

## **MODBUS interface description**

# Table of Contents

<b>General description.....</b>	<b>1</b>
<u>Introduction.....</u>	1
<u>Why a MODBUS Server on the MSX-E modules?.....</u>	1
<u>Technical details.....</u>	1
<b>FC3 (read multiple register) Functions.....</b>	<b>3</b>
<u>Function GetLastCommandStatus.....</u>	4
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>GetLastCommandStatusEx.....</u>	4
<u>Description.....</u>	4
<u>Query frame layout.....</u>	4
<u>Response frame layout.....</u>	5
<u>Exception frame layout.....</u>	5
<u>Function GetLastCommandStatusEx.....</u>	6
<u>Description.....</u>	6
<u>Query frame layout.....</u>	6
<u>Response frame layout.....</u>	7
<u>Exception frame layout.....</u>	7
<u>Function MXCommon GetModuleType.....</u>	8
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon GetModuleTypeEx.....</u>	8
<u>Description.....</u>	8
<u>Query frame layout.....</u>	8
<u>Response frame layout.....</u>	8
<u>Exception frame layout.....</u>	9
<u>Function MXCommon GetModuleTypeEx.....</u>	9
<u>Description.....</u>	9
<u>Query frame layout.....</u>	10
<u>Response frame layout.....</u>	10
<u>Exception frame layout.....</u>	11
<u>Function MXCommon GetTime.....</u>	11
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon GetTimeEx.....</u>	11
<u>Description.....</u>	11
<u>Query frame layout.....</u>	11
<u>Response frame layout.....</u>	12
<u>Exception frame layout.....</u>	13
<u>Function MXCommon GetTimeEx.....</u>	13
<u>Description.....</u>	13
<u>Query frame layout.....</u>	13
<u>Response frame layout.....</u>	14
<u>Exception frame layout.....</u>	14
<u>Function MXCommon TestCustomerID.....</u>	15
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon TestCustomerIDEx.....</u>	15
<u>Description.....</u>	15
<u>Query frame layout.....</u>	15
<u>Response frame layout.....</u>	16

# Table of Contents

## **FC3 (read multiple register) Functions**

Exception frame layout.....	16
Function <u>MXCommon TestCustomerIDEx</u> .....	17
Description.....	17
Query frame layout.....	17
Response frame layout.....	18
Exception frame layout.....	18
Function <u>MSXE17xx MFSinCosReadAll</u> .....	19
Description.....	19
Query frame layout.....	20
Response frame layout.....	20
Exception frame layout.....	21
Function <u>MSXE17xx MFSinCosRead0</u> .....	21
Description.....	21
Query frame layout.....	22
Response frame layout.....	22
Exception frame layout.....	23
Function <u>MSXE17xx MFSinCosRead1</u> .....	23
Description.....	23
Query frame layout.....	24
Response frame layout.....	24
Exception frame layout.....	25
Function <u>MSXE17xx MFSinCosRead2</u> .....	25
Description.....	25
Query frame layout.....	26
Response frame layout.....	26
Exception frame layout.....	27
Function <u>MSXE17xx MFSinCosRead3</u> .....	27
Description.....	27
Query frame layout.....	28
Response frame layout.....	28
Exception frame layout.....	29
Function <u>MSXE17xx MFSinCosReadAbsPos0</u> .....	29
Description.....	30
Query frame layout.....	30
Response frame layout.....	31
Exception frame layout.....	31
Function <u>MSXE17xx MFSinCosReadAbsPos1</u> .....	32
Description.....	32
Query frame layout.....	33
Response frame layout.....	33
Exception frame layout.....	34
Function <u>MSXE17xx MFSinCosReadAbsPos2</u> .....	34
Description.....	34
Query frame layout.....	35
Response frame layout.....	35
Exception frame layout.....	36
Function <u>MSXE17xx MFSinCosReadAbsPos3</u> .....	37

# Table of Contents

## **FC3 (read multiple register) Functions**

<u>Description</u> .....	37
<u>Query frame layout</u> .....	37
<u>Response frame layout</u> .....	38
<u>Exception frame layout</u> .....	38
<b><u>Function MSXE17xx MFSinCosReadAbsPosAll</u></b> .....	39
<u>Description</u> .....	39
<u>Query frame layout</u> .....	40
<u>Response frame layout</u> .....	41
<u>Exception frame layout</u> .....	42
<b><u>Function MSXE17xx DigitalIOReadAllChannelsValue</u></b> .....	42
<u>Description</u> .....	42
<u>Query frame layout</u> .....	42
<u>Response frame layout</u> .....	43
<u>Exception frame layout</u> .....	43
<b><u>Function MSXE17xx DigitalIOTestShortCircuit</u></b> .....	44
<u>Description</u> .....	44
<u>Query frame layout</u> .....	44
<u>Response frame layout</u> .....	45
<u>Exception frame layout</u> .....	45
<b><u>Function MSXE17xx IOWatchdogGetStatusAndValue</u></b> .....	46
<u>Description</u> .....	46
<u>Query frame layout</u> .....	46
<u>Response frame layout</u> .....	47
<u>Exception frame layout</u> .....	47

