PITreader Modbus-Connection with different PLC systems

PILZ
THE SPIRIT OF SAFETY

Product

Type:	Operating mode selection and access permission system
Name:	PITreader
Manufacturer:	Pilz GmbH & Co. KG, Safe Automation

Document

Release Number:	01
Release Date:	03 June 2020

Application Note – 1005380_EN_01

Document Revision History

Release	Date	Changes	Chapter
01	2020-06-03	Creation	all

Validity of Application Note

This present Application Note is valid until a new version of the document is published.
This and other Application Notes can be downloaded in the latest version and for free from www.pilz.com. For a simple search, use our content document (1002400) or the direct search function in the download area.

The Pilz newsletter is free of charge and keeps you up-to-date on all the latest issues and trends in safe automation.

Exclusion of Liability

We have taken great care in compiling our application note. It contains information about our company and our products. All statements are made in accordance with the current status of technology and to the best of our knowledge and belief.
While every effort has been made to ensure the information provided is accurate, we cannot accept liability for the accuracy and entirety of the information provided, except in the case of gross negligence. In particular, all information on applicable standards, safety-related classifications and time characteristics should be viewed as provisional. In particular it should be noted that statements do not have the legal quality of assurances or assured properties.
We are grateful for any feedback on the contents.

June 2020

Industrial Security

To secure plants, systems, machines and networks against cyberthreats it is necessary to implement (and continuously maintain) an overall Industrial Security concept that is state of the art.
Perform a risk assessment in accordance with VDI/VDE 2182 or IEC 62443-3-2 and plan the security measures with care. If necessary, seek advice from Pilz Customer Support.

Abbreviations

Abbreviation / term	Description	Source
AN	**A**pplication **N**ote	www.pilz.com > AN content (1002400)
PNOZ	Pilz E-STOP positive-guided (DE: **P**ilz **NO**T-AUS-**Z**wangsgeführt)	www.pilz.com > PNOZ
PSS	Programmable control system (DE: **P**rogrammierbares **S**teuerungs**s**ystem)	www.pilz.com > PSS
PSS u2	**PSSu**niversal, **2**nd generation	www.pilz.com > PSS u2
POU	**P**rogram **O**rganisation **U**nit	
NC	**N**ormally **C**losed	
NO	**N**ormally **O**pen	
FB	**Function Block**	
DB	**Data Block**	

Definition of Symbols

Information that is particularly important is identified as follows:

CAUTION!

This refers to a hazard that can lead to a less serious or minor injury plus material damage, and also provides information on preventive measures that can be taken.

NOTICE

This describes a situation in which the product or devices could be damaged and also provides information on preventive measures that can be taken.
It also highlights areas within the text that are of particular importance.

INFORMATION

This gives advice on applications and provides information on special features.

Contents

1 Useful documentation

Reading the documentation listed below is necessary for understanding this Application Note. The availability of the software used and its safe handling are also presupposed for the user.

1.1 Documentation from Pilz GmbH & Co. KG

No.	Description	Item No. /Download
1	Pilz international homepage, download section	www.pilz.com
2	Operation manual PITreader	1005200-EN-xx www.pilz.com > Download 1005200
3		
4		

1.2 Documentation from other sources of information

No.	Description	Item No. / Download
1	Siemens Manual SIMATIC STEP 7 Basic/Professional V15.1 and SIMATIC WinCC V15.1	109755202, 10/2018 support.industry.siemens.com > 109755202
2	Siemens Manual Programming Guideline for S7-1200/S7-1500	90885040, 28.03.2017 support.industry.siemens.com > 90885040
3	Siemens Homepage, SCE Training Curriculums (TIA)	w3.siemens.com > MCMS/SCE
4	Application example How do you program and parameterize Modbus/TCP communication between S7-1500 CPUs and S7-1200 CPUs?	102020340, 15.08.2019 support.industry.siemens.com > 102020340

2 Used hardware and software

2.1 Pilz products

No.	Descriptions	Order number	Version	Number
1	PITreader base unit	402255	-	1
2	PITreader key adapter h	402308	-	1
3	PITreader key ye g (Generic transponder key)	402260	-	1
4				

2.2 Third-party products

No.	Descriptions	Order number	Version	Number
1	Siemens Simatic CPU1518F-4 PN/DP	6ES7 513-1AL01-0AB0	2.6	1
2	Siemens Simatic CPU1215FC DC/DC/RLY	6ES7 215-1HF40-0XB0	4.2	1
3	Kunbus Revolution PI (RevPi)	4260498390007	1.2	1
4				

3 Application description

This application note describes how to establish a Modbus connection between the PITreader and a Simatic S7-1200 / 1500, Rockwell ControlLogix and Kunbus RevPi.
The basics for handling with the PLC and the tool are not part of this document.

Siemens S7-1200 / 1500: A example project from Siemens is the basis for a Modbus connection.
In the online help of the TIA portal you can find the description of the Modbus block.
There you can also find a description of the error codes.

INFORMATION

The PITreader is Modbus Server, so the PLC has to be configured as Modbus Client.

4 Hardware configuration

4.1 Used hardware

- Siemens S7-1200/1500
- Rockwell ControlLogix
- Kunbus RevPi
- Pilz PITreader

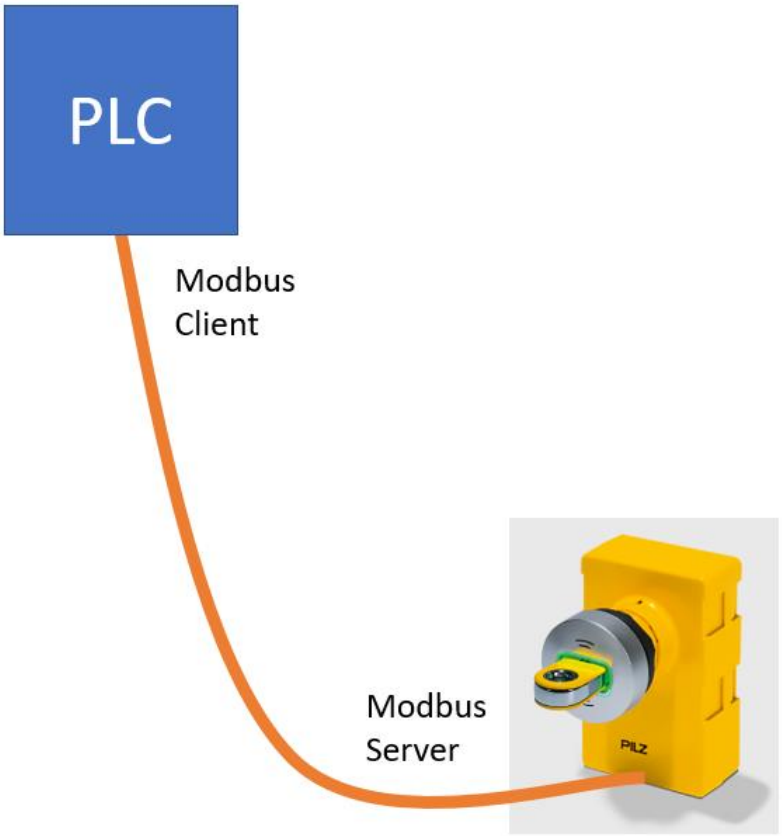

Figure 1: Hardware configuration

4.1.1 Pilz devices

- The used Pilz modules are listed in the table of the following link:
  - Chapter 2.1 [📖6]
- The relevant important documents are named and linked here:
  - Chapter 1.1 [📖5]

4.1.2 Third-party devices

- The used Third-party devices are listed in the table of the following link:
  - Chapter 2.1 [📖6]
- The relevant important documents are named and linked here:
  - Chapter 1.2 [📖5]

5 Siemens S7-1200

5.1 Modify the PITreader IP address

5.2 Example project

The Modbus example project can be downloaded from Siemens.
https://support.industry.siemens.com/cs/document/102020340/how-do-you-program-and-parameterize-modbus-tcp-communication-between-s7-1500-cpus-and-s7-1200-cpus-?dti=0&lc=en-DE

The example project contains two different devices. The S7-1200 and the S7-1500 hardware.
For the Application Note the example project with the suitable hardware was used.

INFORMATION

There are Modbus Client and Modbus Server blocks. Because the PITreader is a Modbus server, the Modbus client block must be used on the PLC side.

A few adjustments need to be made in the example project.
This is described on the following pages.

5.3 FB1 “ModbusClient”

The function block FB1 "ModbusClient" is called cyclically in OB1.

The FB1 "ModbusClient" calls the "MB_CLIENT" instruction internally to establish the Modbus/TCP connection and read the holding register from the Modbus TCP server.
The communication request to read the holding register is controlled via the "ModbusData".clientData.request tag at the "request" input.
In this example the Modbus TCP connection with connection number=2 is established to Port 502 of the Modbus TCP server. The Modbus TCP server has the IP address 192.168.0.12.
122 holding registers are read on the remote address 0. For this you set the input parameters "modbusMode", "modbusDataAddress" and "modbusDataLen" as follows:

- modbusMode = 103
- modebusDataAddress = 0
- modbusDataLen = 122

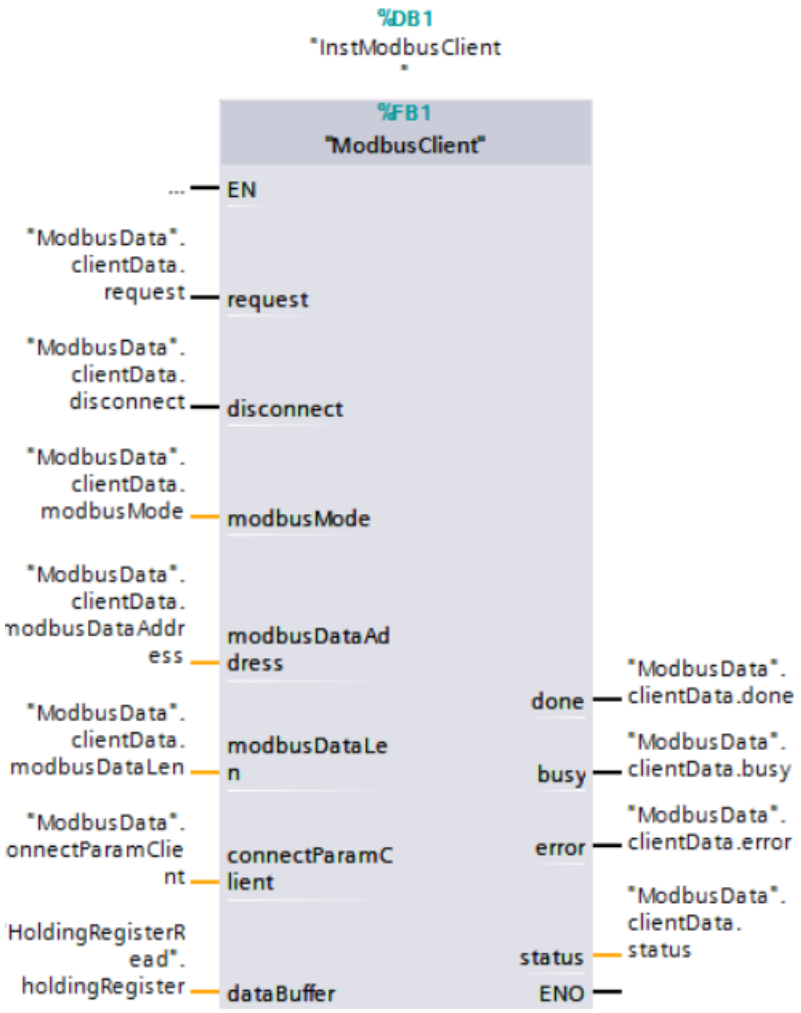

Figure 2: Simatic S7-1200 / FB1 – Modbus Client

5.3.1 Configuring DB100 “connection data”

5.3.1.1 Modbus parameter

In DB100 the IP address of the connection partner
- RemoteAddress
- modbusMode
- modbusDataAddress
- modbusDataLen

must be adjusted.

