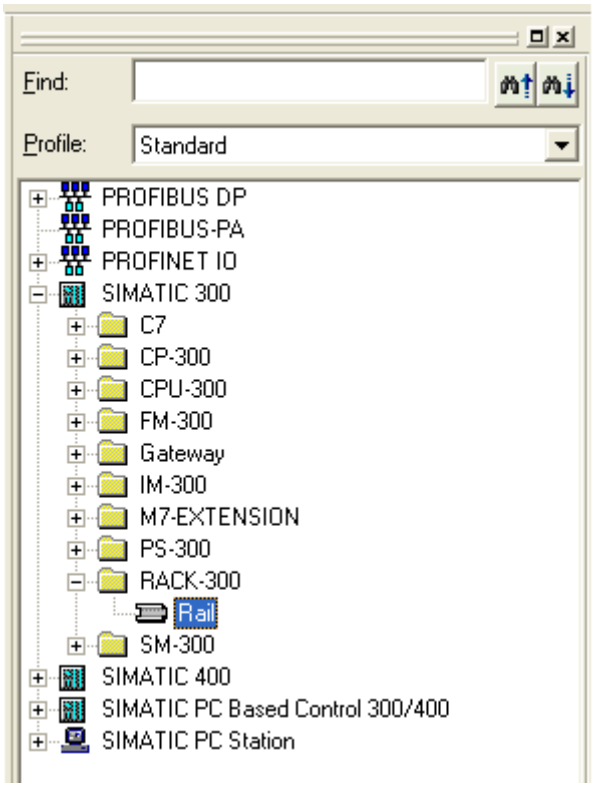
1. Open a new project in the SIMATIC Manager and insert a new station for your CPU.



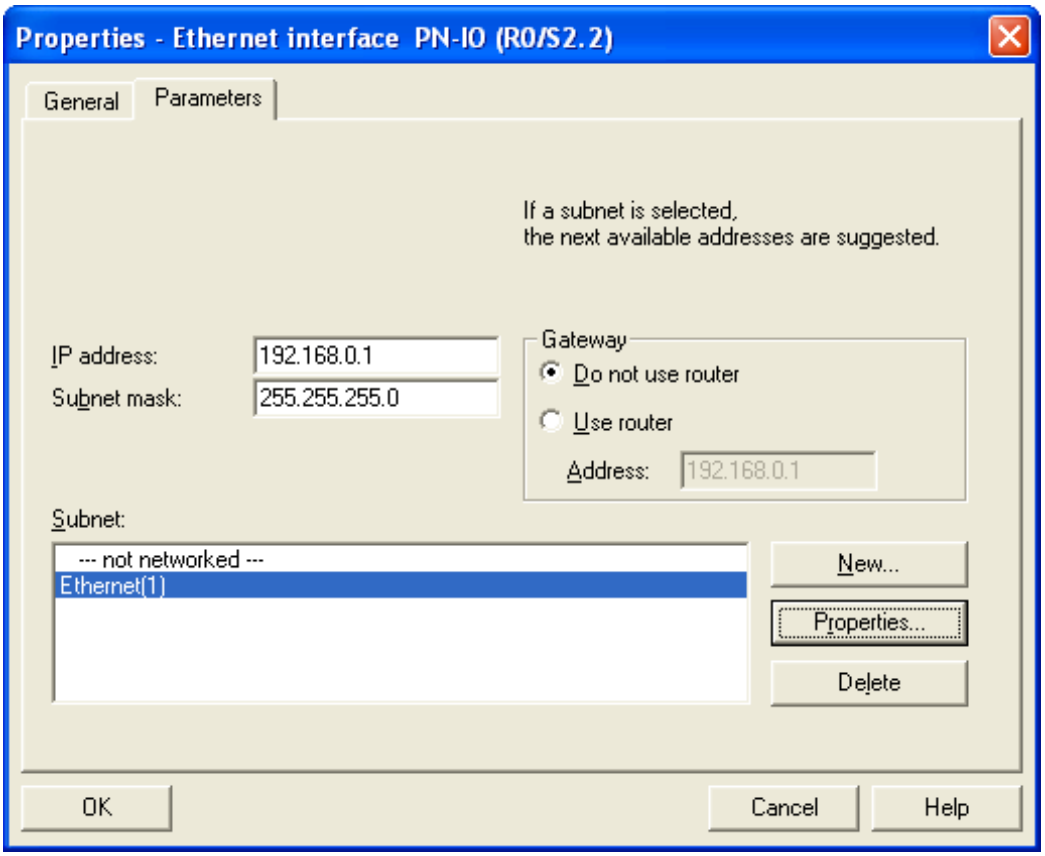
2. Select the inserted SIMATIC station and double click to the Hardware object on the right side to configure the hardware settings.



3. Add a new rail to your hardware configuration as shown on the image below.



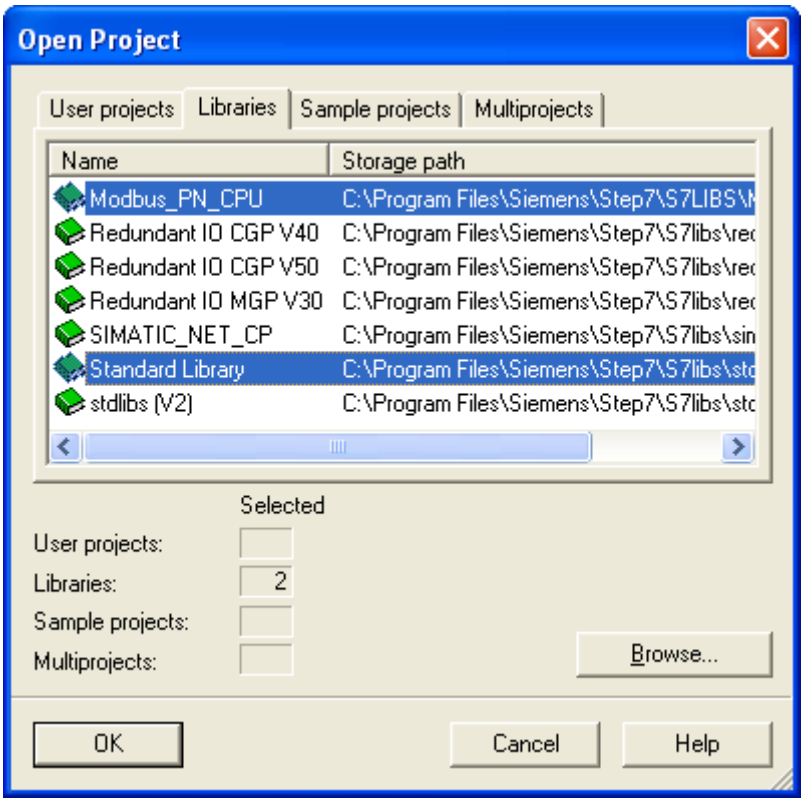
4. Drag and drop the correct CPU type to the rail and on the Parameters tab configure the IP settings then click on the New button.