## **FC16 (write multiple register) Functions**.....49

<b><u>Function MXCommon SetHardwareTriggerFilterTime</u></b> .....	50
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon SetHardwareTriggerFilterTimeEx</u> .....	50
<u>Description</u> .....	50
<u>Query frame layout</u> .....	51
<u>Response frame layout</u> .....	52
<u>Exception frame layout</u> .....	52
<b><u>Function MXCommon SetHardwareTriggerFilterTimeEx</u></b> .....	53
<u>Description</u> .....	53
<u>Query frame layout</u> .....	53
<u>Response frame layout</u> .....	54
<u>Exception frame layout</u> .....	54
<b><u>Function MXCommon InitAndStartSynchroTimer</u></b> .....	55
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon InitAndStartSynchroTimerEx</u> .....	55
<u>Description</u> .....	55
<u>Query frame layout</u> .....	56
<u>Response frame layout</u> .....	57
<u>Exception frame layout</u> .....	57
<b><u>Function MXCommon InitAndStartSynchroTimerEx</u></b> .....	58
<u>Description</u> .....	58

# Table of Contents

## **FC16 (write multiple register) Functions**

<u>Query frame layout</u> .....	59
<u>Response frame layout</u> .....	60
<u>Exception frame layout</u> .....	60
<u>Function MXCommon StopAndReleaseSynchroTimer</u> .....	61
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon StopAndReleaseSynchroTimerEx</u> .....	61
<u>Description</u> .....	61
<u>Query frame layout</u> .....	61
<u>Response frame layout</u> .....	62
<u>Exception frame layout</u> .....	62
<u>Function MXCommon StopAndReleaseSynchroTimerEx</u> .....	63
<u>Description</u> .....	63
<u>Query frame layout</u> .....	63
<u>Response frame layout</u> .....	64
<u>Exception frame layout</u> .....	64
<u>Function MXCommon Reboot</u> .....	65
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon RebootEx</u> .....	65
<u>Description</u> .....	65
<u>Query frame layout</u> .....	65
<u>Response frame layout</u> .....	66
<u>Exception frame layout</u> .....	66
<u>Function MXCommon RebootEx</u> .....	67
<u>Description</u> .....	67
<u>Query frame layout</u> .....	67
<u>Response frame layout</u> .....	68
<u>Exception frame layout</u> .....	68
<u>Function MXCommon SetCustomerKey</u> .....	69
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon SetCustomerKeyEx</u> .....	69
<u>Description</u> .....	69
<u>Query frame layout</u> .....	69
<u>Response frame layout</u> .....	70
<u>Exception frame layout</u> .....	71
<u>Function MXCommon SetCustomerKeyEx</u> .....	71
<u>Description</u> .....	71
<u>Query frame layout</u> .....	71
<u>Response frame layout</u> .....	72
<u>Exception frame layout</u> .....	73
<u>Function MXCommon SetFilterChannels</u> .....	73
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon SetFilterChannelsEx</u> .....	73
<u>Description</u> .....	73
<u>Query frame layout</u> .....	74
<u>Response frame layout</u> .....	74
<u>Exception frame layout</u> .....	75
<u>Function MXCommon SetFilterChannelsEx</u> .....	75