The Modbus registers of the PITreader are described in the operating manual and a short summary from 🕮 14.

Name	Data type	Start value
Static		
serverData	Struct	
clientData	Struct	
request	Bool	TRUE
enable	Bool	FALSE
disconnect	Bool	FALSE
done	Bool	FALSE
busy	Bool	TRUE
error	Bool	FALSE
status	Word	16#7006
statusSave	Word	16#80C4
modbusMode	USInt	103
modbusDataAddr...	UDInt	0
modbusDataLen	UInt	122
connectParamServer	TCON_IP_v4	
connectParamClient	TCON_IP_v4	
InterfaceId	HW_ANY	64
ID	CONN_OUC	16#0002
ConnectionType	Byte	16#0B
ActiveEstablished	Bool	TRUE
RemoteAddress	IP_V4	
ADDR	Array[1..4] of Byte	
ADDR[1]	Byte	16#C0
ADDR[2]	Byte	16#A8
ADDR[3]	Byte	16#00
ADDR[4]	Byte	16#0C
RemotePort	UInt	502
LocalPort	UInt	0

1

2

Figure 3: Simatic S7-1200 / DB100 – Connection Data

1

- modbusMode: In the following table [📖 13] you read the "modbusMode" according to the data you want to read out from the PITreader.
- modbusDataAddress: This is the start address of the requested Modbus Register.

INFORMATION

On the PITreader, the addressing for Modbus/TCP data areas starts at "1".
On the Simatic PLCs addressing start at "0".

- modbusDataLen: Look in on the PITreader user manual to find out which data you need and enter the required data length.

2

- RemoteAddress in HEX:
  - ADDR[1]: 16#C0 (192 dec)
  - ADDR[2]: 16#A8 (168 dec)
  - ADDR[3]: 16#00 (0 dec)
  - ADDR[4]: 16#0C (12 dec)
- RemotePort: 502

5.3.1.2 Simatic S7-1200 / modbusMode

MB_MODE, MB_DATA_ADDR and MB_DATA_LEN parameters

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Modbus function	Function and data type
0	1 to 9,999	1 to 2,000	01	Read 1 to 2,000 output bits on the remote address 0 to 9,998
0	10,001 to 19,999	1 to 2,000	02	Read 1 to 2,000 input bits on the remote address 0 to 9,998
0	• 40,001 to 49,999 • 400,001 to 465,535	1 to 125	03	• Read 1 to 125 holding registers on the remote address 0 to 9,998 • Read 1 to 125 holding registers on the remote address 0 to 65,534
0	30,001 to 39,999	1 to 125	04	Read 1 to 125 input words on the remote address 0 to 9,998
1	1 to 9,999	1	05	Write 1 output bit on the remote address 0 to 9,998
1	• 40,001 to 49,999 • 400,001 to 465,535	1	06	• Write 1 holding register on the remote address 0 to 9,998 • Write 1 holding register on the remote address 0 to 65,534
1	1 to 9,999	2 to 1,968	15	Write 2 to 1,968 output bits on the remote address 0 to 9,998
1	• 40,001 to 49,999 • 400,001 to 465,535	2 to 123	16	• Write 2 to 123 holding registers on the remote address 0 to 9,998 • Write 2 to 123 holding registers on the remote address 0 to 65,534
2	1 to 9,999	1 to 1,968	15	Write 1 to 1,968 output bits on the remote address 0 to 9,998
2	• 40,001 to 49,999 • 400,001 to 465,535	1 to 123	16	• Write 1 to 123 holding registers on the remote address 0 to 9,998 • Write 1 to 123 holding registers on the remote address 0 to 65,534
11	The MB_DATA_ADDR and MB_DATA_LEN parameters are not evaluated when this function is executed.		11	Read status word and event counter of the server: • The status word reflects the the processing status (0 – not processing, 0xFFFF – processing). • The event counter is incremented when the Modbus request was executed successfully. If an error occurred during execution of a Modbus function, a message is sent by the server but the event counter is not incremented.
80	–	1	08	Check the server status with the diagnostic code 0x0000 (return loop test – the server sends the request back): • 1 WORD per call
81	–	1	08	Reset the event counter of the server with the diagnostic code 0x000A: • 1 WORD per call
101	0 to 65,535	1 to 2,000	01	Read 1 to 2,000 output bits on the remote address 0 to 65,535
102	0 to 65,535	1 to 2,000	02	Read 1 to 2,000 input bits on the remote address 0 to 65,535
103	0 to 65,535	1 to 125	03	Read 1 to 125 holding registers on the remote address 0 to 65,535
104	0 to 65,535	1 to 125	04	Read 1 to 125 input words on the remote address 0 to 65,535
105	0 to 65,535	1	05	Write 1 output bit on the remote address 0 to 65,535
106	0 to 65,535	1	06	Write 1 holding register on the remote address 0 to 65,535
115	0 to 65,535	1 to 1,968	15	Write 1 to 1,968 output bits on the remote address 0 to 65,535
116	0 to 65,535	1 to 123	16	Write 1 to 123 holding registers on the remote address 0 to 65,535

Figure 4: Simatic S7-1200 / modbusMode

5.3.1.3 PITreader – Modbus data areas

The product supports the following Modbus/TCP data areas:

- Discrete Inputs (Bit)
  PITreader -> Modbus Client, bit access read (with FC02)

Address	Contents
1x4001	Is authenticated (data from the transponder key)

- Input Register (Word/16 Bits)
  PITreader -> Modbus Client, register access read (with FC04)

Address	Contents
3x0001 … 3x0002	PITreader order number
3x0003 ... 3x0004	PITreader serial number
3x0005 ... 3x0006	Operating hours counter in minutes
3x0007 ... 3x0008	RTC time stamp, seconds since 01.01.2000 00:00 (UTC)
3x0009	LED colour
3x0010	LED flash mode
3x0011	Diagnostic status
3x0025 … 3x0028	Security ID (data from the transponder key)
3x0029 … 3x0030	Reserved
3x0031 … 3x0032	Permission (code word)
3x0033	Permission (integer, 0 to 64)
3x0034	Authentication status
3x0035 … 3x0036	Order number
3x0037 … 3x0038	Serial number
3x0039	Reserved
3x0040	Reserved
3x0059 … 3x0060	Group 0
3x0061 … 3x0062	Group 1
3x0063 … 3x0064	Group 2
3x0065 … 3x0066	Group 3
3x0067 … 3x0068	Group 4
3x0069 … 3x0070	Group 5

Figure 5: PITreader Modbus Register (1)

Address	Contents
3x0071 … 3x0072	Group 6
3x0073 … 3x0074	Group 7
3x0075 … 3x0076	Group 8
3x0077 … 3x0078	Group 9
3x0079 … 3x0080	Group 10
3x0081 … 3x0082	Group 11
3x0083 … 3x0084	Group 12
3x0085 … 3x0086	Group 13
3x0087 … 3x0088	Group 14
3x0089 … 3x0090	Group 15
3x0091 … 3x0092	Group 16
3x0093 … 3x0094	Group 17
3x0095 … 3x0096	Group 18
3x0097 … 3x0098	Group 19
3x0099 … 3x0100	Group 20
3x0101 … 3x0102	Group 21
3x0103 … 3x0104	Group 22
3x0105 … 3x0106	Group 23
3x0107 … 3x0108	Group 24
3x0109 … 3x0110	Group 25
3x0111 … 3x0112	Group 26
3x0113 … 3x0114	Group 27
3x0115 … 3x0116	Group 28
3x0117 … 3x0118	Group 29
3x0119 … 3x0120	Group 30
3x0121 …3x0122	Group 31

- Holding Register (Word/16 Bits)
  Modbus Client -> PITreader, register access read (with FC03) and write (with FC06 or FC16)

Address	Contents
4x6001	Overwrite colour (PITreader LED access)
4x6002	Overwrite flash mode (PITreader LED access)
4x6003	Activate (=1) or deactivate (=0) overwrite

Figure 6: PITreader Modbus Register (2)

5.3.2 Request Modbus Data

To request the Modbus data the bit "request" must change from "false" to "true".

At the output "status" you can see whether the data request is successfully executed.
The error codes can be found in the online help and in the following figure.

5.3.2.1 Output parameter “Status” / Error codes

Parameter STATUS

Parameter STATUS (general status information)

STATUS* (W#16#)	Description
0000	Instruction executed without errors.
0001	Connection established.
0003	Connection terminated.
7000	No job active and no connection established (REQ=0, DISCONNECT=1).
7001	Connection establishment triggered.
7002	Intermediate call. Connection is being established.
7003	Connection is being terminated.
7004	Connection established and monitored. No job processing active.
7005	Data is being sent.
7006	Data is being received.
* The status codes can be displayed as integer or hexadecimal values in the program editor. For information on switching the display formats, refer to "See also".	

Figure 7: Simatic S7-1200 / Parameter Status (1)

Parameter STATUS (protocol error)

STATUS* (W#16#)	Local and/or remote errors	Error code in the answer from MB_SERVER (B#16#)	Description
80C8	Local	–	No response of the server in the defined period. Check the connection to the Modbus server. This error is only reported on completion of the configured repeated attempts. If the "MB_CLIENT" instruction does not receive an answer with the originally transferred transaction ID (see static tag MB_TRANSACTION_ID) within the defined period, this error code is output.
8380	Local	–	Received Modbus frame has incorrect format or too few bytes were received.
8381	Remote	01	Function code is not supported.
8382	Local	–	• The length of the Modbus frame in the frame header does not match the number of received bytes. • The number of bytes does not match the number of actually transmitted bytes (only functions 1–4). For example, this is the case when "MB_CLIENT" requests an odd number of words, but "MB_SERVER" always sends an even number of words. • The start address in the received frame does not match the saved start address (functions 5, 6, 15, 16). • The number of words does not match the number of actually transmitted words (functions 15 and 16).
	Remote	03	Invalid length specification in received Modbus frame. Check the server side.
8383	Local	–	Error reading or writing data or access outside the address area of MB_DATA_PTR.
	Remote	02	Error reading or writing data or access outside the address area of the server
8384	Local	–	• Invalid exception code received. • A different data value was received than was originally sent by the client (functions 5, 6 and 8). • Invalid status value received (function 11)
	Remote	03	Error in data value for function 5

Figure 8: Simatic S7-1200 / Parameter Status (2)

8385	Local	–	• Diagnostics code not supported. • A different subfunction code was received than was originally sent by the client (function 8).
	Remote	03	Diagnostics code not supported
8386	Local	–	Received function code does not match the one sent originally.
8387	Local	–	The protocol ID of the Modbus TCP frame received by the server is not "0".
8388	Local	–	The Modbus server sent a different data length than was requested. This error occurs only when using the Modbus functions 5, 6, 15 or 16.
* The status codes can be displayed as integer or hexadecimal values in the program editor. For information on switching the display formats, refer to "See also".			

