SIEMENS

SICAM RTUs

MODBUS TCP/IP

Interoperability

Preface, Table of Contents

Introduction 1

Interoperability of SICAM RTUs using MODBUS TCP/IP Master "Client" (MBCiA0) 2

Interoperability of SICAM RTUs using MODBUS/TCP Slave “Server” (MBSiA0) 3

MODBUS Data Formats A

Literature

DC0-151-2.01

Disclaimer of Liability
Although we have carefully checked the contents of this publication for conformity with the hardware and software described, we cannot guarantee complete conformity since errors cannot be excluded. The information provided in this manual is checked at regular intervals and any corrections that might become necessary are included in the next releases. Any suggestions for improvement are welcome.

Subject to change without prior notice.

Document label: SICRTUs-HBIntopMODBUSTCPIP-ENG_V2.01

Release date: 2017-12-22

Copyright
Copyright © Siemens AG 2017
The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Preface

This document is applicable to the following product(s):

- SICAM RTUs
  (SICAM AK 3, SICAM AK, SICAM TM, SICAM BC)

Purpose of this manual

This manual describes the interoperability of SICAM RTUs using protocol element according to MODBUS TCP/IP and essentially contains

- Interoperability MODBUS TCP/IP

Target Group

The document you are reading right now is addressed to users, who are in charge of the following tasks:

- Sales engineering and technical clarification
- Conceptual activities, as for example design and configuration

Table of Contents

1. Introduction 7

1.1. Area of Application 8
1.2. General Information 9

2. Interoperability of SICAM RTUs using MODBUS TCP/IP Master "Client" (MBCiA0) 11

2.1. Network Configurations 13
2.2. Physical layer 14
2.2.1. Electrical Interface 14
2.2.2. Transmission Speed 14
2.2.3. TCP-Port 14
2.2.4. Connections 14
2.3. Link Layer (MODBUS) 15
2.3.1. TCP Mode 15
2.3.2. Link Transmission Procedure 15
2.3.3. Frame Length 16
2.3.4. Address field of the link 16
2.4. Application Layer 17
2.4.1. MODBUS Function Codes 17
2.4.2. MODBUS Exception Status 18

3. Interoperability of SICAM RTUs using MODBUS/TCP Slave “Server” (MBSiA0) 19

3.1. Network Configurations 21
3.2. Physical layer 22
3.2.1. Electrical Interface 22
3.2.2. Transmission Speed 22
3.2.3. TCP-Port 22
3.2.4. Connections 22
3.3. Link Layer (MODBUS) 23
3.3.1. TCP Mode 23
3.3.2. Link Transmission Procedure 23
3.3.3. Frame Length 24
3.3.4. Address field of the link 24
3.4. Application Layer 25
3.4.1. MODBUS Function Codes 25
3.4.2. MODBUS Exception Status 26

A. MODBUS Data Formats 27

1. Introduction

Contents

1.1. Area of Application 8
1.2. General Information 9

1.1. Area of Application

In this documentation, all definitions are described that are necessary for communication between automation units or between automation and control room process computer systems as per MODBUS protocol specification.

1.2. General Information

Legend

£	Function is not supported
T	Function is supported
n	Function not defined for this application!

Definition

MASTER / SLAVE (Client / Server)

Protocol elements for communication with 3rd party systems using MODBUS protocol according definitions in this interoperability document.

	System Element	Note
MODBUS TCP/IP Client (=Master)		
MBCiA0	SM-2558/MBCiA0	SICAM AK 3, SICAM AK, SICAM TM, SICAM BC
MODBUS TCP/IP Server (=Slave)		
MBSiA0	SM-2558/MBSiA0	SICAM AK 3, SICAM AK, SICAM TM, SICAM BC

2. Interoperability of SICAM RTUs using MODBUS TCP/IP Master "Client" (MBCiA0)

Contents

2.1. Network Configurations 13
2.2. Physical layer 14
2.3. Link Layer (MODBUS) 15
2.4. Application Layer 17

The companion standard defined presents sets of parameters and alternatives from which subsets have to be selected to implement particular telecontrol systems. Certain parameter values, represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in Monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment stemming from different manufacturers it is necessary that all partners agree on the selected parameters.

The selected parameters should be crossed in the white boxes.



Note

In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

2.1. Network Configurations

	Configuration	Remark
T	LAN / WAN	

2.2. Physical layer

2.2.1. Electrical Interface

	Configuration	Remark
£	Ethernet (electrical)	
£	Ethernet (optical)	*)

*) External converter (electrical - optical) may be required!

2.2.2. Transmission Speed

	Speed	Remark
T	10 mbit/s	
T	100 mbit/s	
£	1000 mbit/s	

2.2.3. TCP-Port

	TCP-Port	Remark
T	502 *)	registered to MODBUS/TCP

*) TCP-Port fixed for MODBUS TCP/IP (not configurable)!

2.2.4. Connections

	Number of Connections	Remark
T	max. 100	

Note: each connection can be used as "SERVER" with individual data base, unique MODBUS Unit-ID Address and unique IP-address of the Client.