# Table of Contents

## **FC16 (write multiple register) Functions**

<u>Description</u> .....	75
<u>Query frame layout</u> .....	76
<u>Response frame layout</u> .....	76
<u>Exception frame layout</u> .....	77
<u>Function MSXE17xx MFCommonSetInputsFilter</u> .....	77
<u>Description</u> .....	77
<u>Query frame layout</u> .....	78
<u>Response frame layout</u> .....	79
<u>Exception frame layout</u> .....	80
<u>Function MSXE17xx MFCommonReferenceVoltageActivation</u> .....	80
<u>Description</u> .....	80
<u>Query frame layout</u> .....	81
<u>Response frame layout</u> .....	81
<u>Exception frame layout</u> .....	82
<u>Function MSXE17xx MFCommonSetFIFO0Level</u> .....	82
<u>Description</u> .....	82
<u>Query frame layout</u> .....	83
<u>Response frame layout</u> .....	84
<u>Exception frame layout</u> .....	84
<u>Function MSXE17xx MFCommonEnableDisableTriggerGate</u> .....	85
<u>Description</u> .....	85
<u>Query frame layout</u> .....	85
<u>Response frame layout</u> .....	86
<u>Exception frame layout</u> .....	87
<u>Function MSXE17xx MFSinCosInit</u> .....	87
<u>Description</u> .....	87
<u>Query frame layout</u> .....	89
<u>Response frame layout</u> .....	90
<u>Exception frame layout</u> .....	90
<u>Function MSXE17xx MFSinCosClear</u> .....	91
<u>Description</u> .....	91
<u>Query frame layout</u> .....	91
<u>Response frame layout</u> .....	92
<u>Exception frame layout</u> .....	93
<u>Function MSXE17xx MFSinCosRelease</u> .....	93
<u>Description</u> .....	93
<u>Query frame layout</u> .....	94
<u>Response frame layout</u> .....	94
<u>Exception frame layout</u> .....	95
<u>Function MSXE17xx MFSinCosInitAndEnableLatch</u> .....	95
<u>Description</u> .....	95
<u>Query frame layout</u> .....	97
<u>Response frame layout</u> .....	97
<u>Exception frame layout</u> .....	98
<u>Function MSXE17xx MFSinCosDisableAndReleaseLatch</u> .....	98
<u>Description</u> .....	99
<u>Query frame layout</u> .....	99

# Table of Contents

## **FC16 (write multiple register) Functions**

<u>Response frame layout</u> .....	100
<u>Exception frame layout</u> .....	100
<u>Function MSXE17xx MFSinCosInitAndEnableClear</u> .....	101
<u>Description</u> .....	101
<u>Query frame layout</u> .....	102
<u>Response frame layout</u> .....	103
<u>Exception frame layout</u> .....	103
<u>Function MSXE17xx MFSinCosDisableAndReleaseClear</u> .....	104
<u>Description</u> .....	104
<u>Query frame layout</u> .....	104
<u>Response frame layout</u> .....	105
<u>Exception frame layout</u> .....	105
<u>Function MSXE17xx MFSinCosInitAndEnableCompareLogic</u> .....	106
<u>Description</u> .....	106
<u>Query frame layout</u> .....	107
<u>Response frame layout</u> .....	108
<u>Exception frame layout</u> .....	109
<u>Function MSXE17xx MFSinCosDisableAndReleaseCompareLogic</u> .....	109
<u>Description</u> .....	109
<u>Query frame layout</u> .....	109
<u>Response frame layout</u> .....	110
<u>Exception frame layout</u> .....	111
<u>Function MSXE17xx MFSinCosInitHardwareTrigger</u> .....	111
<u>Description</u> .....	111
<u>Query frame layout</u> .....	112
<u>Response frame layout</u> .....	113
<u>Exception frame layout</u> .....	113
<u>Function MSXE17xx MFSinCosReleaseHardwareTrigger</u> .....	114
<u>Description</u> .....	114
<u>Query frame layout</u> .....	114
<u>Response frame layout</u> .....	115
<u>Exception frame layout</u> .....	115
<u>Function MSXE17xx MFSinCosInitIndex</u> .....	116
<u>Description</u> .....	116
<u>Query frame layout</u> .....	116
<u>Response frame layout</u> .....	117
<u>Exception frame layout</u> .....	118
<u>Function MSXE17xx MFSinCosReleaseIndex</u> .....	118
<u>Description</u> .....	118
<u>Query frame layout</u> .....	119
<u>Response frame layout</u> .....	119
<u>Exception frame layout</u> .....	120
<u>Function MSXE17xx MFSinCosInitAbsPos</u> .....	120
<u>Description</u> .....	120
<u>Query frame layout</u> .....	122
<u>Response frame layout</u> .....	123
<u>Exception frame layout</u> .....	124

# Table of Contents

## **FC16 (write multiple register) Functions**

<b><u>Function MSXE17xx DigitalIOWriteAllChannelsValue</u></b> .....	124
<u>Description</u> .....	124
<u>Query frame layout</u> .....	124
<u>Response frame layout</u> .....	125
<u>Exception frame layout</u> .....	126
<b><u>Function MSXE17xx DigitalIORearmShortCircuit</u></b> .....	126
<u>Description</u> .....	126
<u>Query frame layout</u> .....	126
<u>Response frame layout</u> .....	127
<u>Exception frame layout</u> .....	128
<b><u>Function MSXE17xx DigitalIOInitPort</u></b> .....	128
<u>Description</u> .....	128
<u>Query frame layout</u> .....	129
<u>Response frame layout</u> .....	129
<u>Exception frame layout</u> .....	130
<b><u>Function MSXE17xx IOWatchdogInitAndStart</u></b> .....	130
<u>Description</u> .....	130
<u>Query frame layout</u> .....	131
<u>Response frame layout</u> .....	131
<u>Exception frame layout</u> .....	132
<b><u>Function MSXE17xx IOWatchdogStopAndRelease</u></b> .....	132
<u>Description</u> .....	133
<u>Query frame layout</u> .....	133
<u>Response frame layout</u> .....	133
<u>Exception frame layout</u> .....	134