5. After you have finished configuring your Hardware Configuration Save and Compile your config, then download it to the PLC. Now you can close the Hardware Configuration window.

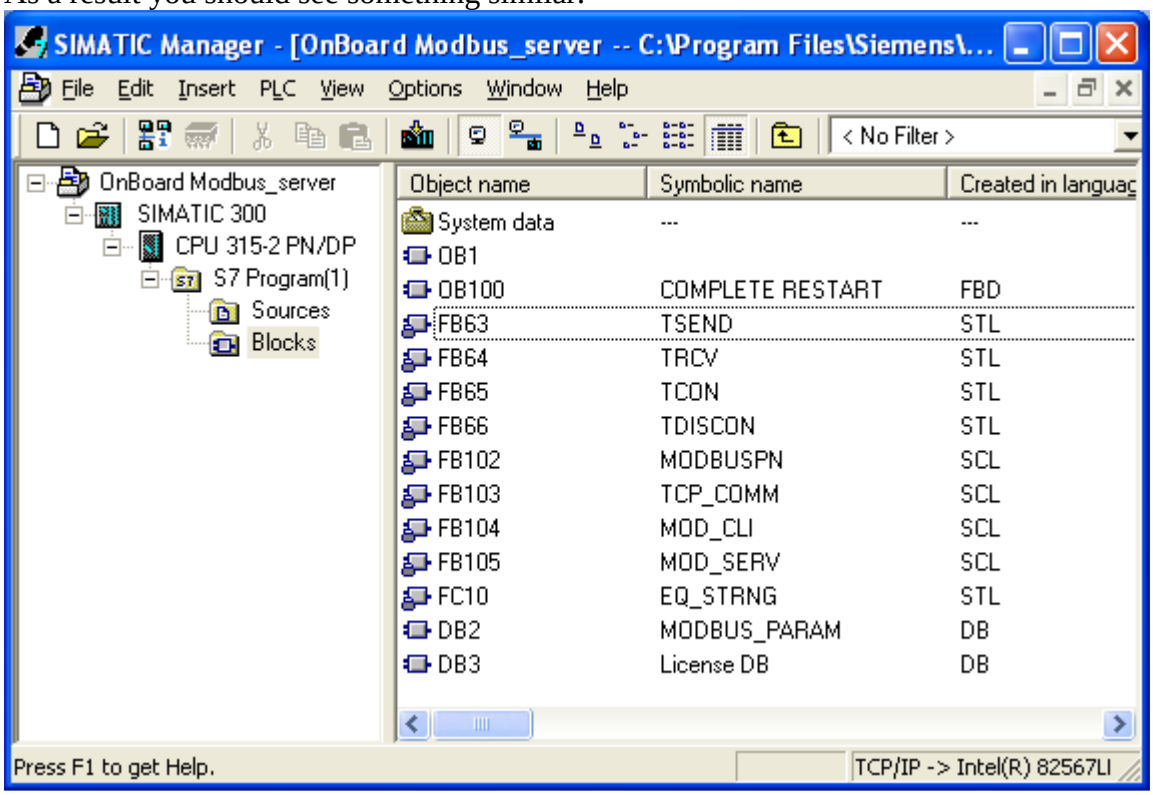
6. In SIMATIC Manager insert the following elements.
   - OB1 – This element will handle the cyclic data exchange via the Modbus system.
   - OB100 – This organization block will do the initialization of the Modbus communication.
   - FB63-FB66 – These blocks handles the TCP communication. These FBs can be found in Standard Library/Communication Blocks. (FB63-TSEND; FB64-TRCV; FB65-TCON; FB66-TDISCON) For more data on the FBs please visit the SIMATIC help.
   - FB102-FB105 – FB102 will be called during the communication. FB103-FB105 is called by FB102. These FBs can be found in Modbus_PN_CPU/PN CPU. (FB102-MODBUSPN; FB103-TCP_COMM; FB104-MOD_CLI; FB105-MOD_SERV)
   - DB2, DB3 – Parameter data blocks. These DBs can be found in Modbus_PN_CPU/PN CPU. (DB2-MODBUS_PARAM; DB3-License DB)
   - FC10 – This FB will be called as well by the FB102. This FC can be found in Standard Library/IEC Function Blocks. (FC10-EQ_STRING)

   To insert the FB63-FB66 do the following. In SIMATIC Manager open the menu File/Open, go to the Libraries tab and double click Standard Libraries. Go to Communication Blocks as listed above and copy the FBs. Change to your project view and insert the copied elements. To insert all other elements please use the same technique.