2.3. Link Layer (MODBUS)

	Transmission Mode
n	ASCII-Mode
n	RTU-Mode
T	TCP-Mode
£	UDP Mode

2.3.1. TCP Mode

	Byte Framing	Remark
T	8 Data Bits	

	Procedure	Remark
T	MODBUS Transaction Number (16 Bit)	MBAP Header
T	Protocol Identifier = "0" (16 Bit)	MBAP Header
T	Length (16 Bit)	MBAP Header
T	Unit Identifier (8 Bit)	MBAP Header
n	MODBUS Slave Address (8 Bit)	
T	MODBUS Function Code (8 Bit)	
T	MODBUS Register Address (16 Bit)	Address will address a 16 Bit MODBUS register
n	Cyclical Redundancy Check "CRC" (16 Bit)	CRC is not used by MODBUS/TCP (TCP-Checksum used only)

MBAP Header … MODBUS Application Protocol Header

2.3.2. Link Transmission Procedure

	Description	Remark
T	Unbalanced Master / Slave Communication	
T	MODBUS TCP/IP Master (=Client)	
£	MODBUS TCP/IP Slave (=Server)	

2.3.3. Frame Length

	Description	Remark
T	TCP-Mode: Maximum Message length 253 Bytes	

2.3.4. Address field of the link

	Description	Remark
T	1 octet (8 Bit) … Unit Identifier	Unit Identifier (1-247, 255) (MODBUS Slave Address 1-247)
£	BROADCAST Addressing Mode	

2.4. Application Layer

2.4.1. MODBUS Function Codes

	MODBUS Function Code - Description	Data Formats
	Data Access (Bit Access)	
T	01 = READ COILS	<0a> SPI, <0b> DPI
T	02 = READ DISCRETE INPUTS	<0a> SPI, <0b> DPI
T	05 = WRITE SINGLE COIL	<0a> SPI <0c> SC <0c> SC (pulse), <0d> DC2 (pulse)
T	15 = WRITE MULTIPLE COILS	<0a> SPI <0c> SC, <0d> DC <0c> SC (pulse), <0d> DC2 (pulse)
	Data Access (16 Bit Access)	
T	03 = READ HOLDING REGISTERS	<0a> SPI, <0b> DPI <20> SPI + IV, <21,22> DPI + IV <1> INT16, <2> UINT16 <23> INT16 + IV, <24> UINT16 + IV <3> INT32 (H/L), <4> UINT32 (H/L) <5> INT32 (L/H), <6> UINT32 (L/H) <7a> FLOAT32 <7b> FLOAT32 (swapped) <16> Bitstring 16 Bit
T	04 = READ INPUT REGISTERS	<0a> SPI, <0b> DPI <20> SPI + IV, <21,22> DPI + IV <1> INT16, <2> UINT16 <23> INT16 + IV, <24> UINT16 + IV <3> INT32 (H/L), <4> UINT32 (H/L) <5> INT32 (L/H), <6> UINT32 (L/H) <7a> FLOAT32 <7b> FLOAT32 (swapped) <16> Bitstring 16 Bit
T	06 = WRITE SINGLE REGISTER	<0a> SPI <0c> SC, <0d> DC <0c> SC (pulse), <0d> DC2 (pulse) <1> INT16, <2> UINT16 <16> Bitstring 16 Bit
T	16 = WRITE MULTIPLE REGISTERS	<0a> SPI <0c> SC, <0d> DC <0c> SC (pulse), <0d> DC2 (pulse) <1> INT16, <2> UINT16 <3> INT32 (H/L), <4> UINT32 (H/L) <5> INT32 (L/H), <6> UINT32 (L/H) <7a> FLOAT32 <7b> FLOAT32 (swapped) <16> Bitstring 16 Bit <1xx> DTx
£	22 = MASK WRITE REGISTER	
£	23 = READ / WRITE MULTIPLE REGISTERS	

	MODBUS Function Code - Description	Data Formats
£	24 = READ FIFO QUEUE	
	Data Access (File Record Access)	
£	20 = READ FILE RECORD	
£	21 = WRITE FILE RECORD	
	Diagnostics	
£	07 = READ EXCEPTION STATUS	
£	08 = DIAGNOSTICS (SUB-Code 00-18,20)	
£	11 = GET COM EVENT COUNTER	
£	12 = GET COM EVENT LOG	
£	17 = REPORT SLAVE ID	
£	43 = READ DEVCE Identification (SUB-Code = 14)	
	Other	
£	43 = Encapsulated Interface Transport (SUB-Code = 13,14)	

2.4.2. MODBUS Exception Status

	MODBUS Exception Code Description	Remark
T	01 = ILLEGAL FUNCTION	MODBUS Function Code not implemented
T	02 = ILLEGAL DATA ADDRESS	Requested Data Address not implemented
T	03 = ILLEGAL DATA VALUE	
£	04 = SERVER (Slave) DEVICE FAILURE	
£	05 = ACKNOWLEDGE	
£	06 = SERVER (Slave) DEVICE BUSY	
£	07 = NEGATIVE ACKNOWLEDGE (NAK)	
£	08 = MEMORY PARITY ERROR	
£	10 (0x0A) = GATEWAY PATH UNAVAILABLE	Retry by MODBUS TCP/IP Master for serial MODBUS Slaves – (connected via Gateway)
£	11 (0x0B) = GATEWAY TARGET DEVICE FAILED TO RESPOND	Retry by MODBUS TCP/IP Master for serial MODBUS Slaves – (connected via Gateway)

Note:
- Requested data answered by MODBUS TCP/IP Slave with exception code (except 10, 11) will be emulated SICAM RTUs internal with "NT=1" (not topical) – no retries on MODBUS TCP/IP level.