## **FC23 (read/write registers) Functions**.....135

<u>Query frame layout</u> .....	135
<u>Response frame layout</u> .....	136
<u>Exception frame layout</u> .....	136

## **Exception code description**.....137

## **Siemens Step 7 compatibility information (AWL/SDF code)**.....138



# General description

[Top](#)

## Introduction

This document describes the protocol used by the MODBUS server of the module. The OPEN MODBUS protocol is based on the widely known MODBUS protocol. OPEN MODBUS is an open protocol and is not manufacturer dependent. It is mainly used to connect PLC and I/O devices.

## Why a MODBUS Server on the MSX-E modules?

Thanks to the MODBUS server, it is possible to manage an MSX-E module with e.g.: a Siemens S7 PLC. The S7 PLC can start acquisitions and read data from the MSX-E module!

## Technical details

Please note that only MODBUS over TCP is standardized. Nonetheless in this present version the server implements OPEN MODBUS/TCP class 0 and one function of the class 2 even on UDP sockets.

The MODBUS/TCP class 0 defines two types of query: FC3 and FC16.

- **FC3 functions** read register content from the memory of the remote system
- **FC16 functions** write new register content on the memory of the remote system

The MODBUS/TCP server implement the following query of the class 2 : FC23.

- **FC23 functions** read/write registers content from/to the memory of the remote system

The MODBUS server offer a virtual memory organisation: registers (functions) are mapped to be equivalent to SOAP functions.

Characteristics of this communication channel as the standardisation document describes it are:

- The default port used by the server is **512** in both UDP/IP and TCP/IP. You can change this via the web server.
- Data are sent in network order, i.e. **big endian (Motorola formata)**. Use the standard C functions `atons/atohl` and `ntohs/ntohl` to convert values bigger than 1 bytes.
- Datastructures used to describe parameters that are embedded in on-wire frames **must** be packed. How to do that is compiler-dependant.

The ADDI-DATA MSX-E Modbus server offers the following extension to the standard:

- It is possible to configure the server to accept data sent in **little endian (Intel format)** (native order)
- In this case, the default port used is **215**. You can change this via the web server.

## MODBUS interface description

As answer to query a client may receive an acknowledgement (named *standard response* onward) or an exception.

If an exception or an error occurred, you can use the GetLastCommandStatus command to get the real error number (from the remote server).

Real error numbers are described for each command in the "Returns" field.

The chapter below describes the available functions and their parameters.

It also contains the precise description of all frames implied in a given action.

# FC3 (read multiple register) Functions

[Top](#)

Functions in this group are used to read values on the module.

• <a href="#"><u>GetLastCommandStatus</u></a>	Register: <b>0</b>
• <a href="#"><u>GetLastCommandStatusEx</u></a>	Register: <b>10000</b>
• <a href="#"><u>MXCommon_GetModuleType</u></a>	Register: <b>1</b>
• <a href="#"><u>MXCommon_GetModuleTypeEx</u></a>	Register: <b>10200</b>
• <a href="#"><u>MXCommon_GetTime</u></a>	Register: <b>2</b>
• <a href="#"><u>MXCommon_GetTimeEx</u></a>	Register: <b>10500</b>
• <a href="#"><u>MXCommon_TestCustomerID</u></a>	Register: <b>3</b>
• <a href="#"><u>MXCommon_TestCustomerIDEx</u></a>	Register: <b>10550</b>
• <a href="#"><u>MSXE17xx_MFSinCosReadAll</u></a>	Register: <b>1000</b>
• <a href="#"><u>MSXE17xx_MFSinCosRead0</u></a>	Register: <b>1050</b>
• <a href="#"><u>MSXE17xx_MFSinCosRead1</u></a>	Register: <b>1100</b>
• <a href="#"><u>MSXE17xx_MFSinCosRead2</u></a>	Register: <b>1150</b>
• <a href="#"><u>MSXE17xx_MFSinCosRead3</u></a>	Register: <b>1200</b>
• <a href="#"><u>MSXE17xx_MFSinCosReadAbsPos0</u></a>	Register: <b>2000</b>
• <a href="#"><u>MSXE17xx_MFSinCosReadAbsPos1</u></a>	Register: <b>2050</b>
• <a href="#"><u>MSXE17xx_MFSinCosReadAbsPos2</u></a>	Register: <b>2100</b>
• <a href="#"><u>MSXE17xx_MFSinCosReadAbsPos3</u></a>	Register: <b>2150</b>
• <a href="#"><u>MSXE17xx_MFSinCosReadAbsPosAll</u></a>	Register: <b>2200</b>
• <a href="#"><u>MSXE17xx_DigitalIOReadAllChannelsValue</u></a>	Register: <b>7000</b>
• <a href="#"><u>MSXE17xx_DigitalIOTestShortCircuit</u></a>	Register: <b>7050</b>