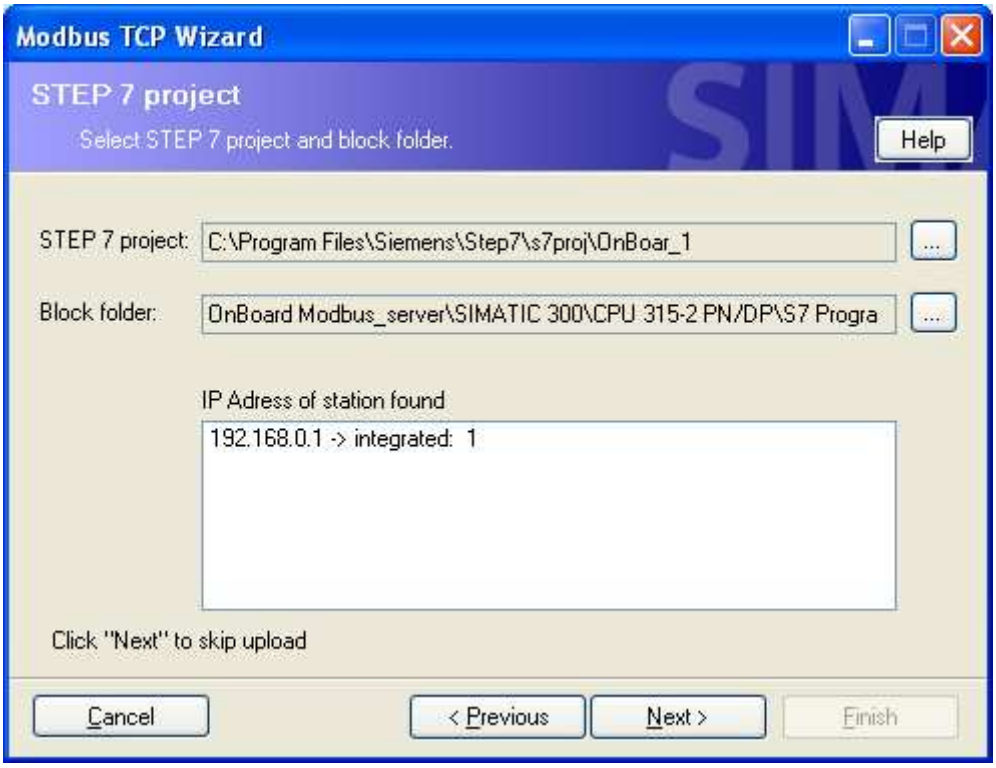


Note: To be able to insert the FB102-FB105 function blocks you have to install the S7-OpenModbus/TCP license for PN-CPU V2.X package that can be bought from Siemens.

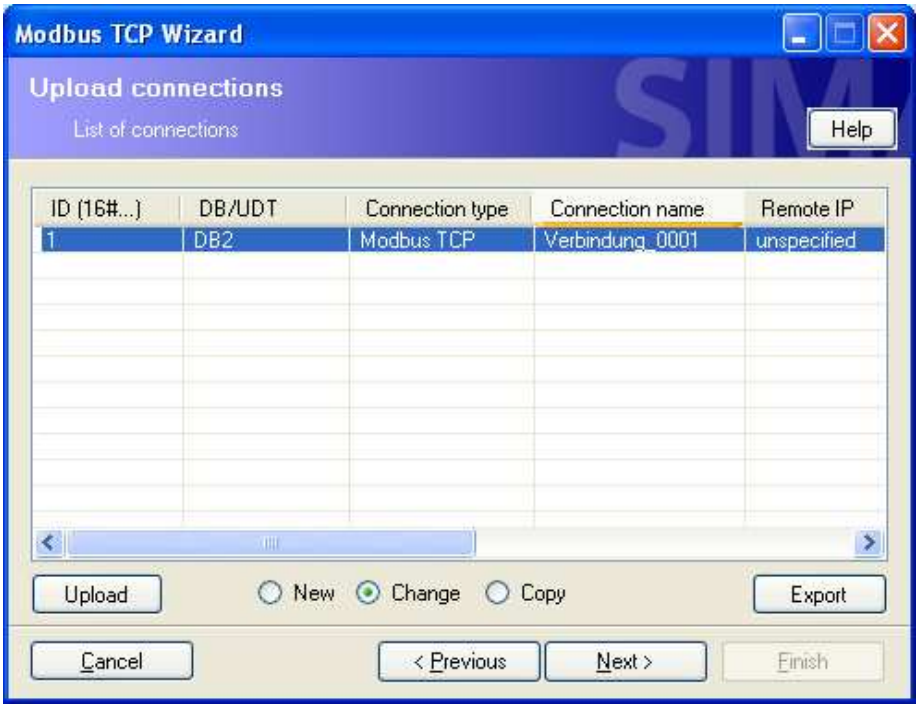
As a result you should see something similar.



7. For configuring the Modbus communication we will use Modbus TCP Wizard that is supplied by Siemens. In chapter I you can find the download link of the wizard. After you have installed it, you can open from Start/SIMATIC/Modbus TCP Wizard. On the second page browse for your project and the block folder where you've inserted the above listed elements, than click on Next.

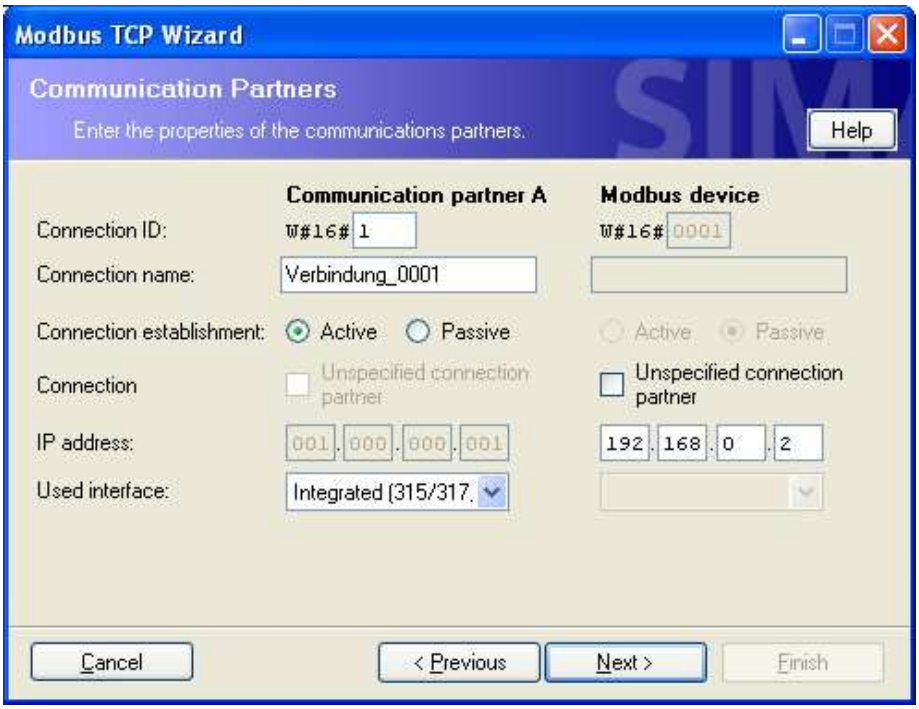


8. Select the Change radio button on the bottom of the window and click on the Upload button. Click on Ok in the new window and select the connection listed, than click on Next.