3. Interoperability of SICAM RTUs using MODBUS/TCP Slave “Server” (MBSiA0)

Contents

3.1. Network Configurations 21
3.2. Physical layer 22
3.3. Link Layer (MODBUS) 23
3.4. Application Layer 25

The companion standard defined presents sets of parameters and alternatives from which subsets have to be selected to implement particular telecontrol systems. Certain parameter values, represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in Monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment stemming from different manufacturers it is necessary that all partners agree on the selected parameters.

The selected parameters should be crossed in the white boxes.



Note

In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

3.1. Network Configurations

	Configuration	Remark
T	LAN / WAN	

3.2. Physical layer

3.2.1. Electrical Interface

	Configuration	Remark
T	Ethernet (electrical)	
T	Ethernet (optical)	*)

*) External converter (electrical - optical) may be required!

3.2.2. Transmission Speed

	Speed	Remark
T	10 mbit/s	
T	100 mbit/s	
£	1000 mbit/s	

3.2.3. TCP-Port

	TCP-Port	Remark
T	502 *)	registered to MODBUS/TCP

*) TCP-Port fixed for MODBUS TCP/IP (not configurable)!

3.2.4. Connections

	Number of Connections	Remark
T	max. 100	

Note: each connection can be used as "SERVER" with individual data base, unique MODBUS Unit-ID Address and unique IP-address of the Client.

3.3. Link Layer (MODBUS)

	Transmission Mode
n	ASCII-Mode
n	RTU-Mode
T	TCP-Mode
£	UDP Mode

3.3.1. TCP Mode

	Byte Framing	Remark
T	8 Data Bits	

	Procedure	Remark
T	MODBUS Transaction Number (16 Bit)	MBAP Header
T	Protocol Identifier = "0" (16 Bit)	MBAP Header
T	Length (16 Bit)	MBAP Header
T	Unit Identifier (8 Bit)	MBAP Header
n	MODBUS Slave Address (8 Bit)	
T	MODBUS Function Code (8 Bit)	
T	MODBUS Register Address (16 Bit)	Address will address a 16 Bit MODBUS register
n	Cyclical Redundancy Check "CRC" (16 Bit)	CRC is not used by MODBUS/TCP (TCP-Checksum used only)

MBAP Header … MODBUS Application Protocol Header

3.3.2. Link Transmission Procedure

	Description	Remark
T	Unbalanced Master / Slave Communication	
£	MODBUS TCP/IP Master (=Client)	
T	MODBUS TCP/IP Slave (=Server)	

3.3.3. Frame Length

	Description	Remark
T	TCP-Mode: Maximum Message length 253 Bytes	

3.3.4. Address field of the link

	Description	Remark
T	1 octet (8 Bit) … Unit Identifier	Unit Identifier (MODBUS Slave Address 1-247)
£	BROADCAST Addressing Mode	

3.4. Application Layer

3.4.1. MODBUS Function Codes

	MODBUS Function Code - Description	Data Formats
	Data Access (Bit Access)	
T	01 = READ COILS	<0a> SPI, <0b> DPI
T	02 = READ DISCRETE INPUTS	<0a> SPI, <0b> DPI
T	05 = WRITE SINGLE COIL	<0a> SPI <0c> SC <0c> SC (pulse), <0d> DC2 (pulse)
T	15 = WRITE MULTIPLE COILS	<0a> SPI, <0b> DPI <0c> SC, <0d> DC <0c> SC (pulse), <0d> DC2 (pulse)
	Data Access (16 Bit Access)	
T	03 = READ HOLDING REGISTERS	<0a> SPI, <0b> DPI <20> SPI + IV, <21,22> DPI + IV <1> INT16, <2> UINT16 <23> INT16 + IV, <24> UINT16 + IV <3> INT32 (H/L), <4> UINT32 (H/L) <5> INT32 (L/H), <6> UINT32 (L/H) <7a> FLOAT32 <7b> FLOAT32 (swapped) <16> Bitstring 16 Bit
T	04 = READ INPUT REGISTERS	<0a> SPI, <0b> DPI <20> SPI + IV, <21,22> DPI + IV <1> INT16, <2> UINT16 <23> INT16 + IV, <24> UINT16 + IV <3> INT32 (H/L), <4> UINT32 (H/L) <5> INT32 (L/H), <6> UINT32 (L/H) <7a> FLOAT32 <7b> FLOAT32 (swapped) <16> Bitstring 16 Bit
T	06 = WRITE SINGLE REGISTER	<0a> SPI, <0b> DPI <0c> SC, <0d> DC <0c> SC (pulse), <0d> DC2 (pulse) <1> INT16, <2> UINT16
T	16 = WRITE MULTIPLE REGISTERS	<0a> SPI, <0b> DPI <0c> SC, <0d> DC <0c> SC (pulse), <0d> DC2 (pulse) <1> INT16, <2> UINT16 <3> INT32 (H/L), <4> UINT32 (H/L) <5> INT32 (L/H), <6> UINT32 (L/H) <7a> FLOAT32 <7b> FLOAT32 (swapped)
£	22 = MASK WRITE REGISTER	
£	23 = READ / WRITE MULTIPLE REGISTERS	
£	24 = READ FIFO QUEUE	