Parameter STATUS (parameter error)

STATUS* (W#16#)	Description
80B6	Invalid connection type, only TCP connections are supported.
80BB	Invalid value at ActiveEstablished parameter (identifier for the type of connection establishment, see CONNECT parameter): • Only passive connection establishment permitted for server (ActiveEstablished = FALSE). • Only active connection establishment permitted for client (ActiveEstablished = TRUE).
8188	The MB_MODE parameter has an invalid value.
8189	Invalid addressing of data at the MB_DATA_ADDR parameter.
818A	Invalid data length at the MB_DATA_LEN parameter.
818B	The MB_DATA_PTR parameter has an invalid pointer. You should also check the values of the MB_DATA_ADDR and MB_DATA_LEN parameters.
818C	Timeout at parameter BLOCKED_PROC_TIMEOUT or RCV_TIMEOUT (see static tags of instruction). BLOCKED_PROC_TIMEOUT and RCV_TIMEOUT must be between 0.5 s and 55 s.
8200	• A different Modbus request is currently being processed via the port. • Another instance of MB_CLIENT with the same connection parameters is processing an existing Modbus request.
* The status codes can be displayed as integer or hexadecimal values in the program editor. For information on switching the display formats, refer to "See also".	

Figure 9: Simatic S7-1200 / Parameter Status (3)

Note

Error codes of internally used communications instructions

With the "MB_CLIENT" instruction, in addition to the errors listed in the tables, errors caused by the communication instructions ("TCON", "TDISCON", "TSEND", "TRCV", "T_DIAG" and "TRESET") used by the instruction can occur.

The error codes are assigned via the instance data block of the "MB_CLIENT" instruction. The error codes are displayed for the respective instruction under STATUS in the "Static" section.

The meaning of the error codes is available in the documentation of the corresponding communications instruction.

Note

Communication error when sending or receiving data

If a communication error occurs when sending or receiving data (80C4 (Temporary communications error. The specified connection is temporarily down.), 80C5 (Remote partner closed connection actively.), 80A1 (The specified connection is disconnected or is not yet established.)), the existing connection is terminated.

This also means that you can see all STATUS values that are returned when the connection is terminated and that the STATUS code that caused the connection to be terminated is only output when the connection is terminated.

Example: If a temporary communication error occurs when data is received, the STATUS 7003 (ERROR=false) is output initially and then 80C4 (ERROR=true).

Figure 10: Simatic S7-1200 / Parameter Status (4)

5.3.3 DB4 HoldingRegisterRead

The data received by the PITreader is stored in the DB4 “HoldingRegisterRead”.

5.3.4 Monitoring Modbus Data

Monitoring the Received Modbus Data in DB4

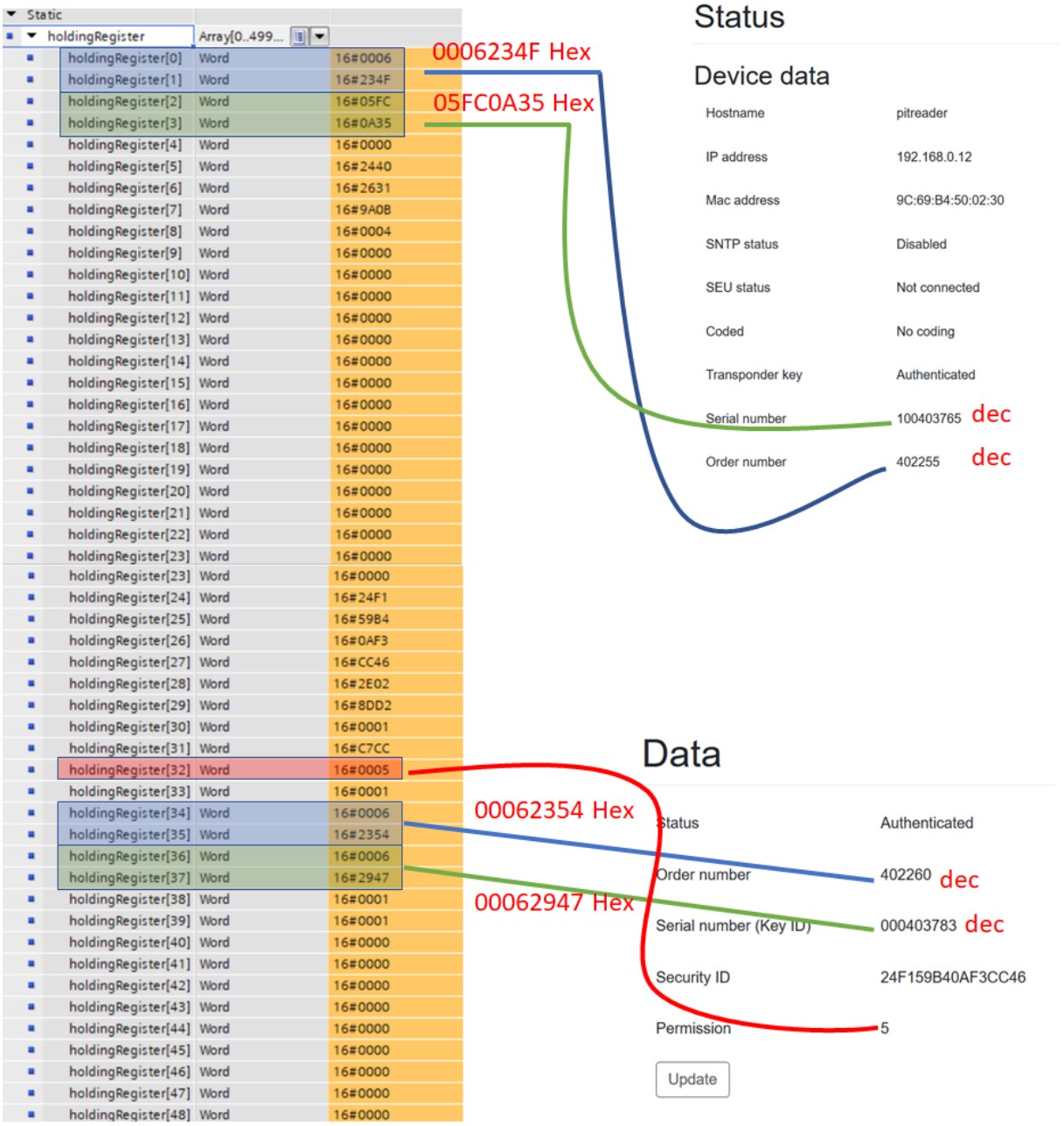

Figure 11: Simatic S7-1200 / Monitoring Modbus Data

6 Siemens S7-1500

6.1 Modify the PITreader IP address

6.2 Example project

The Modbus example project can be downloaded from Siemens.
https://support.industry.siemens.com/cs/document/102020340/how-do-you-program-and-parameterize-modbus-tcp-communication-between-s7-1500-cpus-and-s7-1200-cpus-?dti=0&lc=en-DE

The example project contains two different devices. The S7-1200 and the S7-1500 hardware.
For the Application Note the example project with the suitable hardware was used.

INFORMATION

There are Modbus Client and Modbus Server blocks. Because the PITreader is a Modbus server, the Modbus client block must be used on the PLC side.

A few adjustments need to be made in the example project.
This is described on the following pages.

6.3 FB1 “ModbusClient”

The function block FB1 "ModbusClient" is called cyclically in OB1.

The FB1 "ModbusClient" calls the "MB_CLIENT" instruction internally to establish the Modbus/TCP connection and read the holding register from the Modbus TCP server.
The communication request to read the holding register is controlled via the "ModbusData".clientData.request tag at the "request" input.
In this example the Modbus TCP connection with connection number=2 is established to Port 502 of the Modbus TCP server. The Modbus TCP server has the IP address 192.168.0.12.
122 holding registers are read on the remote address 0. For this you set the input parameters "modbusMode", "modbusDataAddress" and "modbusDataLen" as follows:

- modbusMode = 103
- modebusDataAddress = 0
- modbusDataLen = 122

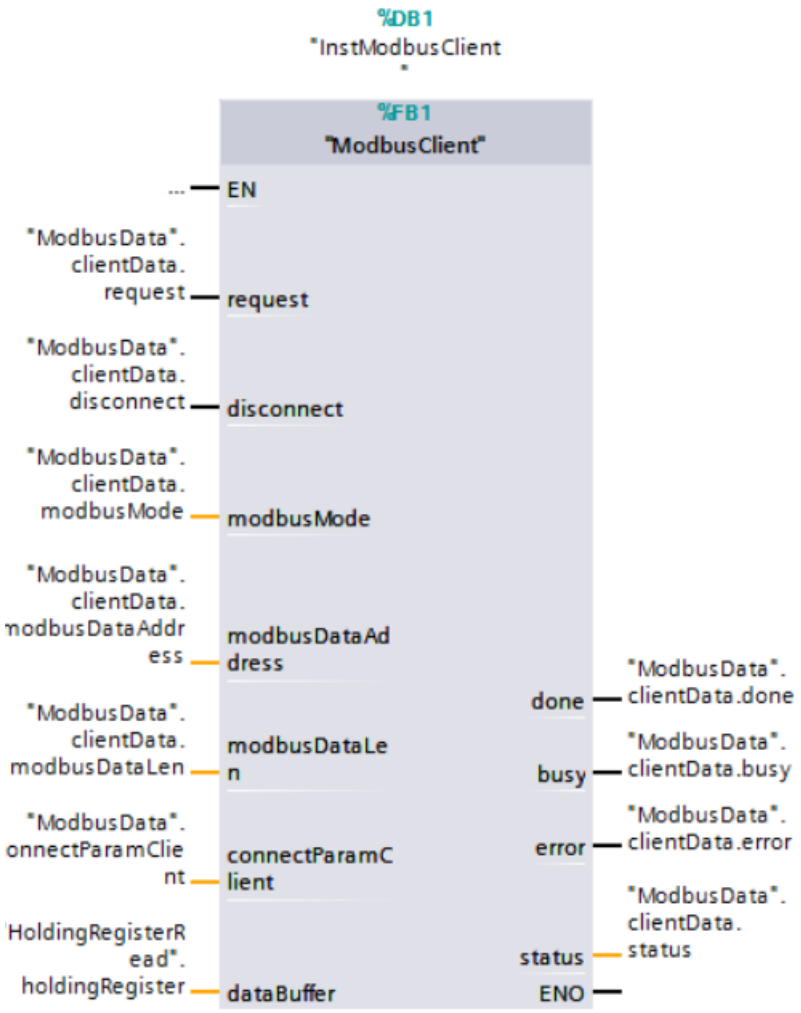

Figure 12: Simatic S7-1500 / FB1 – Modbus Client

6.3.1 Configuring DB100 “connection data”

6.3.1.1 Modbus parameter

In DB100 the IP address of the connection partner
- RemoteAddress
- modbusMode
- modbusDataAddress
- modbusDataLen

must be adjusted.