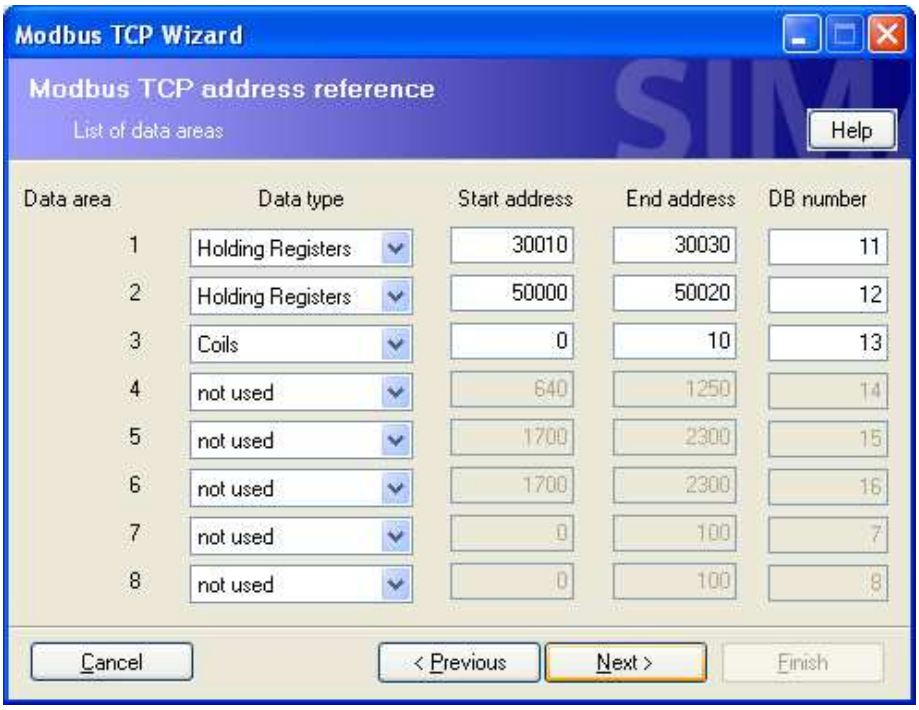
9. Make sure that the checkbox called SIMATIC S7 acts as a Server is unchecked, connect at startup and single write should be checked. Click on Next.

10. Communication Partner A means the CPU and the Modbus Device is the camera. Select the correct Used Interface and make sure that Active Connection Establishment is selected. Click on Next.



11. In this view you can set the port which will be used for the communication. Set port 502 as this port is used in the In-Sight System, when the camera acts as a Modbus server. Click on Next.

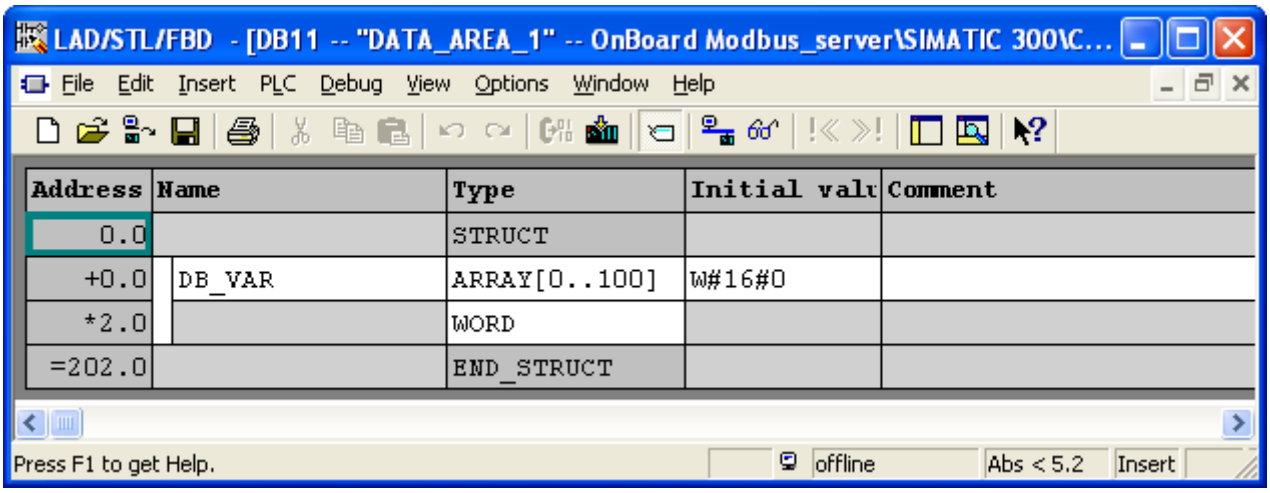
12. Here you can set the data exchange. You can only read complete blocks. Since DB2-DB3 is already used, so use DB11-DB18 in order to prevent conflicts. The PLC will read from Holding Register 30010 – 30137 address and will write to Holding Registers 50000 – 50127. As this sample only shows a short sample, a 20 byte block will be enough now. To be able to trigger the camera and fire the soft events in the camera enable a Coil range from address 0 to 10. There are no more settings to do in the wizard so click on Next until your settings are written into your project and than click on Finish.