	MODBUS Function Code - Description	Data Formats
	Data Access (File Record Access)	
£	20 = READ FILE RECORD	
£	21 = WRITE FILE RECORD	
	Diagnostics	
£	07 = READ EXCEPTION STATUS	
£	08 = DIAGNOSTICS (SUB-Code 00-18,20)	
£	11 = GET COM EVENT COUNTER	
£	12 = GET COM EVENT LOG	
£	17 = REPORT SLAVE ID	
£	43 = READ DEVCE Identification (SUB-Code = 14)	
	Other	
n/i	43 = Encapsulated Interface Transport (SUB-Code = 13,14)	

3.4.2. MODBUS Exception Status

	MODBUS Exception Code Description	Remark
T	01 = ILLEGAL FUNCTION	MODBUS Function Code not implemented
T	02 = ILLEGAL DATA ADDRESS	Requested data address not implemented (not available) or MODBUS data formats requires more registers/coils not fully requested (e.g. FLOAT32, INT32, DP,…).
T	03 = ILLEGAL DATA VALUE	
T	04 = SLAVE DEVICE FAILURE	If Slave internal status of requested data is “IV=1“, then send: - old-/replacement value or - exception code=4 (parameterizable)
£	05 = ACKNOWLEDGE	
£	06 = SLAVE DEVICE BUSY	
£	07 = NEGATIVE ACKNOWLEDGE (NAK)	
£	08 = MEMORY PARITY ERROR	
£	10 (0x0A) = GATEWAY PATH UNAVAILABLE	
£	11 (0x0B) = GATEWAY TARGET DEVICE FAILED TO RESPOND	

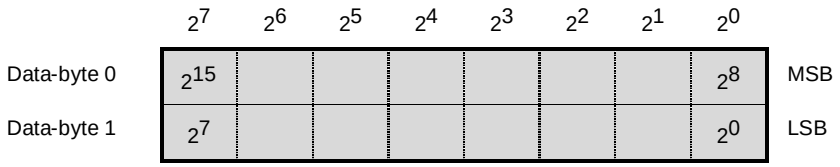
A. MODBUS Data Formats

Supported MODBUS Data Formats:

Format #	Format	Meaning	Coil	Register
General Formats				
1	INT16	Signed Integer 16 Bit		ü
2	UINT16	Unsigned Integer 16 Bit		ü
3	INT32 (H/L)	Signed Integer 32 Bit (“HIGH before LOW“)		ü
4	UINT32 (H/L)	Unsigned Integer 32 Bit (“HIGH before LOW“)		ü
5	INT32 (L/H)	Signed Integer 32 Bit (“LOW before HIGH“)		ü
6	UINT32 (L/H)	Unsigned Integer 32 Bit (“LOW before HIGH“)		ü
7a	FLOAT32	Short Floating Point (IEEE 754)		ü
7b	FLOAT32 (swapped)	Short Floating Point (IEEE 754) “swapped“		ü
16	Bitstring 16 Bit	Bitstring of 16 Bit		ü
20	SPI	Single Point Information	ü	ü
21	DPI (1=off, 2=on)	Double Point Information (OFF before ON)	ü	ü
22	DPI (1=on, 2=off)	Double Point Information (ON before OFF)	ü	ü
30	SC	Single Command	ü	ü
31	SC (pulse)	Single Command “pulse“	ü	ü
32	DC	Double Command	ü	ü
35	DC2 (pulse)	Double Command “pulse“	ü	ü
Device Specific Formats				
50	SPI + IV	Single Point Information + "Invalid-Bit"		ü
51	DPI (1=off, 2=on) + IV	Double Point Information (OFF before ON) + "Invalid-Bit"		ü
52	DPI (1=on, 2=off) + IV	Double Point Information (ON before OFF) + "Invalid-Bit"		ü
53	INT16 + IV	Signed Integer 16 Bit + "Invalid-Bit"		ü
54	UINT16 + IV	Unsigned Integer 16 Bit + "Invalid-Bit"		ü
1xx	DTx	Date & Time (free configurable)		ü

Data formats in a MODBUS register will be represented/sent always in “Big Endian" HIGH before LOW order.

Data in MODBUS Register:



Most significant byte (MSB) is sent first!

Format-1: INT16 – Signed Integer 16 Bit

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	VZ	2^{14}						2^8
Data-byte 1	2^7							2^0

VZ: "sign" (0="+", 1="-")
Range of value: -32768...0...+32767
Note: negative values will be stored in two's complement.

Format-2: UINT16 – Unsigned Integer 16 Bit

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0

Range of value: 0 ... 65535

Format-3: INT32 (H/L) – Signed Integer 32 Bit (“HIGH before LOW“)

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	VZ	2^{30}						2^{24}
Data-byte 1	2^{23}							2^{16}
Data-byte 2	2^{15}							2^8
Data-byte 3	2^7							2^0

VZ: "sign" (0="+", 1="-")
Range of value: -2 147 483 648...0...+2 147 483 647
Note: negative values will be stored in two's complement.

Format-4: UINT32 (H/L) – Unsigned Integer 32 Bit (“HIGH before LOW“)

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{31}							2^{24}
Data-byte 1	2^{23}							2^{16}
Data-byte 2	2^{15}							2^8
Data-byte 3	2^7							2^0

Range of value: 0 ... 4 294 967 295

Format-5: INT32 (L/H) – Signed Integer 32 Bit (“LOW before HIGH“)

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0
Data-byte 2	VZ	2^{30}						2^{24}
Data-byte 3	2^{23}							2^{16}

VZ: "sign" (0="+", 1="-")
Range of value: -2 147 483 648...0...+2 147 483 647
Note: negative values will be stored in two's complement.