The Modbus registers of the PITreader are described in the operating manual and a short summary from 🕮 26.

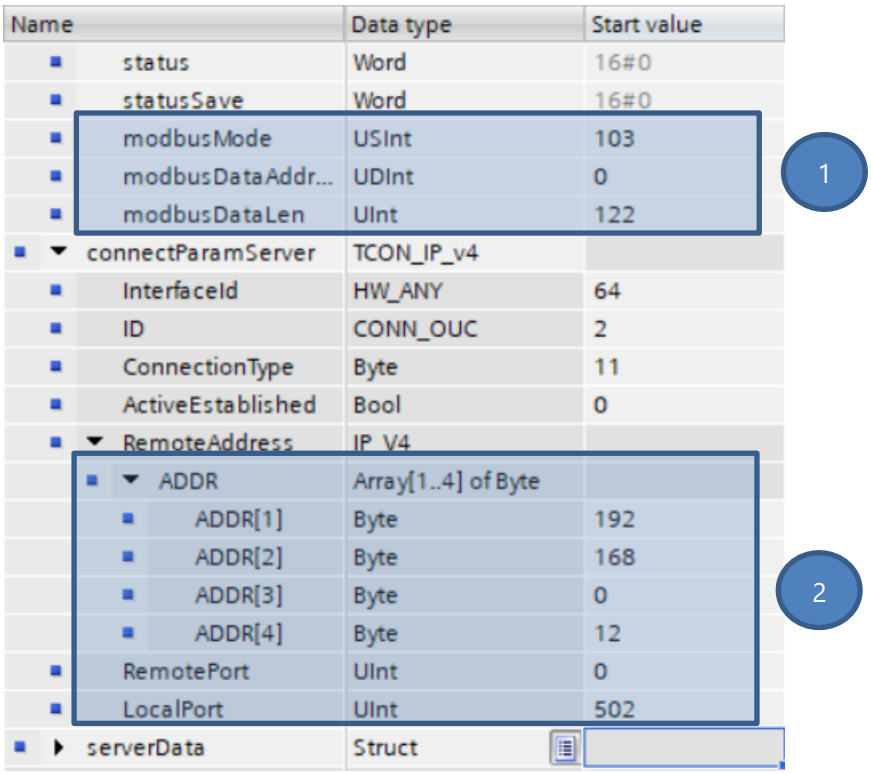

Figure 13: Simatic S7-1500 / DB100 – Connection Data

1

- modbusMode: In the following table [📖 25] you read the "modbusMode" according to the data you want to read out from the PITreader.
- modbusDataAddress: This is the start address of the requested Modbus Register.

INFORMATION

On the PITreader, the addressing for Modbus/TCP data areas starts at "1".
On the Simatic PLCs addressing start at "0".

- modbusDataLen: Look in on the PITreader user manual to find out which data you need and enter the required data length.

2

- RemoteAddress in Dec:
  - ADDR[1]: 192
  - ADDR[2]:168
  - ADDR[3]: 0
  - ADDR[4]: 12
- RemotePort: 502

6.3.1.2 Simatic S7-1500 / modbusMode

MB_MODE, MB_DATA_ADDR and MB_DATA_LEN parameters

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Modbus function	Function and data type
0	1 to 9,999	1 to 2,000	01	Read 1 to 2,000 output bits on the remote address 0 to 9,998
0	10,001 to 19,999	1 to 2,000	02	Read 1 to 2,000 input bits on the remote address 0 to 9,998
0	• 40,001 to 49,999 • 400,001 to 465,535	1 to 125	03	• Read 1 to 125 holding registers on the remote address 0 to 9,998 • Read 1 to 125 holding registers on the remote address 0 to 65,534
0	30,001 to 39,999	1 to 125	04	Read 1 to 125 input words on the remote address 0 to 9,998
1	1 to 9,999	1	05	Write 1 output bit on the remote address 0 to 9,998
1	• 40,001 to 49,999 • 400,001 to 465,535	1	06	• Write 1 holding register on the remote address 0 to 9,998 • Write 1 holding register on the remote address 0 to 65,534
1	1 to 9,999	2 to 1,968	15	Write 2 to 1,968 output bits on the remote address 0 to 9,998
1	• 40,001 to 49,999 • 400,001 to 465,535	2 to 123	16	• Write 2 to 123 holding registers on the remote address 0 to 9,998 • Write 2 to 123 holding registers on the remote address 0 to 65,534
2	1 to 9,999	1 to 1,968	15	Write 1 to 1,968 output bits on the remote address 0 to 9,998
2	• 40,001 to 49,999 • 400,001 to 465,535	1 to 123	16	• Write 1 to 123 holding registers on the remote address 0 to 9,998 • Write 1 to 123 holding registers on the remote address 0 to 65,534
11	The MB_DATA_ADDR and MB_DATA_LEN parameters are not evaluated when this function is executed.		11	Read status word and event counter of the server: • The status word reflects the the processing status (0 – not processing, 0xFFFF – processing). • The event counter is incremented when the Modbus request was executed successfully. If an error occurred during execution of a Modbus function, a message is sent by the server but the event counter is not incremented.
80	–	1	08	Check the server status with the diagnostic code 0x0000 (return loop test – the server sends the request back): • 1 WORD per call
81	–	1	08	Reset the event counter of the server with the diagnostic code 0x000A: • 1 WORD per call
101	0 to 65,535	1 to 2,000	01	Read 1 to 2,000 output bits on the remote address 0 to 65,535
102	0 to 65,535	1 to 2,000	02	Read 1 to 2,000 input bits on the remote address 0 to 65,535
103	0 to 65,535	1 to 125	03	Read 1 to 125 holding registers on the remote address 0 to 65,535
104	0 to 65,535	1 to 125	04	Read 1 to 125 input words on the remote address 0 to 65,535
105	0 to 65,535	1	05	Write 1 output bit on the remote address 0 to 65,535
106	0 to 65,535	1	06	Write 1 holding register on the remote address 0 to 65,535
115	0 to 65,535	1 to 1,968	15	Write 1 to 1,968 output bits on the remote address 0 to 65,535
116	0 to 65,535	1 to 123	16	Write 1 to 123 holding registers on the remote address 0 to 65,535

Figure 14: Simatic S7-1500 / modbusMode

6.3.1.3 PITreader – Modbus data areas

The product supports the following Modbus/TCP data areas:

- Discrete Inputs (Bit)
  PITreader -> Modbus Client, bit access read (with FC02)

Address	Contents
1x4001	Is authenticated (data from the transponder key)

- Input Register (Word/16 Bits)
  PITreader -> Modbus Client, register access read (with FC04)

Address	Contents
3x0001 … 3x0002	PITreader order number
3x0003 ... 3x0004	PITreader serial number
3x0005 ... 3x0006	Operating hours counter in minutes
3x0007 ... 3x0008	RTC time stamp, seconds since 01.01.2000 00:00 (UTC)
3x0009	LED colour
3x0010	LED flash mode
3x0011	Diagnostic status
3x0025 … 3x0028	Security ID (data from the transponder key)
3x0029 … 3x0030	Reserved
3x0031 … 3x0032	Permission (code word)
3x0033	Permission (integer, 0 to 64)
3x0034	Authentication status
3x0035 … 3x0036	Order number
3x0037 … 3x0038	Serial number
3x0039	Reserved
3x0040	Reserved
3x0059 … 3x0060	Group 0
3x0061 … 3x0062	Group 1
3x0063 … 3x0064	Group 2
3x0065 … 3x0066	Group 3
3x0067 … 3x0068	Group 4
3x0069 … 3x0070	Group 5

Figure 15: PITreader Modbus Register (1)

Address	Contents
3x0071 … 3x0072	Group 6
3x0073 … 3x0074	Group 7
3x0075 … 3x0076	Group 8
3x0077 … 3x0078	Group 9
3x0079 … 3x0080	Group 10
3x0081 … 3x0082	Group 11
3x0083 … 3x0084	Group 12
3x0085 … 3x0086	Group 13
3x0087 … 3x0088	Group 14
3x0089 … 3x0090	Group 15
3x0091 … 3x0092	Group 16
3x0093 … 3x0094	Group 17
3x0095 … 3x0096	Group 18
3x0097 … 3x0098	Group 19
3x0099 … 3x0100	Group 20
3x0101 … 3x0102	Group 21
3x0103 … 3x0104	Group 22
3x0105 … 3x0106	Group 23
3x0107 … 3x0108	Group 24
3x0109 … 3x0110	Group 25
3x0111 … 3x0112	Group 26
3x0113 … 3x0114	Group 27
3x0115 … 3x0116	Group 28
3x0117 … 3x0118	Group 29
3x0119 … 3x0120	Group 30
3x0121 …3x0122	Group 31

- Holding Register (Word/16 Bits)
  Modbus Client -> PITreader, register access read (with FC03) and write (with FC06 or FC16)

Address	Contents
4x6001	Overwrite colour (PITreader LED access)
4x6002	Overwrite flash mode (PITreader LED access)
4x6003	Activate (=1) or deactivate (=0) overwrite

Figure 16: PITreader Modbus Register (2)

6.3.2 Request Modbus Data

To request the Modbus data the bit "request" must change from "false" to "true".

At the output "status" you can see whether the data request is successfully executed.
The error codes can be found in the online help and in the following figure.

6.3.2.1 Output parameter “Status” / Error codes

Parameter STATUS

Parameter STATUS (general status information)

STATUS* (W#16#)	Description
0000	Instruction executed without errors.
0001	Connection established.
0003	Connection terminated.
7000	No job active and no connection established (REQ=0, DISCONNECT=1).
7001	Connection establishment triggered.
7002	Intermediate call. Connection is being established.
7003	Connection is being terminated.
7004	Connection established and monitored. No job processing active.
7005	Data is being sent.
7006	Data is being received.
* The status codes can be displayed as integer or hexadecimal values in the program editor. For information on switching the display formats, refer to "See also".	