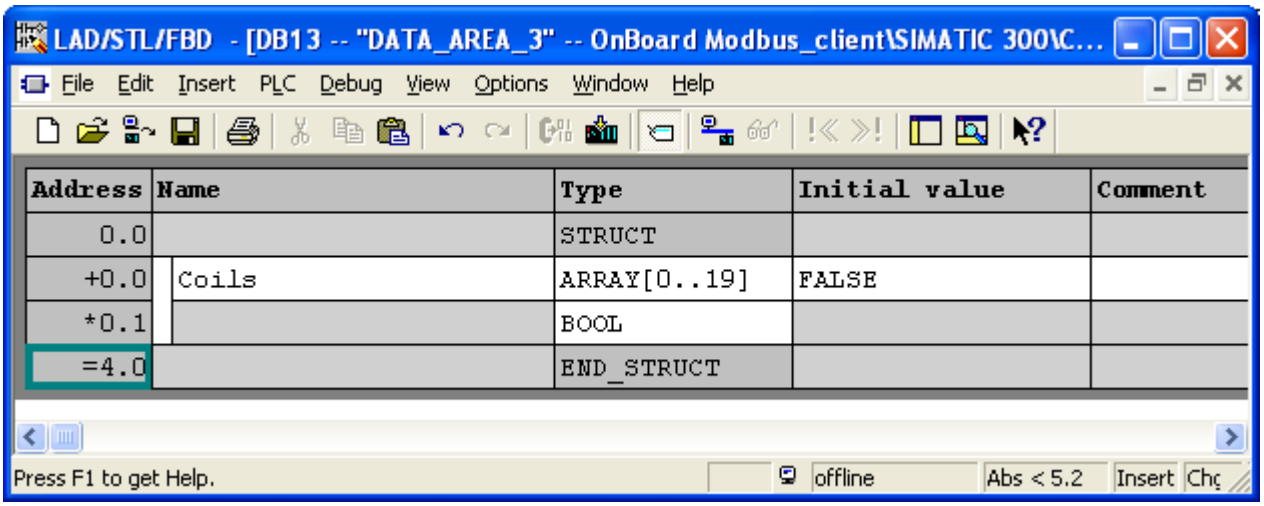
13. In your project please insert a new DB1 and name it CONTROL_DAT. The DB should look like the one below. This DB has been used in a sample created by Siemens showing the communication. This DB can also be copied from the sample project supplied with this document.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	ID	WORD	W#16#0
+2.0	DB_PARAM	WORD	W#16#0
+4.0	RECV_TIME	TIME	T#0MS
+8.0	CONN_TIME	TIME	T#0MS
+12.0	KEEP_ALIVE	BOOL	FALSE
+12.1	ENQ_ENR	BOOL	FALSE
+12.2	DISCONNECT	BOOL	FALSE
+14.0	RESERVED	STRING[17]	''
+33.0	LICENSED	BOOL	FALSE
+33.1	BUSY	BOOL	FALSE
+33.2	CONN_ESTABLISHED	BOOL	FALSE
+33.3	DONE_NDR	BOOL	FALSE
+33.4	ERROR	BOOL	FALSE
+34.0	STATUS_MODBUS	WORD	W#16#0
+36.0	STATUS_CONN	WORD	W#16#0
+38.0	STATUS_FUNC	STRING[8]	''
+48.0	IDENT_CODE	STRING[18]	''
+68.0	UNIT	BYTE	B#16#0
+69.0	DATA_TYPE	BYTE	B#16#0
+70.0	START_ADDRESS	WORD	W#16#0
+72.0	LENGTH	WORD	W#16#0
+74.0	TI	WORD	W#16#0
+76.0	WRITE_READ	BOOL	FALSE
+76.1	Save_DONE_NDR	BOOL	FALSE
+77.0	Save_UNIT	BYTE	B#16#0
+78.0	Save_DATA_TYPE	BYTE	B#16#0
+80.0	Save_START_ADDRESS	WORD	W#16#0
+82.0	Save_LENGTH	WORD	W#16#0
+84.0	Save_TI	WORD	W#16#0
+86.0	Save_WRITE_READ	BOOL	FALSE
+88.0	Save_STATUS_MODBUS	WORD	W#16#0
+90.0	Save_STATUS_CONN	WORD	W#16#0
+92.0	Save_STATUS_FUNC	STRING[8]	''
+102.0	Count_Done	WORD	W#16#0
+104.0	Count_Error	WORD	W#16#0
=106.0		END_STRUCT	

14. Insert 2 new Data Blocks into your project and call it DB11 and DB12. These DBs will contain the data that is sent/read from the In-Sight system. The name has to be the same like specified in the wizard. In our case we have given the number 11 and 12 for the DBs. The DB size should be at least 20 bytes as specified before as well, but can be more. Observe point 12 above.



15. Insert one more Data Block into your project and call it DB12. This DB will contain the Coils information which can be used for triggering the camera or firing Soft Events.



16. Open your OB100 and program the following code.

```
// setting the communication ID
      L     1
      T     "CONTROL_DAT".ID //DB1.DBW0

// call fb102, db102; click on yes to create IDB
      CALL  "MODBUSPN" , DB102
       ID              :="CONTROL_DAT".ID //DB1.DBW0
       DB_PARAM        :="MODBUS_PARAM" //DB2
       RECV_TIME       :=
       CONN_TIME       :=
       KEEP_ALIVE      :=
       ENQ_ENR         :=
       DISCONNECT      :=
       REG_KEY         :=
       LICENSED        :=
       BUSY            :="CONTROL_DAT".BUSY //DB1.DBX33.1
       CONN_ESTABLISHED:="CONTROL_DAT".CONN_ESTABLISHED //DB1.DBX33.2
       DONE_NDR        :="CONTROL_DAT".DONE_NDR //DB1.DBX33.3
       ERROR           :="CONTROL_DAT".ERROR //DB1.DBX33.4
       STATUS_MODBUS   :="CONTROL_DAT".STATUS_MODBUS //DB1.DBW34
       STATUS_CONN     :="CONTROL_DAT".STATUS_CONN //DB1.DBW36
       STATUS_FUNC     :="CONTROL_DAT".STATUS_FUNC
       IDENT_CODE      :=
       UNIT            :=
       DATA_TYPE       :=
       START_ADDRESS   :=
       LENGTH          :=
       TI              :=
       WRITE_READ      :=

L     500
T     "CONTROL_DAT".RECV_TIME //DB1.DBD4

L     T#5S
T     "CONTROL_DAT".CONN_TIME //DB1.DBD8

AN    "CONTROL_DAT".ERROR // DB1.DBX33.4
BEC

L     "CONTROL_DAT".STATUS_CONN // DB1.DBW36
T     "CONTROL_DAT".Save_STATUS_CONN // DB1.DBW90

L     "CONTROL_DAT".STATUS_MODBUS // DB1.DBW34
T     "CONTROL_DAT".Save_STATUS_MODBUS // DB1.DBW88
```

```
// STATUS_FUNC => Save_STATUS_FUNC
L     DB1.DBD   38
T     DB1.DBD   92
L     DB1.DBD   42
T     DB1.DBD   96
L     DB1.DBW   46
T     DB1.DBW  100
```