Format-6: UINT32 (L/H) – Unsigned Integer 32 Bit (“LOW before HIGH“)

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0
Data-byte 2	2^{31}							2^{24}
Data-byte 3	2^{23}							2^{16}

Range of value: 0...4 294 967 295

Format-7: FLOAT32 – Short Floating Point (IEEE754)

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Data-byte 0	2^{15}							2^8	mantissa
Data-byte 1	2^7							2^0	mantissa
Data-byte 2	S	2^7						2^1	exponent, S (=sign)
Data-byte 3	2^0	2^{22}						2^{16}	mantissa

Range of value: $-2^{128} + 2^{104} \ldots +2^{128} - 2^{104}$
$-3{,}4.10^{38} \ldots +3{,}4.10^{38}$
Smallest negative number: $-2^{-149} = -1{,}4.10^{-45}$
Smallest positive number: $+2^{-149} = +1{,}4.10^{-45}$
S (sign): <0> = "+"; <1> = "-"

Format-7b: FLOAT32 (swapped) – Short Floating Point (IEEE754) "swapped"

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Data-byte 0	S	2^7						2^1	exponent, S (=sign)
Data-byte 1	2^0	2^{22}						2^{16}	mantissa
Data-byte 2	2^{15}							2^8	mantissa
Data-byte 3	2^7							2^0	mantissa

Range of value: $-2^{128} + 2^{104} \ldots +2^{128} - 2^{104}$
$-3{,}4.10^{38} \ldots +3{,}4.10^{38}$
Smallest negative number: $-2^{-149} = -1{,}4.10^{-45}$
Smallest positive number: $+2^{-149} = +1{,}4.10^{-45}$
S (sign): <0> = "+"; <1> = "-"

Format-16: Bitstring 16 Bit – Bitstring 16 Bit

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0

Format-20: SPI – Single-Point Information

Single-point information (bit) in MODBUS register or coil.

	N
Coil	SPI

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0

Data-byte 0	SPI_{15}	SPI_{14}	SPI_{13}	SPI_{12}	SPI_{11}	SPI_{10}	SPI_9	SPI_8
Data-byte 1	SPI_7	SPI_6	SPI_5	SPI_4	SPI_3	SPI_2	SPI_1	SPI_0

Note: MODBUS Master firmware supports only 1x SPI per MODBUS register with function code FC=6,16 in transmit direction!

Range of value: 0, 1

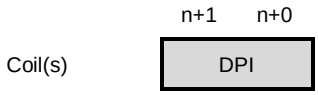
SPI – Single-Point Information
<0> = OFF
<1> = ON

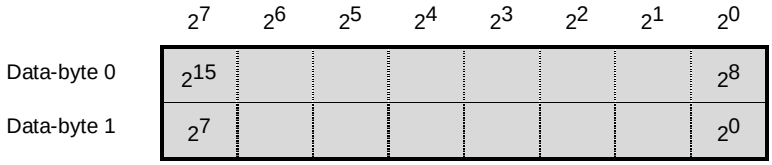
	SPI	Coding (IEC60870-5-101/-104)
0	**0**	**OFF**
1	**1**	**ON**

Format-21: DPI (1=off, 2=on) – Double-Point Information (OFF before ON)

Format-22: DPI (1=off, 2=on) – Double-Point Information (ON before OFF)

Double-point information will be sent in 2 neighboring bits in MODBUS register or as coils.
Note:
The 2 bits of DPI must be located always in same byte of a MODBUS register.





Data-byte 0	DPI_7	DPI_6	DPI_5	DPI_4
Data-byte 1	DPI_3	DPI_2	DPI_1	DPI0

Note: MODBUS Master firmware supports only 1x DPI per MODBUS register with function code FC=6,16 in transmit direction!

Range of value: 0…3

DPI – Double-Point Information (OFF before ON) - DPI (1=off, 2=on)
<0> = indeterminate or intermediate state
<1> = OFF
<2> = ON
<3> = indeterminate state

DPI – Double-Point Information (ON before OFF) - DPI (1=on, 2=off)
<0> = indeterminate or intermediate state
<1> = ON
<2> = OFF
<3> = indeterminate state

	Bit (n+1)	Bit (n+0)	Coding "OFF before ON" (IEC60870-5-101/-104)	Coding "ON before OFF"
0	0	0	DIFF	DIFF
1	**0**	**1**	**OFF**	**ON**
2	**1**	**0**	**ON**	**OFF**
3	1	1	STÖR	STÖR

DIFF……Indeterminate or intermediate state
STÖR……Indeterminate state

Format-30: SC – Single Command

A single command with command state SCS=<ON> or SCS=<OFF> will be sent as single coil or as single bit in MODBUS register.

	n
Coil	SCS

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0

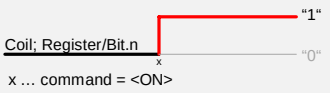
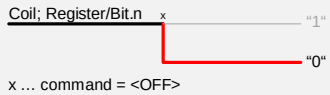
Data-byte 0	SCS_{15}	SCS_{14}	SCS_{13}	SCS_{12}	SCS_{11}	SCS_{10}	SCS_9	SCS_8
Data-byte 1	SCS_7	SCS_6	SCS_5	SCS_4	SCS_3	SCS_2	SCS_1	SCS_0

Note: MODBUS Master firmware supports only 1x SC per MODBUS register with function code FC=6,16 in transmit direction!