Figure 17: Simatic S7-1500 / Parameter Status (1)

Parameter STATUS (protocol error)

STATUS* (W#16#)	Local and/or remote errors	Error code in the answer from MB_SERVER (B#16#)	Description
80C8	Local	–	No response of the server in the defined period. Check the connection to the Modbus server. This error is only reported on completion of the configured repeated attempts. If the "MB_CLIENT" instruction does not receive an answer with the originally transferred transaction ID (see static tag MB_TRANSACTION_ID) within the defined period, this error code is output.
8380	Local	–	Received Modbus frame has incorrect format or too few bytes were received.
8381	Remote	01	Function code is not supported.
8382	Local	–	• The length of the Modbus frame in the frame header does not match the number of received bytes. • The number of bytes does not match the number of actually transmitted bytes (only functions 1–4). For example, this is the case when "MB_CLIENT" requests an odd number of words, but "MB_SERVER" always sends an even number of words. • The start address in the received frame does not match the saved start address (functions 5, 6, 15, 16). • The number of words does not match the number of actually transmitted words (functions 15 and 16).
	Remote	03	Invalid length specification in received Modbus frame. Check the server side.
8383	Local	–	Error reading or writing data or access outside the address area of MB_DATA_PTR.
	Remote	02	Error reading or writing data or access outside the address area of the server
8384	Local	–	• Invalid exception code received. • A different data value was received than was originally sent by the client (functions 5, 6 and 8). • Invalid status value received (function 11)
	Remote	03	Error in data value for function 5

Figure 18: Simatic S7-1500 / Parameter Status (2)

8385	Local	–	• Diagnostics code not supported. • A different subfunction code was received than was originally sent by the client (function 8).
	Remote	03	Diagnostics code not supported
8386	Local	–	Received function code does not match the one sent originally.
8387	Local	–	The protocol ID of the Modbus TCP frame received by the server is not "0".
8388	Local	–	The Modbus server sent a different data length than was requested. This error occurs only when using the Modbus functions 5, 6, 15 or 16.
* The status codes can be displayed as integer or hexadecimal values in the program editor. For information on switching the display formats, refer to "See also".			

Parameter STATUS (parameter error)

STATUS* (W#16#)	Description
80B6	Invalid connection type, only TCP connections are supported.
80BB	Invalid value at ActiveEstablished parameter (identifier for the type of connection establishment, see CONNECT parameter): • Only passive connection establishment permitted for server (ActiveEstablished = FALSE). • Only active connection establishment permitted for client (ActiveEstablished = TRUE).
8188	The MB_MODE parameter has an invalid value.
8189	Invalid addressing of data at the MB_DATA_ADDR parameter.
818A	Invalid data length at the MB_DATA_LEN parameter.
818B	The MB_DATA_PTR parameter has an invalid pointer. You should also check the values of the MB_DATA_ADDR and MB_DATA_LEN parameters.
818C	Timeout at parameter BLOCKED_PROC_TIMEOUT or RCV_TIMEOUT (see static tags of instruction). BLOCKED_PROC_TIMEOUT and RCV_TIMEOUT must be between 0.5 s and 55 s.
8200	• A different Modbus request is currently being processed via the port. • Another instance of MB_CLIENT with the same connection parameters is processing an existing Modbus request.
* The status codes can be displayed as integer or hexadecimal values in the program editor. For information on switching the display formats, refer to "See also".	

Figure 19: Simatic S7-1500 / Parameter Status (3)

Note

Error codes of internally used communications instructions

With the "MB_CLIENT" instruction, in addition to the errors listed in the tables, errors caused by the communication instructions ("TCON", "TDISCON", "TSEND", "TRCV", "T_DIAG" and "TRESET") used by the instruction can occur.

The error codes are assigned via the instance data block of the "MB_CLIENT" instruction. The error codes are displayed for the respective instruction under STATUS in the "Static" section.

The meaning of the error codes is available in the documentation of the corresponding communications instruction.

Note

Communication error when sending or receiving data

If a communication error occurs when sending or receiving data (80C4 (Temporary communications error. The specified connection is temporarily down.), 80C5 (Remote partner closed connection actively.), 80A1 (The specified connection is disconnected or is not yet established.)), the existing connection is terminated.

This also means that you can see all STATUS values that are returned when the connection is terminated and that the STATUS code that caused the connection to be terminated is only output when the connection is terminated.

Example: If a temporary communication error occurs when data is received, the STATUS 7003 (ERROR=false) is output initially and then 80C4 (ERROR=true).

Figure 20: Simatic S7-1500 / Parameter Status (4)

6.3.3 DB4 HoldingRegisterRead

The data received by the PITreader is stored in the DB4 “HoldingRegisterRead”.

6.3.4 Monitoring Modbus Data

Monitoring the Received Modbus Data in DB4

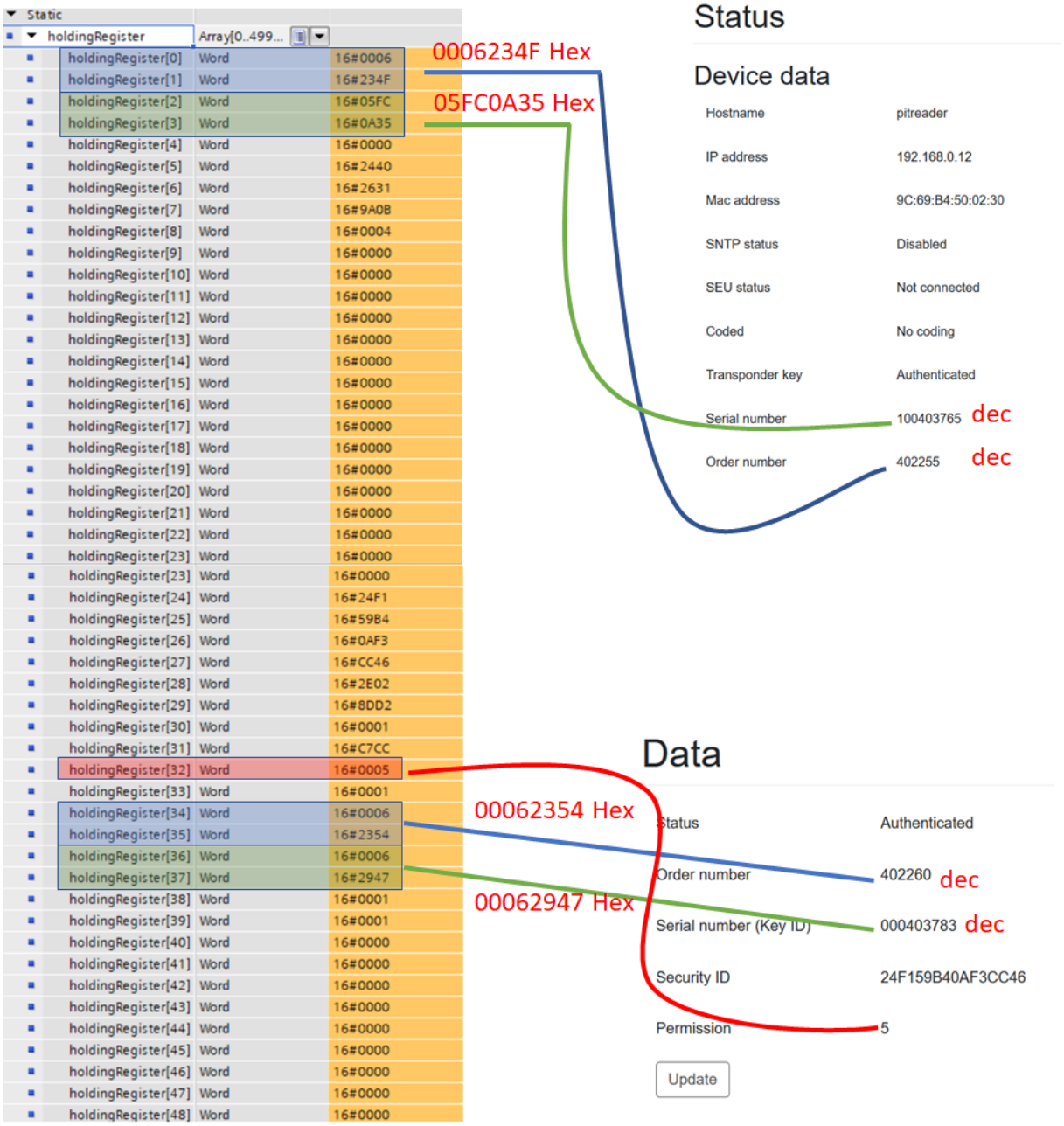

Figure 21: Simatic S7-1500 / Monitoring Modbus Data

7 Rockwell ControlLogix

In progress

7.1 Example project

The Modbus example project can be downloaded from Rockwell Homepage.

ID 101037
https://www.rockwellautomation.com/search/ra_en_GLOBAL;keyword=101037;startIndex=0;activeTab=Sample_Code;spellingCorrect=true;facets=;languages=en;locales=en_GLOBAL;sort=bma;isPLS=false;sessionID=607557b3-17c4-d953-6c82-df47b983fa35;deepLinking=false