17. Open your OB1 and program the following code.

```
CALL  "MODBUSPN" , DB102
       ID               :=
       DB_PARAM         :=
       RECV_TIME        :="CONTROL_DAT".RECV_TIME //DB1.DBD4
       CONN_TIME        :="CONTROL_DAT".CONN_TIME //DB1.DBD8
       KEEP_ALIVE       :=
       ENQ_ENR          :="CONTROL_DAT".ENQ_ENR //DB1.DBX12.1
       DISCONNECT       :="CONTROL_DAT".DISCONNECT //DB1.DBX12.2
       REG_KEY          :="License DB".REG_KEY
       LICENSED         :="CONTROL_DAT".LICENSED //DB1.DBX33.0
       BUSY             :="CONTROL_DAT".BUSY //DB1.DBX33.1
       CONN_ESTABLISHED:="CONTROL_DAT".CONN_ESTABLISHED //DB1.DBX33.2
       DONE_NDR         :="CONTROL_DAT".DONE_NDR //DB1.DBX33.3
       ERROR            :="CONTROL_DAT".ERROR //DB1.DBX33.4
       STATUS_MODBUS    :="CONTROL_DAT".STATUS_MODBUS //DB1.DBW34
       STATUS_CONN      :="CONTROL_DAT".STATUS_CONN //DB1.DBW36
       STATUS_FUNC      :="CONTROL_DAT".STATUS_FUNC
       IDENT_CODE       :="CONTROL_DAT".IDENT_CODE
       UNIT             :="CONTROL_DAT".UNIT //DB1.DBB68
       DATA_TYPE        :="CONTROL_DAT".DATA_TYPE //DB1.DBB69
       START_ADDRESS    :="CONTROL_DAT".START_ADDRESS //DB1.DBW70
       LENGTH           :="CONTROL_DAT".LENGTH //DB1.DBW72
       TI               :="CONTROL_DAT".TI //DB1.DBW74
       WRITE_READ       :="CONTROL_DAT".WRITE_READ //DB1.DBX76.0


//reset trigger after job has started
A     "CONTROL_DAT".ENQ_ENR //DB1.DBX12.1
R     "CONTROL_DAT".ENQ_ENR //DB1.DBX12.1

//for cyclic data exchange remove the comment keys
//O     "CONTROL_DAT".DONE_NDR //DB1.DBX33.3
//O     "CONTROL_DAT".ERROR //DB1.DBX33.4
//S     "CONTROL_DAT".ENQ_ENR //DB1.DBX12.1
```

```
A     "CONTROL_DAT".DONE_NDR //DB1.DBX33.3
JCN   err

//job finished without error
L     "CONTROL_DAT".Count_Done //DB1.DBW102
INC   1
T     "CONTROL_DAT".Count_Done //DB1.DBW102

//TI should be incremented with each job
L     "CONTROL_DAT".TI //DB1.DBW74
INC   1
T     "CONTROL_DAT".TI //DB1.DBW74

BEU

err:  A     "CONTROL_DAT".ERROR //DB1.DBX33.4
JCN   end

//job finished with error
L     "CONTROL_DAT".Count_Error //DB1.DBW104
INC   1
T     "CONTROL_DAT".Count_Error //DB1.DBW104

L     "CONTROL_DAT".STATUS_CONN //DB1.DBW36
T     "CONTROL_DAT".Save_STATUS_CONN //DB1.DBW90

L     "CONTROL_DAT".STATUS_MODBUS //DB1.DBW34
T     "CONTROL_DAT".Save_STATUS_MODBUS //DB1.DBW88

// STATUS_FUNC => Save_STATUS_FUNC
L     DB1.DBD   38
T     DB1.DBD   92
L     DB1.DBD   42
T     DB1.DBD   96
L     DB1.DBW   46
T     DB1.DBW  100

end:  NOP   0
```

18. If you have not licensed the S7-OpenModbus/TCP package so far, you will get System Fails in every 4 seconds. For test reasons you can prevent this by adding OB121. Please note that this organization block will prevent all system fails in your program so this shouldn’t be used in live environments!

19. Save your project and download it to the PLC.

IV. Setting up the communication in the In-Sight Vision System

Before starting make sure that you have done all the settings listed in chapter I & II.

For this example, create a new job within In-Sight Explorer and then perform the following steps to configure the communication.

Note: If you are missing the Palette, than open View menu and click on the point Palette.

Note: If you are not in Spreadsheet mode (meaning that you don't see and excel like interface), open the View menu and click on the Spreadsheet menu point.

1. Open the AcquireImage function in call A0 and change the camera trigger type to Continuous. This setting is only needed to make the camera update the Modbus communication buffer.

2. Right click on cell D5 and select insert function. In the Insert Function Window select the group Graphics/Controls and highlight the function called EditInt, then press Ok. Set the maximum value to 255. This ensures that the number entered in the cell won't overflow 1 byte size.

3. Insert a new EditInt function into D6, with the same technique detailed above. Set the maximum value to 65535. This ensures that the number entered in the cell won't overflow 2 byte size.

4. Insert a new EditInt function into D7. Set the maximum to 9999999. This is the maximum of the function and the entered number won't overflow 4 byte size.

5. Insert a new EditFloat function into D8. You can find it in the same group as EditInt.

6. Insert a new EditString into D9 and set the maximum string length to 4 chars. Fill D5-D9 cells with values.

7. Insert a FormatOutputBuffer to A4 from the group Input/Output. In the function insert a 8 bit integer referring to cell D5, a 16 bit integer referring to cell D6, a 32 bit integer referring to cell D7, a 32 bit float referring to cell D8 and a 4 byte size string referring to cell D9. Don't change the order.

8. Insert a WriteModbusBuffer to cell B4 from the group Input/Output and in the Buffer field you need to reference the cell A4. The buffer that was just inserted. No more changes needed for this function. The functions inserted in point 2-8 are responsible for the outgoing data. The control system will read these data.

9. Insert a FormatInputBuffer to cell A12 from the group Input/Output and add the following data types in the same order: 8 bit integer, 16 bit integer, 32 bit integer, 32 bit float and a 4 byte string.

10. Insert a ReadModbusBuffer to cell B12 from the group Input/Output and in the Buffer field you need to reference the cell A12. The buffer that was just inserted. No more changes needed for this function.

11. Click on cell D13 and type "GetBufferData(" then click on cell B12 press a coma and type 0, than press enter. You have inserted a GetBufferData function and set up to read the data with index 0 from the camera input Modbus area. In this case index 0 means the 8 bit integer that was created in point 9.