## Function GetLastCommandStatus

**For new application(s) or automate communication it is recommended to use the function GetLastCommandStatusEx.**

### Description

Return the result of the last remote function call

#### Parameters:

[Response frame layout] **ReturnValue:** The return value of the remote function.

- ◆ 0 Always means success
- ◆ -100 means you should check Syserrno;
- ◆ for other values, check the documentation of the function

[Response frame layout] **Syserrno:** the value of the libc errno after the call to the remote function

[Response frame layout] **Errstr:** A nul-terminated string describing the error code Syserrno

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	0	0x0000	0x0000
word count	2	16-bit integer	54	0x3600	0x0036

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	112	0x7000	0x0070
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	108	0x6C00	0x006C
ReturnValue	4	32-bit integer	See the description above	0x???????	0x???????
Syserrno	4	32-bit integer	See the description above	0x???????	0x???????
Errstr	100	8-bit integer array	See the description above	0x??[100]	0x??[100]

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83

Exception code	1	8-bit integer	See corresponding chapter	??	??
----------------	---	---------------	---------------------------	----	----

## Function GetLastCommandStatusEx

### Description

Return the result of the last remote function call

#### Parameters:

[Response frame layout] **ReturnValue:** The return value of the remote function.

- ◆ 0 Always means success
- ◆ -100 means you should check Syserrno;
- ◆ for other values, check the documentation of the function

[Response frame layout] **Syserrno:** the value of the libc errno after the call to the remote function

[Response frame layout] **Errstr:** A nul-terminated string describing the error code Syserrno

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	10000	0x1027	0x2710
word count	2	16-bit integer	54	0x3600	0x0036

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	111	0x6F00	0x006F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	108	0x6C	0x6C
ReturnValue	4	32-bit integer	See the description above	0x???????	0x???????
Syserrno	4	32-bit integer	See the description above	0x???????	0x???????
Errstr	100	8-bit integer array	See the description above	0x??[100]	0x??[100]

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83

Exception code	1	8-bit integer	See corresponding chapter	??	??
----------------	---	---------------	---------------------------	----	----

## Function MXCommon\_\_GetModuleType

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_GetModuleTypeEx.

### Description

Returns the type of the MSX-E Module

#### Parameters:

[Response frame layout] **str**: A 200-characters string

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1	0x0100	0x0001
word count	2	16-bit integer	100	0x6400	0x0064

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
-------	--------------	------	-------	-----------------------	-----------------------

Exception frame layout



## MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	204	0xCC00	0x00CC
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	200	0xC800	0x00C8
str	200	8-bit integer array	See the description above	0x??[200]	0x??[200]

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MXCommon\_\_GetModuleTypeEx

### Description

Returns the type of the MSX-E Module

#### Parameters:

Response frame layout

[Response frame layout] **str**: A 200-characters string

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	10200	0xD827	0x27D8
word count	2	16-bit integer	100	0x6400	0x0064

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	203	0xCB00	0x00CB
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	200	0xC8	0xC8

## MODBUS interface description

str	200	8-bit integer array	See the description above	0x??[200]	0x??[200]
-----	-----	---------------------	---------------------------	-----------	-----------

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MXCommon\_\_GetTime

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_GetTimeEx.

### Description

Get the time on the module

#### Parameters:

[Response frame layout] **tv\_sec:** Number of seconds since the Epoch

[Response frame layout] **tv\_usec:** Number of microseconds since the begin of the second

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied	0x0000	0x0000

Response frame layout

## MODBUS interface description

			by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	2	0x0200	0x0002
word count	2	16-bit integer	4	0x0400	0x0004

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	12	0x0C00	0x000C
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	8	0x0800	0x0008
tv_sec	4	32-bit integer	See the description above	0x???????	0x???????
tv_usec	4	32-bit integer	See the description above	0x???????	0x???????