8 Kunbus RevPi

8.1 Prepare RevPi as Modbus Client

8.1.1 Activate Modbus Master on RevPi

- Navigate the Webbrowser to the IP-Address of the RevPi

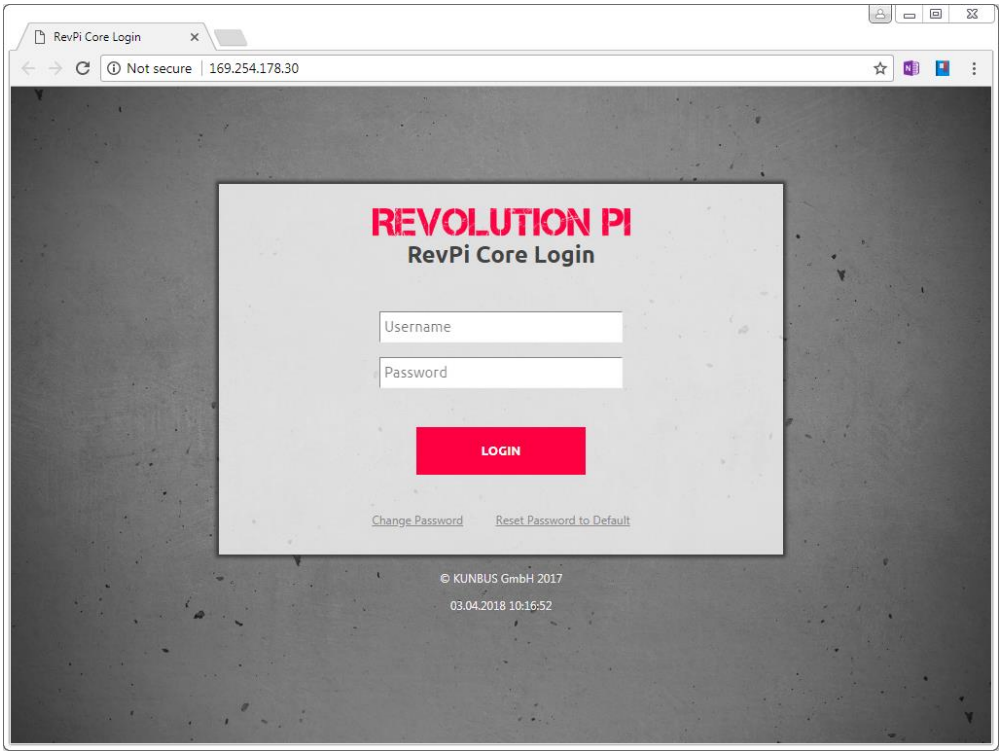

Figure 22: Login page

- Login as user “admin” with the associated password

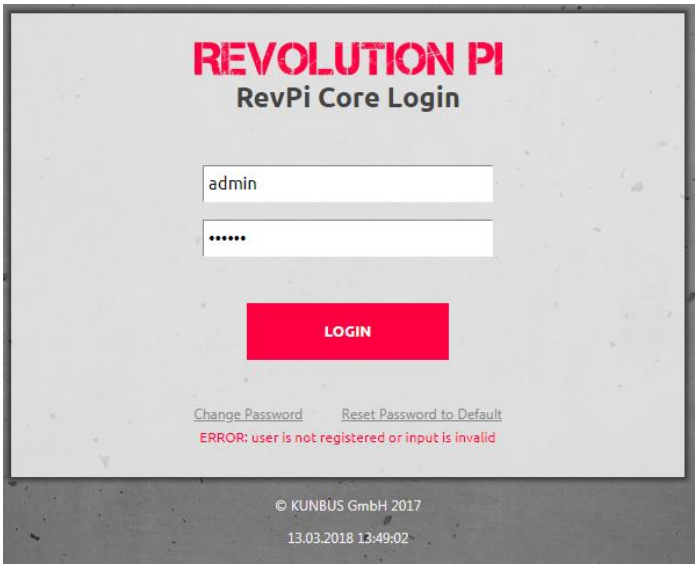

Figure 23: Login as user "admin"

- Switch to “SERVICES” tab

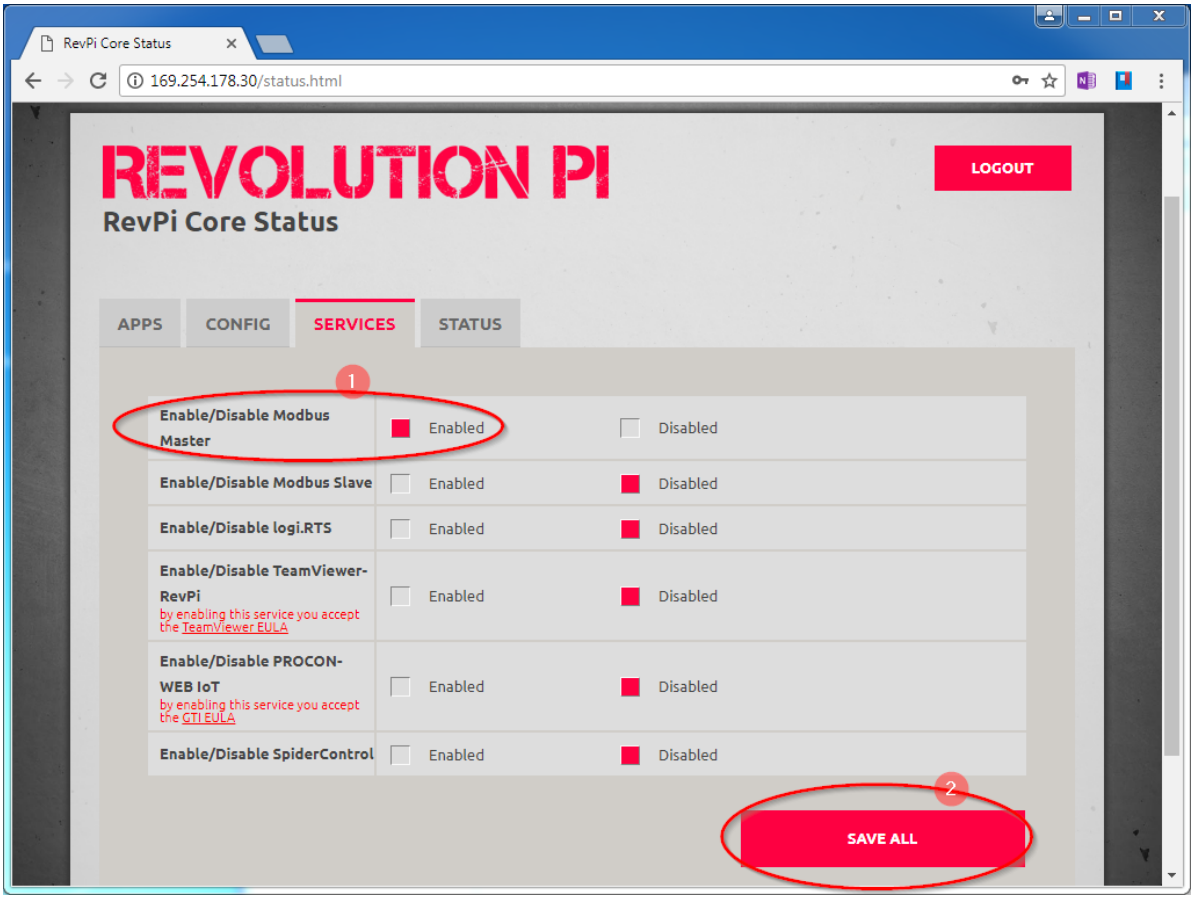

Figure 24: Website of RevPi / Service page

(1) Enable Modbus Master
(2) Save Settings

8.1.2 Check the RevPi Core version

- Open a terminal window to RevPi. In this example we are using PuTTY

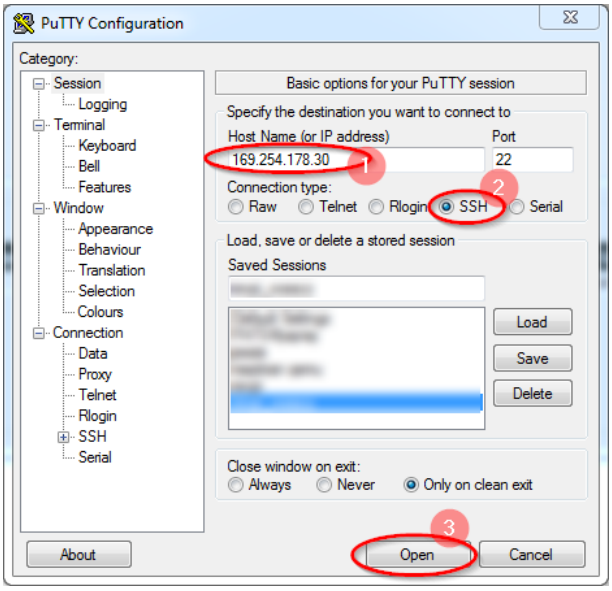

Figure 25: PuTTY

(1) Enter IP-Address of RevPi
(2) Select SSH as protocol
(3) open the connection

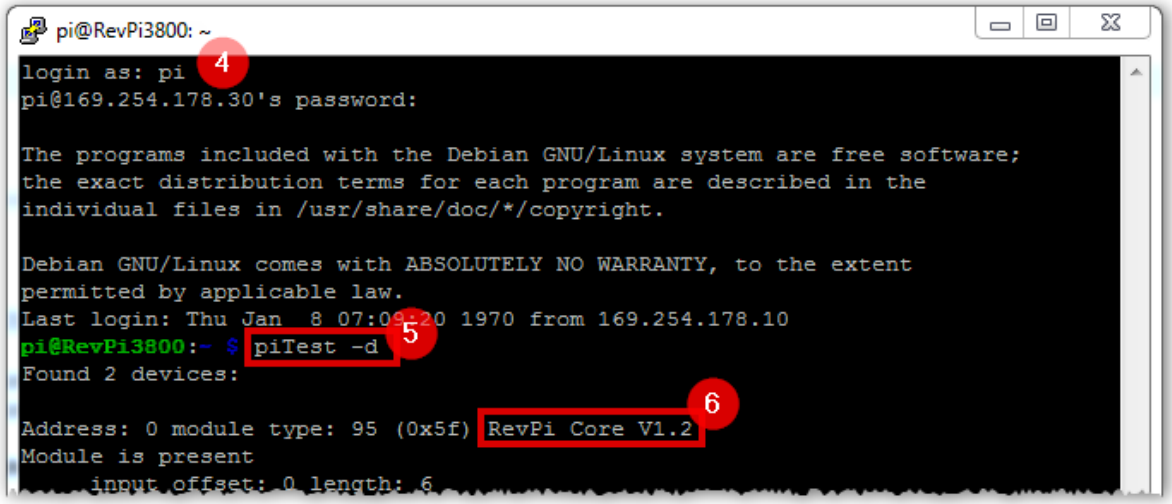

Figure 26: Terminal Window RevPi

(4) login as user "pi"
(5) Execute the following command: piTest –d
(6) Remember the version of the RevPi. This is needed later to configure the RevPi. In this case "RevPi Core V1.2"

8.1.3 Configure Modbus Master via PiCtory

- Navigate the Webbrowser to the IP-Address of the RevPi and login as “admin” like done before in section 8.1.1.
- Start PiCtory to configure the RevPi.