Range of value: 0, 1

SCS - Single Command
<0> = OFF
<1> = ON

Command	MODBUS Format	Command Transmission
Single Command “ON“	SC	COIL (n); Register/Bit (n) = “ON“
Single Command “OFF“	SC	COIL (n); Register/Bit (n) = “OFF“

MODBUS Format	Command State	Command Output SC 1 as Coil or 1 Bit in MODBUS Register
SC	SCS=ON	
	SCS=OFF	

Format-31: SC (Pulse) – Single Command (Pulse)

A single command with command state SCS=<ON> will be sent as pulse with pulse length of configured command output time as single coil or as single bit in MODBUS register.
The command state <OFF> is not evaluated.
The command state will be set to "inactive" after command output time.

	n
Coil	SCP

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0

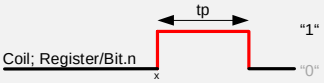
Data-byte 0	SCP_{15}	SCP_{14}	SCP_{13}	SCP_{12}	SCP_{11}	SCP_{10}	SCP_9	SCP_8
Data-byte 1	SCP_7	SCP_6	SCP_5	SCP_4	SCP_3	SCP_2	SCP_1	SCP_0

Note: MODBUS Master firmware supports only 1x SC (pulse) per MODBUS register with function code FC=6,16 in transmit direction!

Range of value: 0, 1

SCP - Single Command (Pulse)
<0> = inactive
<1> = active

Command	MODBUS Format	Command Transmission
Single Command “ON“	SC (pulse)	COIL (n); Register/Bit (n) = Pulse

MODBUS Format	Command State	Command Output SC 1 as Coil or 1 Bit in MODBUS Register
SC (pulse)	SCS=ON	 tp … command output time (pulse duration) x …. command = <ON>
	SCS=OFF	The <OFF> state is not evaluated!

Format-32: DC – Double Command (2 Bit) 2 BIT

The command state of a double command or regulating step command (2 bit) is sent as 2 neighboring bits as coils or bits in MODBUS register.

	n+1	n+0
Coil(s)	DCS	

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0

Data-byte 0	DC_7	DC_6	DC_5	DC_4
Data-byte 1	DC_3	DC_2	DC_1	DC0

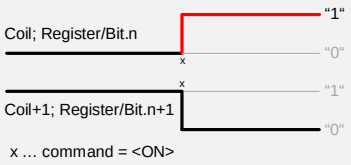
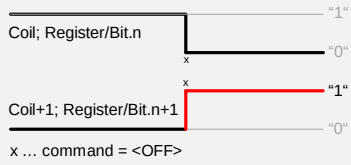
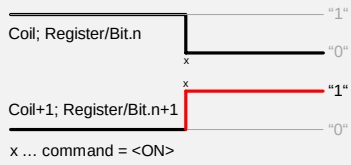
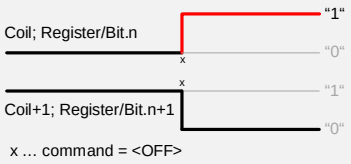
Note: MODBUS Master firmware supports only 1x DC per MODBUS register with function code FC=6,16 in transmit direction!

Range of value: 0…3

DCS - Double Command [MODBUS_command_state = OFF]
<1> = OFF
<2> = ON

DCS - Double Command [MODBUS_command_state = ON]
<1> = ON
<2> = OFF

	Bit (n+1)	Bit (n+0)	Coding "OFF before ON" (IEC60870-5-101/-104) [MODBUS_command_state = OFF]	Coding "ON before OFF" [MODBUS_command_state = ON]
0	0	0	not permitted	not permitted
1	**0**	**1**	**OFF**	**ON**
2	**1**	**0**	**ON**	**OFF**
3	1	1	not permitted	not permitted

MODBUS Format	Command State	**Command Output** DC as 2 Bit Coil or 2 Bit in MODBUS-Register
DC MODBUS_Command_State = ON	DCS=ON RCS=Higher	
	DCS=OFF RCS=Lower	
DC MODBUS_Command_State = OFF	DCS=ON RCS=Higher	
	DCS=OFF RCS=Lower	

Format-33: DC1 – Double Command (1 Bit)

A double command with command state SCS=<ON> and <OFF> will be sent as Coil (1 bit) or as 1 bit in MODBUS register.

	n
Coil	DCS

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0

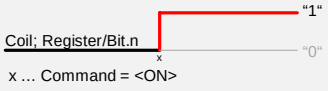
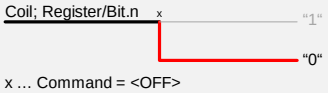
Data-byte 0	DCS_{15}	DCS_{14}	DCS_{13}	DCS_{12}	DCS_{11}	DCS_{10}	DCS_9	DCS_8
Data-byte 1	DCS_7	DCS_6	DCS_5	DCS_4	DCS_3	DCS_2	DCS_1	DCS_0

Note: MODBUS Master firmware supports only 1x DC (1 Bit) per MODBUS register with function code FC=6,16 in transmit direction!