12. Click on cell D14 and insert the GetBufferData function refer once again cell B12, but now the index is 1. Do the same for D15-D17 cells and always increment the index with 1. The functions inserted in point 9-12 are responsible for the incoming data. The control system will write these data.

13. Switch the camera to Online.

14. In your SIMATIC project create the following variable table. If you want you can copy this VAT from the sample project supplied with this document.

	Address	Symbol	Display format	Status value	Modify value
1	DB1.DBX 12.1	"CONTROL_DAT".ENQ_ENR	BOOL		true
2	DB1.DBX 12.2	"CONTROL_DAT".DISCONNECT	BOOL		
3	DB1.DBX 33.0	"CONTROL_DAT".LICENSED	BOOL		
4	DB1.DBX 33.1	"CONTROL_DAT".BUSY	BOOL		
5	DB1.DBX 33.2	"CONTROL_DAT".CONN_ESTABLISHE	BOOL		
6	DB1.DBX 33.3	"CONTROL_DAT".DONE_NDR	BOOL		
7	DB1.DBX 33.4	"CONTROL_DAT".ERROR	BOOL		
8	DB1.DBW 34	"CONTROL_DAT".STATUS_MODBUS	HEX		
9	DB1.DBW 36	"CONTROL_DAT".STATUS_CONN	HEX		
10	DB1.DBD 40		CHARACTER		
11	DB1.DBD 44		CHARACTER		
12					
13	DB1.DBW 88	"CONTROL_DAT".Save_STATUS_MO	HEX		
14	DB1.DBW 90	"CONTROL_DAT".Save_STATUS_CON	HEX		
15	DB1.DBD 94		CHARACTER		
16	DB1.DBD 98		CHARACTER		
17	DB1.DBW 102	"CONTROL_DAT".Count_Done	DEC		
18	DB1.DBW 104	"CONTROL_DAT".Count_Error	DEC		
19					
20	DB1.DBB 68	"CONTROL_DAT".UNIT	DEC		0
21	DB1.DBB 69	"CONTROL_DAT".DATA_TYPE	HEX		B#16#03
22	DB1.DBW 70	"CONTROL_DAT".START_ADDRESS	HEX		W#16#C350
23	DB1.DBW 72	"CONTROL_DAT".LENGTH	DEC		15
24	DB1.DBW 74	"CONTROL_DAT".TI	DEC		2
25	DB1.DBX 76.0	"CONTROL_DAT".WRITE_READ	BOOL		true
26					
27	DB11.DBB 0		DEC		
28	DB11.DBW 1		DEC		
29	DB11.DBD 3		DEC		
30	DB11.DBD 7		FLOATING_POINT		
31	DB11.DBD 11		CHARACTER		
32					
33	DB12.DBB 0		DEC		125
34	DB12.DBW 1		DEC		25414
35	DB12.DBD 3		DEC		L#69514
36	DB12.DBD 7		FLOATING_POINT		54.321
37	DB12.DBD 11		CHARACTER		'char'
38					

15. Switch your VAT to online mode. To read data from the IS camera write the following fields.

    - Write 0 to “CONTROL-DAT”.UNIT
    - Write B#16#03 to “CONTROL-DAT”.DATA_TYPE (03 – Holding Register)
    - Write W#16#753A to “CONTROL-DAT”.START_ADDRESS (address 30010)
    - Write 15 to “CONTROL-DAT”.LENGTH
    - Write FALSE to “CONTROL-DAT”.WRITE_READ (Write=true; Read=false)

    Note: When using the WriteModbusBuffer function, the remote client that is reading from the In-Sight vision system must use Registers 30010 – 30137.

16. To trigger the data reading write a TRUE value to “CONTROL-DAT”.ENQ_ENR. Every time you would like to read/write data the ENQ_ENR bit has to be set to TRUE. After the transmission it is automatically set back to FALSE. If all the settings are correct the read data should be displayed in DB11.DBB0-DB11.DBB14 (in the sample image above it is line 27-31). In the variable table you should see the values entered in the camera job into cells D5-D9 in point 6 above.

17. To write data to the camera fill DB12.DBB0-DB12.DBB14 with values (in the sample image above it is line 33-37). Than change the following parameters:

    - Write W#16#C350 to “CONTROL-DAT”.START_ADDRESS (address 50000)
    - Write TRUE to “CONTROL-DAT”.WRITE_READ (Write=true; Read=false)

    Note: When using the ReadModbusBuffer function, the remote client that is writing to the In-Sight vision system must use Registers 50000 – 50127.

18. Once again write a TRUE value to “CONTROL-DAT”.ENQ_ENR. The sent data should be displayed in the In-Sight spreadsheet in cell D13-D17.

Note: Further details for the above mentioned In-Sight functions can be found in the In-Sight Explorer Help, just do a keyword search with the function name.

Note: The In-Sight system uses single precision floating precision values in the background. The specification for single precision floating point values, that allows 24 bits of precision, reserving 1 bit for a sign, and the remaining 7 bits for the exponent. This means the current implementation of the 32-bit integer is not useful for any type of ID that is more than 24 bits in length. However the rounding errors can usually be tolerated for any type of vision parameter or non-id based vision result.

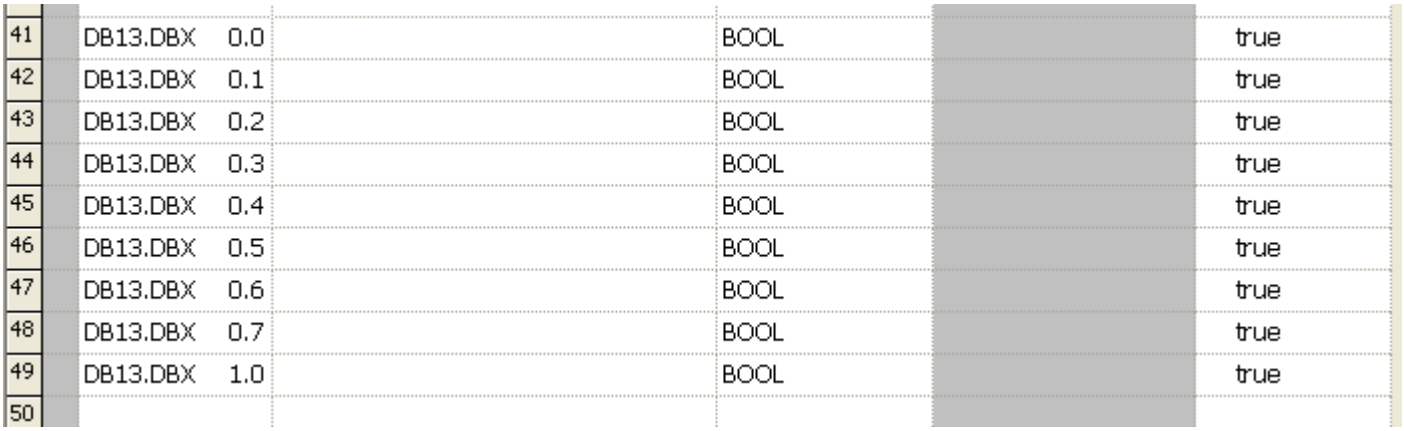
V. Triggering the In-Sight system from the PLC via Modbus