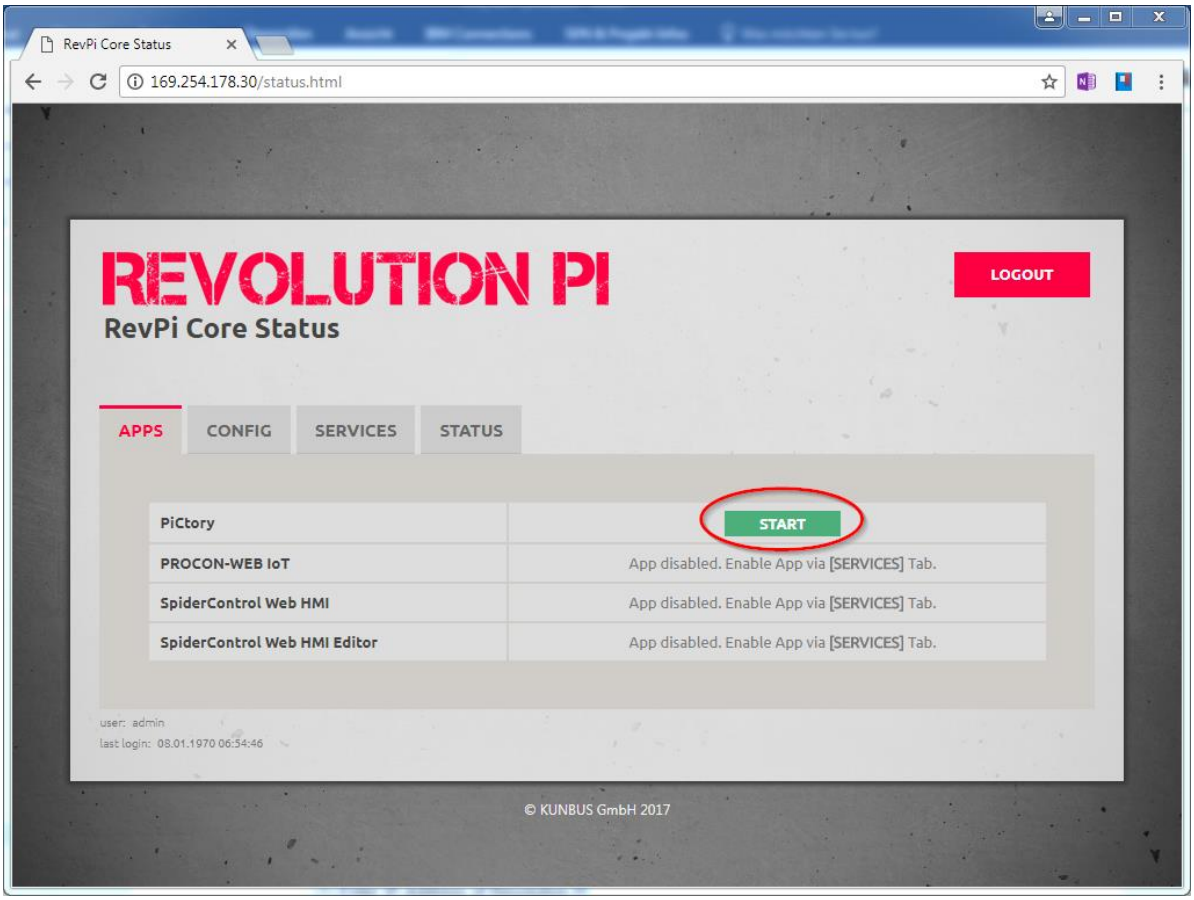

Figure 27: Website of RevPi / Apps page

- Drag’n’Drop the device with the previously found version: RevPi Core V1.2

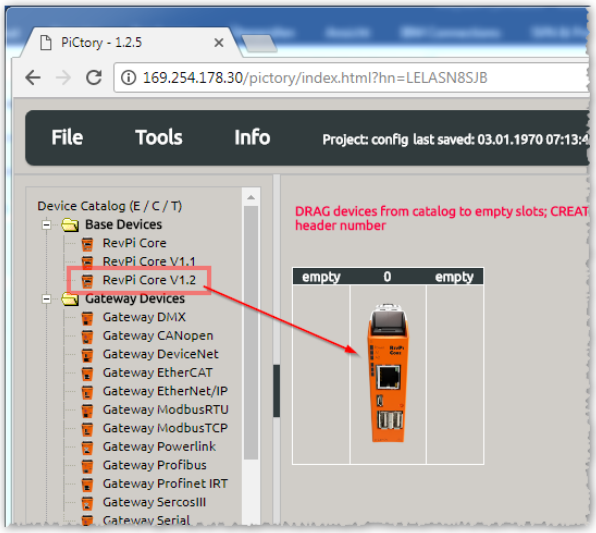

Figure 28: PiCtory device configuration

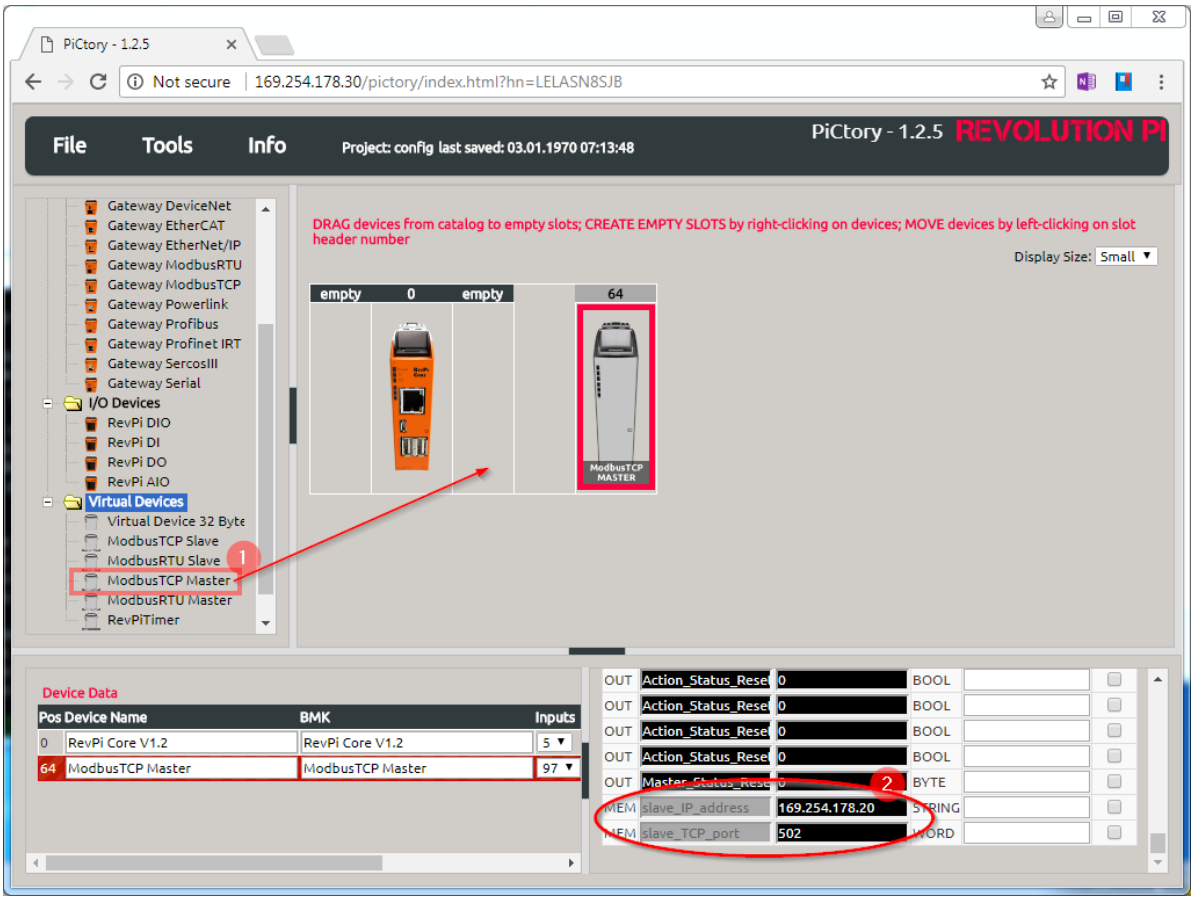

Figure 29: Add Modbus/TCP Master

(1) Drag'n'Drop the virtual device "ModbusTCP MASTER" to an empty slot right of the RevPi device.

(2) Select the "ModbusTCP MASTER" device and parameterize IP address and port of the slave to which the master is to be connected.

8.1.4 Configure extended data of the Modbus Master

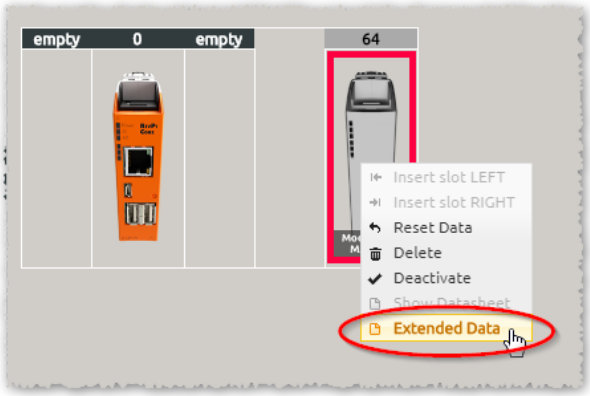

Figure 30: Opening extended data

- Open context menu by a right click on the "ModbusTCP MASTER" device.
- Select "Extended Data" to configure the process data of interest.

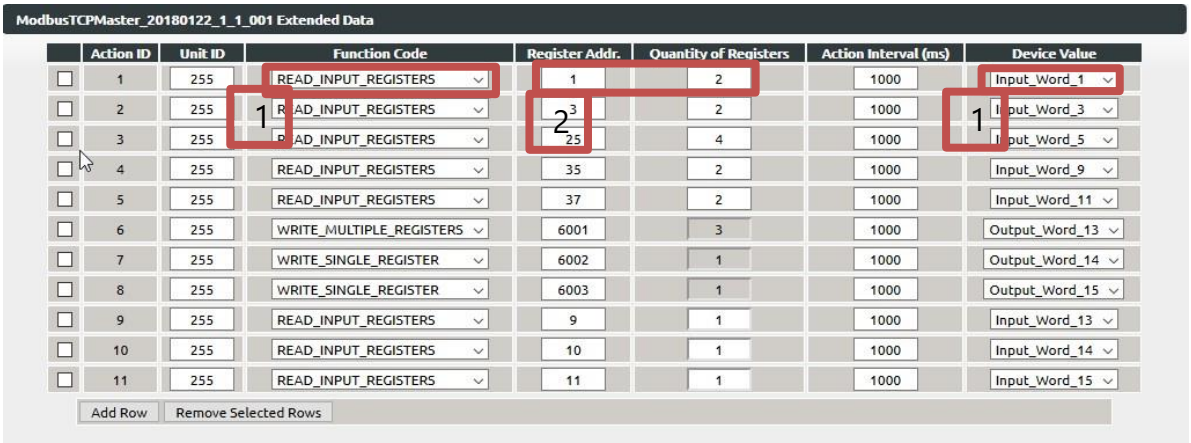

Figure 31: Editing extended data of the virtual ModbusTCP Master

(1) Select the required function code
(2) Set the requested data,- start at register 1 and fetch 2 registers
(3) Write fetched data to variable named "Input_Word_1". The data of the next register will be stored in "Input_Word_2".

Leave the parameter "Action Interval (ms)" at the default value of 1000ms. That means, the RevPi polls the corresponding Modbus register every 1000ms.