Range of value: 0, 1

DCS – Double Command State (1 Bit)
<0> = OFF
<1> = ON

Command	MODBUS Format	Command Transmission
Double Command "ON"	DC1	COIL (n); Register/Bit (n) = "ON"
Double Command "OFF"	DC1	COIL (n); Register/Bit (n) = "OFF"

MODBUS Format	Command State	Command Output DC as Coil or 1 Bit in MODBUS Register
DC1	DCS=ON	
	DCS=OFF	

Format-34 DC2 (pulse) – Double Command (Pulse 2 Bit) 2 Bit

A double command or regulating step command with command state DCS=<ON>/<OFF> or RCS=<Higher>/<Lower> will be transmitted on MODBUS as pulse (2 bit) with configured command output time on 2 coils or 2 bits in MODBUS register.
The command state <ON> or <Higher> will be transmitted as 1 bit pulse at coil address (n+0 or n+1) or at register address/bit (n+0 or n+1).
The command state <OFF> or <Lower> will be transmitted as 1 bit pulse at coil address (n+1 or n+0) or at register address/bit (n+1 or n+0).
The command state will be set to "inactive" after command output time.

	n+1	n+0
Coil(s)	DCP	

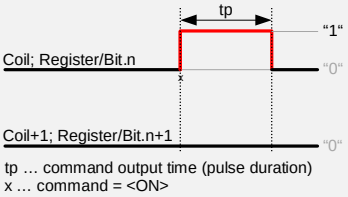
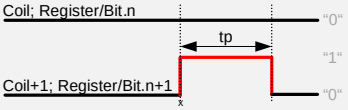
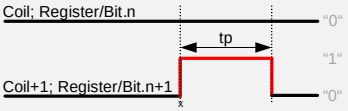
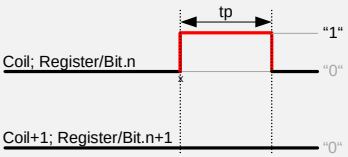
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	2^{15}							2^8
Data-byte 1	2^7							2^0

Data-byte 0	DCP_7	DCP_6	DCP_5	DCP_4
Data-byte 1	DCP_3	DCP_2	DCP_1	DCP0

Note: MODBUS Master firmware supports only 1x DC2 (pulse) per MODBUS register with function code FC=6,16 in transmit direction!

Range of value: 0…3

DCP – Double Command (Pulse) – each bit
<0> = inactive
<1> = active

MODBUS Format	Command State	**Command Output** DC as 2 Bit Coil or 2 Bit in MODBUS-Register
DC2 (pulse) MODBUS_Command_State = ON	DCS=ON RCS=Higher	 tp … command output time (pulse duration) x … command = <ON>
DC2 (pulse) MODBUS_Command_State = ON	DCS=OFF RCS=Lower	 tp … command output time (pulse duration) x … command = <OFF>
DC2 (pulse) MODBUS_Command_State = OFF	DCS=ON RCS=Higher	 tp … command output time (pulse duration) x … command = <ON>
DC2 (pulse) MODBUS_Command_State = OFF	DCS=OFF RCS=Lower	 tp … command output time (pulse duration) x … command = <OFF>

Format 50: SPI + IV – Single Point Information + “Invalid-Bit“

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	IV	SPI_{14}	SPI_{13}	SPI_{12}	SPI_{11}	SPI_{10}	SPI_{9}	SPI_{8}
Data-byte 1	SPI_{7}	SPI_{6}	SPI_{05}	SPI_{04}	SPI_{03}	SPI_{02}	SPI_{01}	SPI_{00}

IV ("Invalid-Bit"):
<0> = valid
<1> = invalid

SPI (“Single Point Information“):
<0> = OFF
<1> = ON

Note:
- IV-Bit is valid for all bits in MODBUS register.
- not used SPI/bits will be sent with state <0>.
- Single-/double point information can be mixed in one MODBUS register.

Format 51: DPI (1=off, 2=on) + IV
Double Point Information (OFF before ON) + “Invalid-Bit“

Format 52: DPI (1=on, 2=off) + IV
Double Point Information (ON before OFF) + “Invalid-Bit“

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	IV		DPI_6		DPI_5		DPI_4	
Data-byte 1	DPI_3		DPI_2		DPI_1		DPI0	

IV ("Invalid-Bit"):
<0> = valid
<1> = invalid

DPI “Double Point Information“ ("OFF before ON"): "DPI (1=off, 2=on)"
<0> = indeterminate or intermediate state
<1> = OFF
<2> = ON
<3> = indeterminate state

DPI “Double Point Information“ (ON before OFF): "DPI (1=on, 2=off)"
<0> = indeterminate or intermediate state
<1> = ON
<2> = OFF
<3> = indeterminate state

	Bit (n+1)	**Bit** (n+0)	**Coding “OFF before ON“ (IEC60870-5-101/-104)**	**Coding “ON before OFF“**
0	0	0	DIFF	DIFF
1	**0**	**1**	**OFF**	**ON**
2	**1**	**0**	**ON**	**OFF**
3	1	1	STÖR	STÖR

DIFF……Indeterminate or intermediate state
STÖR……Indeterminate state

Note:
- IV-Bit is valid for all bits in MODBUS register.
- not used DPI/bits will be sent with state <0>.
- Single-/double point information can be mixed in one MODBUS register.
- Bits of double point information must be located always in same byte of MODBUS register.

Format 53: INT16 + IV – Signed Integer 16 Bit + “Invalid-Bit“

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	IV	S	2^{13}					2^8
Data-byte 1	2^7							2^0

IV ("Invalid-Bit"): <0> = valid
<1> = invalid
S ("Sign"): <0> = +
<1> = -
Range of value: -16384...0...+16383
Note: negative values will be stored in two's complement.