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MXCommon\_\_GetTimeEx

### Description

Get the time on the module

#### Parameters:

[Response frame layout] **tv\_sec**: Number of seconds since the Epoch

[Response frame layout] **tv\_usec**: Number of microseconds since the begin of the second

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000

## MODBUS interface description

length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	10500	0x0429	0x2904
word count	2	16-bit integer	4	0x0400	0x0004

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	8	0x08	0x08
tv_sec	4	32-bit integer	See the description above	0x???????	0x???????
tv_usec	4	32-bit integer	See the description above	0x???????	0x???????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000

## MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MXCommon\_\_TestCustomerID

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_TestCustomerIDEx.

### Description

Permit to test the Customer ID (if the module has the right customer Key )

#### Parameters:

[Response frame layout] **bValueArray**: non crypted value array [16 bytes of random data]

[Response frame layout] **bCryptedValueArray**: Crypted value array [16 bytes of the crypted random data]

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
	1		0x03	0x03	0x03

Exception frame layout

## MODBUS interface description

MODBUS Function code		8-bit integer			
Reference number (=register)	2	16-bit integer	3	0x0300	0x0003
word count	2	16-bit integer	16	0x1000	0x0010

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	36	0x2400	0x0024
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	32	0x2000	0x0020
bValueArray	16	8-bit integer array	See the description above	0x??[16]	0x??[16]
bCryptedValueArray	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
	1		0 or 1		

### Query frame layout



## MODBUS interface description

unit identifier		8-bit integer		0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MXCommon\_\_TestCustomerIDEx

### Description

Permit to test the Customer ID (if the module has the right customer Key )

#### Parameters:

[Response frame layout] **bValueArray**: non crypted value array [16 bytes of random data]

[Response frame layout] **bCryptedValueArray**: Crypted value array [16 bytes of the crypted random data]

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	10550	0x3629	0x2936
word count	2	16-bit integer	16	0x1000	0x0010

Exception frame layout

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	35	0x2300	0x0023
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	32	0x20	0x20
bValueArray	16	8-bit integer array	See the description above	0x??[16]	0x??[16]
bCryptedValueArray	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

# Function MSXE17xx\_\_MFSinCosReadAll

## Description

Read measured value on all modules.

### Parameters:

[Response frame layout] **ulValues** : Array that contain the measured values, as raw or converted in the selected format.

- ◆ ulValues [0] : Module 0 value
- ◆ ulValues [1] : Module 1 value
- ◆ ulValues [2] : Module 2 value
- ◆ ulValues [3] : Module 3 value

[Response frame layout] **ulMeasureErrors** : Array that contain the measure errors.

- ◆ ulMeasureErrors[0] : Module 0
  - ◇ 0 : No error.
  - ◇ 1 : Amplitude error.
  - ◇ 2 : Frequency error (in fast mode is this error not relevant).
  - ◇ 3 : Amplitude and frequency error.
- ◆ ulMeasureErrors[1] : Module 1
  - ◇ 0 : No error.
  - ◇ 1 : Amplitude error.
  - ◇ 2 : Frequency error (in fast mode is this error not relevant).
  - ◇ 3 : Amplitude and frequency error.
- ◆ ulMeasureErrors[2] : Module 2
  - ◇ 0 : No error.
  - ◇ 1 : Amplitude error.
  - ◇ 2 : Frequency error (in fast mode is this error not relevant).
  - ◇ 3 : Amplitude and frequency error.
- ◆ ulMeasureErrors[3] : Module 3
  - ◇ 0 : No error.
  - ◇ 1 : Amplitude error.
  - ◇ 2 : Frequency error (in fast mode is this error not relevant).
  - ◇ 3 : Amplitude and frequency error.

### Returns:

- Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
  - ◆ 0 : No error.
  - ◆ -1 : means an system error occurred
  - ◆ -2 : Multifunction sub module index selection error.
  - ◆ -3 : Multifunction sub module is not a SinCos module.
  - ◆ -4 : Sinus / Cosinus module not initialised

## MODBUS interface description

♦ -100 : Read Sinus Cosinus kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1000	0xE803	0x03E8
word count	2	16-bit integer	16	0x1000	0x0010

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	35	0x2300	0x0023
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	32	0x20	0x20
ulValues	16	32-bit	See the	0x????????[4]	0x????????[4]

## MODBUS interface description

		integer array	description above		
ulMeasureErrors	16	32-bit integer array	See the description above	0x????????[4]	0x????????[4]

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_MFSinCosRead0

### Description

Read measured value on the module 0.

#### Parameters:

[Response frame layout] **ulValue** : Measured value, as raw or converted in the selected format.

[Response frame layout] **ulMeasureError** : Measure error.

- ◆ 0 : No error.
- ◆ 1 : Amplitude error.
- ◆ 2 : Frequency error (in fast mode is this error not relevant).
- ◆ 3 : Amplitude and frequency error.

#### Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - ◆ 0 : No error.
  - ◆ -1 : means an system error occurred

Response frame layout

## MODBUS interface description

- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1050	0x1A04	0x041A
word count	2	16-bit integer	4	0x0400	0x0004

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03

## MODBUS interface description

Byte count	1	8-bit integer	8	0x08	0x08
ulValue	4	32-bit integer	See the description above	0x????????	0x????????
ulMeasureError	4	32-bit integer	See the description above	0x????????	0x????????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_MFSinCosRead1

### Description

Read measured value on the module 1.

#### Parameters:

[Response frame layout] **ulValue** : Measured value, as raw or converted in the selected format.

[Response frame layout] **ulMeasureError** : Measure error.

- ◆ 0 : No error.
- ◆ 1 : Amplitude error.
- ◆ 2 : Frequency error (in fast mode is this error not relevant).
- ◆ 3 : Amplitude and frequency error.

#### Returns:

## MODBUS interface description

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1100	0x4C04	0x044C
word count	2	16-bit integer	4	0x0400	0x0004

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1		0 or 1		



## MODBUS interface description

		8-bit integer		0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	8	0x08	0x08
ulValue	4	32-bit integer	See the description above	0x???????	0x???????
ulMeasureError	4	32-bit integer	See the description above	0x???????	0x???????

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_MFSinCosRead2

### Description

Read measured value on the module 2.

#### Parameters:

[Response frame layout] **ulValue** : Measured value, as raw or converted in the selected format.

[Response frame layout] **ulMeasureError** : Measure error.

- ◆ 0 : No error.
- ◆ 1 : Amplitude error.

## MODBUS interface description

- ◆ 2 : Frequency error (in fast mode is this error not relevant).
- ◆ 3 : Amplitude and frequency error.

### Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1150	0x7E04	0x047E
word count	2	16-bit integer	4	0x0400	0x0004

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000

## MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	8	0x08	0x08
ulValue	4	32-bit integer	See the description above	0x???????	0x???????
ulMeasureError	4	32-bit integer	See the description above	0x???????	0x???????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_MFSinCosRead3

### Description

Read measured value on the module 3.

#### Parameters:

[Response frame layout] **ulValue** : Measured value, as raw or converted in the selected format.

## MODBUS interface description

[Response frame layout] ***ulMeasureError*** : Measure error.

- ◆ 0 : No error.
- ◆ 1 : Amplitude error.
- ◆ 2 : Frequency error (in fast mode is this error not relevant).
- ◆ 3 : Amplitude and frequency error.

### Returns:

• **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1200	0xB004	0x04B0
word count	2	16-bit integer	4	0x0400	0x0004

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
	2			0x0000	0x0000

## MODBUS interface description

transaction identifier		16-bit integer	User defined - copied by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	8	0x08	0x08
ulValue	4	32-bit integer	See the description above	0x????????	0x????????
ulMeasureError	4	32-bit integer	See the description above	0x????????	0x????????

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_MFSinCosReadAbsPos0

## Description

Read measured value and absolute position on the module 0.

### Parameters:

[Response frame layout] **ulValue** : Measured value, as raw or converted in the selected format.

[Response frame layout] **ulAbsPos** : Measured absolute position value, as raw or converted in the selected format and only if ulAbsPosStatus = 1.

[Response frame layout] **ulAbsPosStatus** : 0 if absolute position not available. 1 if absolute position available

[Response frame layout] **ulMeasureError** : Measure error.

- ◆ 0 : No error.
- ◆ 1 : Amplitude error.
- ◆ 2 : Frequency error (in fast mode is this error not relevant).
- ◆ 3 : Amplitude and frequency error.

### Returns:

• **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
	1		0x03	0x03	0x03

## MODBUS interface description

MODBUS Function code		8-bit integer			
Reference number (=register)	2	16-bit integer	2000	0xD007	0x07D0
word count	2	16-bit integer	8	0x0800	0x0008

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	16	0x10	0x10
ulValue	4	32-bit integer	See the description above	0x????????	0x????????
ulAbsPos	4	32-bit integer	See the description above	0x????????	0x????????
ulAbsPosStatus	4	32-bit integer	See the description above	0x????????	0x????????
ulMeasureError	4	32-bit integer	See the description above	0x????????	0x????????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server -	0x0000	0x0000

### Query frame layout

## MODBUS interface description

			usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_MFSinCosReadAbsPos1

### Description

Read measured value and absolut position on the module 1.

#### Parameters:

[Response frame layout] **ulValue** : Measured value, as raw or converted in the selected format.

[Response frame layout] **ulAbsPos** : Measured absolut position value, as raw or converted in the selected format and only if ulAbsPosStatus = 1.