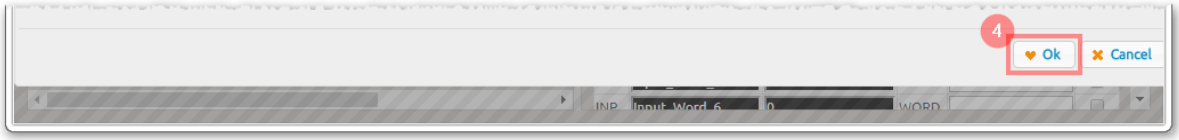

Figure 32: Confirmation of edited extended data

(4) Accept the settings with "Ok"

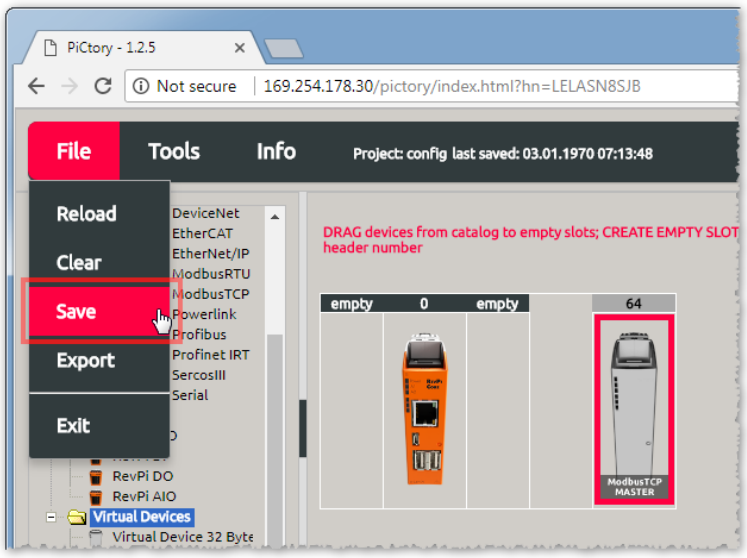

Figure 33: PiCtory – Save configuration

- Save configuration

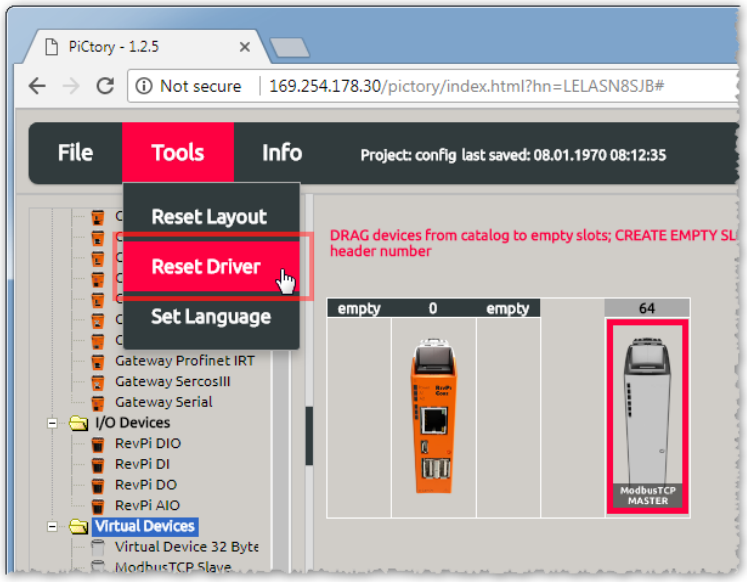

Figure 34: PiCtory – Reset driver

- Reset driver

8.2 Using the process data within the RevPi

One way to access the received Modbus variables is by using the command line tool “piTest” like shown below:

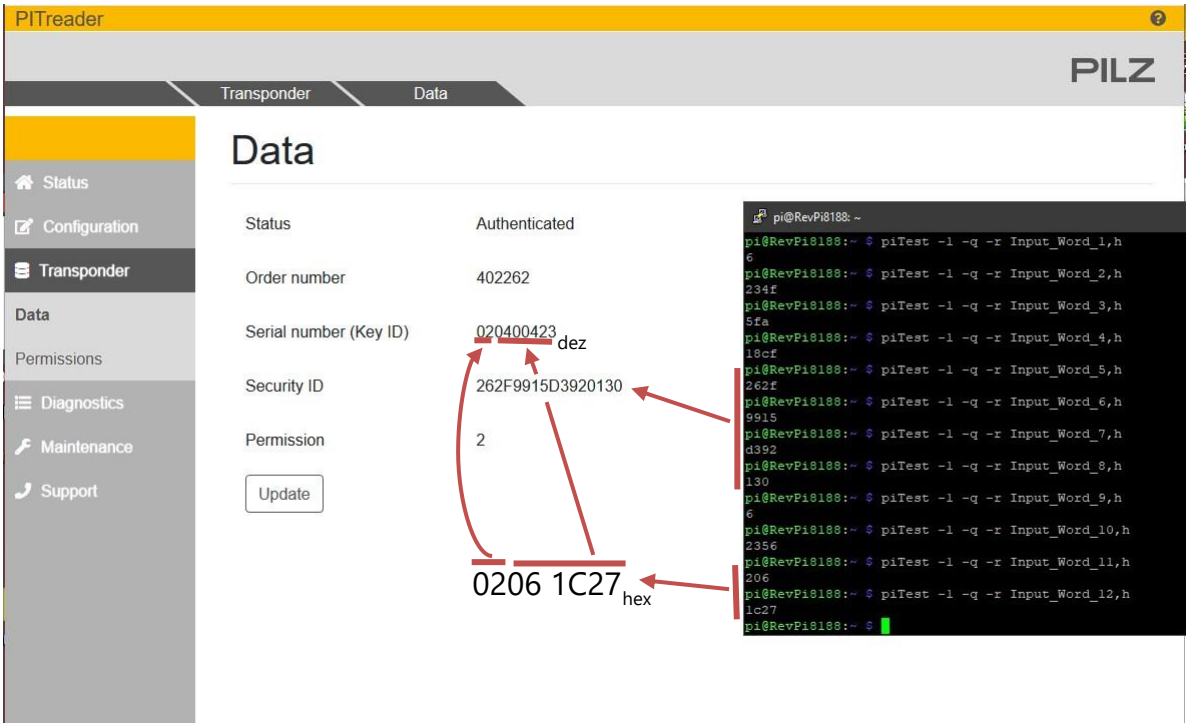

Figure 35: Read variables with the command “piTest”

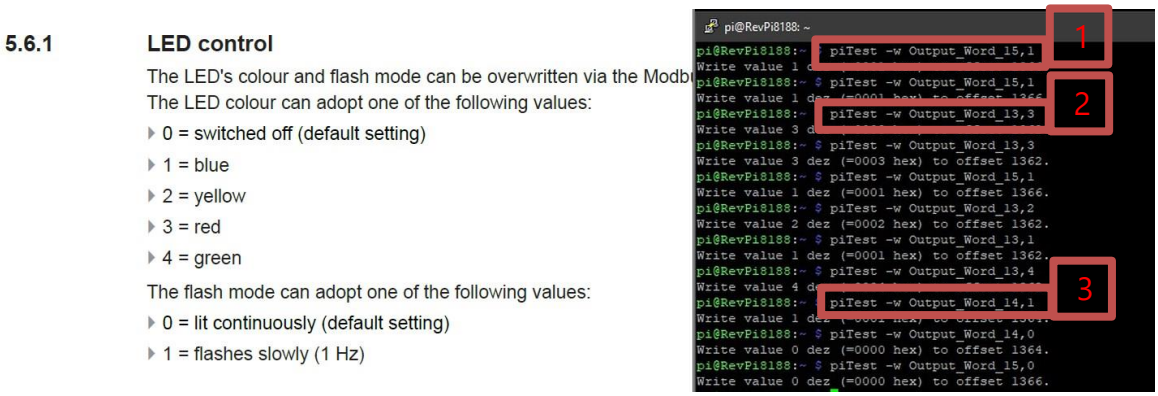

Figure 36: Write variables with the command “piTest”

[1] Enable Overwrite
[2] Set LED to red
[3] Activate flash mode
[4] Disable Overwrite

The command line tool “piTest” has serval options to get the process values of interest.
For a complete list of available options, perform the command without any options.

This command can be integrated in other scripting languages to further process the values according to the requirements. For example, to forward the data to a higher-level control system or a cloud provider. These could be, among others, the following:

- Amazon Web Service https://aws.amazon.com
- Microsoft Azure https://azure.microsoft.com
- IBM Cloud https://www.ibm.com/cloud
- Google Cloud https://cloud.google.com
- Telekom https://cloud.telekom.de/en

The corresponding documentation must then be consulted and checked for feasibility. In addition to Linux knowledge, additional knowledge regarding high-level programming languages or script languages may be required.

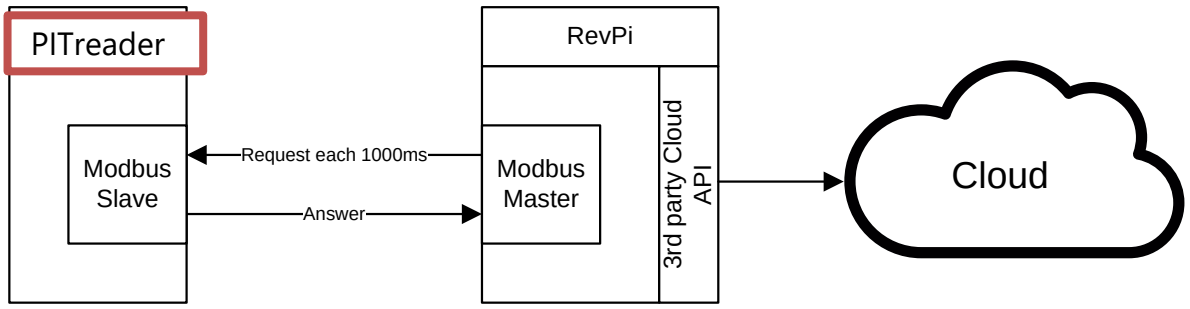

Figure 37: Example diagram

9 Table of figures

Support

Technical support is available from Pilz round the clock.

Americas
Brazil
+55 11 97569-2804
Canada
+1 888 315 7459
Mexico
+52 55 5572 1300
USA (toll-free)
+1 877-PILZUSA (745-9872)

Asia
China
+86 21 60880878-216
Japan
+81 45 471-2281
South Korea
+82 31 778 3300

Australia
+61 3 95600621

Europe
Austria
+43 1 7986263-0
Belgium, Luxembourg
+32 9 3217570
France
+33 3 88104003
Germany
+49 711 3409-444
Ireland
+353 21 4804983
Italy, Malta
+39 0362 1826711
Scandinavia
+45 74436332
Spain
+34 938497433
Switzerland
+41 62 88979-32
The Netherlands
+31 347 320477
Turkey
+90 216 5775552
United Kingdom
+44 1536 462203

You can reach our international hotline on:

support@pilz.com

Pilz develops environmentally-friendly products using ecological materials and energy-saving technologies. Offices and production facilities are ecologically designed, environmentally-aware and energy-saving. So Pilz offers sustainability, plus the security of using energy-efficient products and environmentally-friendly solutions.

Energy saving by Pilz

BLUECOMPETENCE Alliance Member
Partner of the Engineering Industry Sustainability Initiative

pilz
Partner of:
The Best of German Engineering

1005173-EN-01 Draft 01, 2019-06 Printed in Germany

We are represented internationally. Please refer to our homepage www.pilz.com for further details or contact our headquarters.

Headquarters: Pilz GmbH & Co. KG, Felix-Wankel-Straße 2, 73760 Ostfildern, Germany
Telephone: +49 711 3409-0, Telefax: +49 711 3409-133, E-Mail: info@pilz.com, Internet: www.pilz.com

PILZ
THE SPIRIT OF SAFETY