Format 54: UINT16 + IV – Unsigned Integer 16 Bit + “Invalid-Bit“

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Data-byte 0	IV	2^{14}						2^8
Data-byte 1	2^7							2^0

IV ("Invalid-Bit"): <0> = valid
<1> = invalid
Range of value: 0...32767

Format-1xx: DTx – Date and Time (free configurable)

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Data-byte 0	2^{15}							2^8	HIGH – Time-Element
Data-byte 1	2^7							2^0	LOW – Time-Element
Data-byte 2	2^{15}							2^8	HIGH – Time-Element
Data-byte 3	2^7							2^0	LOW – Time-Element
Data-byte 4	2^{15}							2^8	HIGH – Time-Element
Data-byte 5	2^7							2^0	LOW – Time-Element
Data-byte 6	2^{15}							2^8	HIGH – Time-Element
Data-byte 7	2^7							2^0	LOW – Time-Element
Data-byte 8	2^{15}							2^8	HIGH – Time-Element
Data-byte 9	2^7							2^0	LOW – Time-Element
Data-byte 10	2^{15}							2^8	HIGH – Time-Element
Data-byte 11	2^7							2^0	LOW – Time-Element

Note:
- only the configured number of MODBUS Registers bytes will be sent.
  If "End of Frame" is assigned to "HIGH" data-byte, MODBUS register will not be sent.
  If "End of Frame" is assigned to "LOW" data-byte, MODBUS Register will be sent with "0" in "LOW" data-byte.

Byte sending order:
Data-byte 0 (MSB of 1st MODBUS register) is sent 1st.
Data-byte 1 (LSB of 1st MODBUS register) is sent 2nd.
Data-byte 2 (MSB of 2nd MODBUS register) is sent 3rd.
:
:
Data-byte n will be sent as last byte.

Time Element	Range of Value	Example	MBMiA0	MBSiA0
Not used	Dummy: UI8[7..0] <0>		ü	
year (high)	year (high) [7..0] <0..255>	year = 2016 = 07E0 [HEX] à year (high) = 0x07 [HEX]	ü	
year (low)	year (low) [7..0] <0..255>	year = 2016 = 07E0 [HEX] à year (low) = 0xE0 [HEX]	ü	
year - 2000 (high)	year - 2000 (high) [7..0] <0..255>	year = 2016 à 2016 - 2000 = 16 = 0x0010 [HEX] à year - 2000 (high) = 0x00 [HEX]	ü	
year - 2000 (low)	year - 2000 (low) [7..0] <0..255>	year = 2016 à 2016 - 2000 = 16 = 0x0010 [HEX] à year - 2000 (low) = 0x10 [HEX]	ü	
month	month [7..0] <1..12>	month = 12 (December) à month = 0x0C [HEX]	ü	
day	Day of month [7..0] <1..31>	day = 23 à day = 0x17 [HEX]	ü	
day of week	day of week [7..0] <1..7> <1>:= Monday; < 2>:= Tuesday, ...<7>:= Sunday	Day of week = Tuesday à day of week = 0x02 [HEX]	ü	
day + day of week	day [4..0] <1..31> day of week [7..5] <1..7> <1>:= Monday; < 2>:= Tuesday, ...<7>:= Sunday		ü	
hour	hour [7..0] <0..23>	hour = 21 = 0x15 [HEX]	ü	
hour + SU	hour [4..0] <0..23> summer time (SU) [7] <0,1> SU <0>:= standard time (winter time) SU <1>:= summer time		ü	
minute	minute [5..0] <0..59>	minute = 59 = 0x3B [HEX]	ü	
minute + IV	minute [5..0] <0..59> Invalid (IV) [7] <0,1> IV <0>:= valid IV <1>:= invalid		ü	
second	Second [BIN] [7..0] <0..59>	second = 32 = 0x20 [HEX]	ü	
milliseconds (high)	milliseconds n*1ms (high) [7..0] <0... 255> n <0 - 59999> = range including seconds	milliseconds = 998 = 03E6 [HEX] à milliseconds (high) = 0x03 [HEX]	ü	
milliseconds (low)	milliseconds n*1ms (low) [7..0] <0..255> n <0 - 59999> = range including seconds	milliseconds = 998 = 03E6 [HEX] à milliseconds (low) = 0xE6 [HEX]	ü	
ticks (10ms)	milliseconds n*10ms [7..0] <0..99>	milliseconds = 998 = à ticks (10ms) = 99 = 0x63 [HEX]	ü	
ticks (100ms)	milliseconds n*100ms [7..0] <0..9>	milliseconds = 998 = à ticks (100ms) = 9 = 0x09 [HEX]	ü	
EOF (end of frame)		Note: This time element is an indicator for end of time record	ü	

Legend: [7..0] = bit position in data byte of MODBUS register
< > = valid range of value

Literature

SICAM RTUs . Ax 1703 Common Functions MODBUS	DC0-088-2
SICAM RTUs Common Functions System- and Basic System Elements	DC0-015-2

International Standards

Modicon MODBUS Protocol Reference Guide	PI-MBUS-300 (Rev. J)
Modbus Application Protocol Specification, V 1.1b	
Modbus Messaging on TCP Implementation Guide, Rev. 1.0b	