To be able to trigger the camera from the PLC some changes has to be made in the variable table created before.

To trigger the camera or fire soft events the control system has to send a Write Coil command (Modbus Function code 05) with the correct coil number. The coil numbers are the following. Further details can be found for the Modbus packages in the chapter called Understanding Modbus communication and packages.

Coil Number	Description
0	Specifies Soft Trigger 0
1	Specifies Soft Trigger 1
2	Specifies Soft Trigger 2
3	Specifies Soft Trigger 3
4	Specifies Soft Trigger 4
5	Specifies Soft Trigger 5
6	Specifies Soft Trigger 6
7	Specifies Soft Trigger 7
8	Acquire an image and update the spreadsheet. This option requires the **AcquireImage** (cell A0) functions **Trigger** argument to be set to **External** or **Manual**.

1. Insert a new range into the variable table. Here you can see the status of the bits. To fire an event or trigger the camera you can send either true or false values. Basically the event is triggered with the package receive and not with the sent value, so the event or the trigger will fire with false and true values as well.



41	DB13.DBX 0.0		BOOL		true
42	DB13.DBX 0.1		BOOL		true
43	DB13.DBX 0.2		BOOL		true
44	DB13.DBX 0.3		BOOL		true
45	DB13.DBX 0.4		BOOL		true
46	DB13.DBX 0.5		BOOL		true
47	DB13.DBX 0.6		BOOL		true
48	DB13.DBX 0.7		BOOL		true
49	DB13.DBX 1.0		BOOL		true
50					

2. Update the following values:

- Write the value B#16#01 to "CONTROL_DAT".DATA_TYPE
- Write the value W#16#0008 to "CONTROL_DAT".START_ADDRESS (Coil number 8 -> Trigger)
- Write the value 1 to "CONTROL_DAT".LENGTH (always use 1 for length, if you would like to write more coils, do it in several steps)
- Write TRUE to "CONTROL_DAT".WRITE_READ

3. To start the data transmission write a TRUE value to “CONTROL-DAT”.ENQ_ENR.

VI. Understanding Modbus communication and packages

For this Modbus communication we used 3 different communication functions (03 - Read Holding Registers, 05 - Write Coil and 16 - Write Multiple Registers). Below these functions package layouts are detailed to help debugging and show the correct package type for the communication.

03 – Read Holding Registers

This packet reads data from the Holding register of the remote Modbus Server (in this case the In-Sight camera). As the name shows it is possible read more than one registers at one time. The following layout shows what package has to be sent to the camera. The layout shows a 2 byte data read from an In-Sight system.

Byte	Value (Hex)	Description
0	0	Modbus Header
1	0	
2	0	
3	0	
4	0	
5	6	Bytes to follow
6	0	Unit ID
7	3	Modbus Function Code (03 for Read Multiple Registers)
8	75	Register address higher (75) and lower (3A) byte with Hexadecimal representation. (753A Hex = 30010 Dec). 30010 is the start address of the output register of the camera
9	3A	
10	0	Higher byte – Number of registers (including starting register)
11	1	Lower byte – Number of registers (including starting register)

If more than 2 byte has to be read from the camera the last 2 bytes will change, the rest should be untouched.

Note: 1 register is 2 byte.

05 – Read Coils

Initiates an event in the In-Sight spreadsheet. Writing a non-zero value to Coils 0 through 7 will trigger Soft Events 0 through 7 in the spreadsheet. Writing a non-zero value to Coil 8 will trigger an acquisition. Error code 02 (ILLEGAL DATA ADDRESS) is returned if the coil number is not in the valid range (0 to 8).

Byte	Value (Hex)	Description
0	0	Modbus Header
1	0	
2	0	
3	0	
4	0	
5	6	Bytes to follow
6	0	Unit ID
7	3	Modbus Function Code (05 for Write Coil)
8	0	Higher byte – Coil Number
9	8	Lower byte – Coil Number
10	FF	ON. A remote device output can either be ON (255) or OFF (0)
11	0	Always zero

16 – Write Multiple Registers

This packet writes data to the Holding register of the remote Modbus Server (in this case an In-Sight system). As the name shows it is possible write more than one registers at one time. The Write Multiple Registers packet layout is the following. This example shows how the layout look like if a 2 byte data is sent to the camera.

Byte	Value (Hex)	Description
0	0	Modbus Header
1	0	
2	0	
3	0	
4	0	
5	9	Bytes to follow
6	0	Unit ID
7	10	Modbus Function Code (16 for Write Multiple Registers)
8	C3	Register address higher (C3) and lower (50) byte with Hexadecimal representation. (C350 Hex = 50000 Dec). 50000 is the start address of the input register of the camera
9	50	
10	0	Higher byte – Number of registers (including starting register)
11	1	Lower byte – Number of registers (including starting register)
12	2	Data byte count
13	0	Higher byte – Data
14	8	Lower byte – Data
15	…	*Further data bytes (if needed)*

In case more data have to be written into the camera then the ‘Bytes to follow’, the ‘Number of Registers’ and the ‘Data byte count’ have to be updated according to the changes.

VII. Debug help

To debug the communication, you can use some tools that are available on the web. You can download some Modbus emulators form www.modbustools.com. You can get a Modbus server emulator called Modbus Salve and a client emulator called Modbus Poll. Both the server side and the client side can be simulated, which means that either the Camera or the PLC can be emulated. A smart feature of the emulators is that the communication can be monitored and the package contents are visible, which makes the debug process easier and faster.