[Response frame layout] **ulAbsPosStatus** : 0 if absolut position not available. 1 if absolut position available

[Response frame layout] **ulMeasureError** : Measure error.

- ◆ 0 : No error.
- ◆ 1 : Amplitude error.
- ◆ 2 : Frequency error (in fast mode is this error not relevant).
- ◆ 3 : Amplitude and frequency error.

#### Returns:

##### • Possible return value on the remote system (read them with GetLastCommandStatusEx):

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).



## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	2050	0x0208	0x0802
word count	2	16-bit integer	8	0x0800	0x0008

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	16	0x10	0x10
ulValue	4	32-bit integer	See the description	0x????????	0x????????

## MODBUS interface description

			above		
ulAbsPos	4	32-bit integer	See the description above	0x????????	0x????????
ulAbsPosStatus	4	32-bit integer	See the description above	0x????????	0x????????
ulMeasureError	4	32-bit integer	See the description above	0x????????	0x????????

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_MFSinCosReadAbsPos2

### Description

Read measured value and absolut position on the module 2.

### Parameters:

[Response frame layout] **ulValue** : Measured value, as raw or converted in the selected format.

[Response frame layout] **ulAbsPos** : Measured absolut position value, as raw or converted in the selected format and only if ulAbsPosStatus = 1.

[Response frame layout] **ulAbsPosStatus** : 0 if absolut position not available. 1 if absolut position available

## MODBUS interface description

[Response frame layout] ***ulMeasureError***: : Measure error.

- ◆ 0 : No error.
- ◆ 1 : Amplitude error.
- ◆ 2 : Frequency error (in fast mode is this error not relevant).
- ◆ 3 : Amplitude and frequency error.

### Returns:

• **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	2100	0x3408	0x0834
word count	2	16-bit integer	8	0x0800	0x0008

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
	2			0x0000	0x0000

## MODBUS interface description

transaction identifier		16-bit integer	User defined - copied by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	16	0x10	0x10
ulValue	4	32-bit integer	See the description above	0x????????	0x????????
ulAbsPos	4	32-bit integer	See the description above	0x????????	0x????????
ulAbsPosStatus	4	32-bit integer	See the description above	0x????????	0x????????
ulMeasureError	4	32-bit integer	See the description above	0x????????	0x????????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

### Response frame layout

## Function MSXE17xx\_\_MFSinCosReadAbsPos3

### Description

Read measured value and absolut position on the module 3.

#### Parameters:

[Response frame layout] **ulValue** : Measured value, as raw or converted in the selected format.

[Response frame layout] **ulAbsPos** : Measured absolut position value, as raw or converted in the selected format and only if ulAbsPosStatus = 1.

[Response frame layout] **ulAbsPosStatus** : 0 if absolut position not available. 1 if absolut position available

[Response frame layout] **ulMeasureError** : Measure error.

- ◆ 0 : No error.
- ◆ 1 : Amplitude error.
- ◆ 2 : Frequency error (in fast mode is this error not relevant).
- ◆ 3 : Amplitude and frequency error.

#### Returns:

• **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006

## MODBUS interface description

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	2150	0x6608	0x0866
word count	2	16-bit integer	8	0x0800	0x0008

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	16	0x10	0x10
ulValue	4	32-bit integer	See the description above	0x????????	0x????????
ulAbsPos	4	32-bit integer	See the description above	0x????????	0x????????
ulAbsPosStatus	4	32-bit integer	See the description above	0x????????	0x????????
ulMeasureError	4	32-bit integer	See the description above	0x????????	0x????????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
	2			0x0000	0x0000

### Query frame layout

## MODBUS interface description

transaction identifier		16-bit integer	User defined - copied by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_MFSinCosReadAbsPosAll

### Description

Read measured value and absolut position on the all the modules

#### Parameters:

[Response frame layout] **ulValues** : Array that contain the measured values, as raw or converted in the selected format.

- ◆ ulValues [0] : Module 0 value
- ◆ ulValues [1] : Module 1 value
- ◆ ulValues [2] : Module 2 value
- ◆ ulValues [3] : Module 3 value

[Response frame layout] **ulAbsPos** : Array that contain measured absolut position value, as raw or converted in the selected format and only if ulAbsPosStatus = 1.

- ◆ ulAbsPos [0] : Module 0 absolut position value
- ◆ ulAbsPos [1] : Module 1 absolut position value
- ◆ ulAbsPos [2] : Module 2 absolut position value
- ◆ ulAbsPos [3] : Module 3 absolut position value

[Response frame layout] **ulAbsPosStatus** : Array that contain measured absolut position status

- ◆ ulAbsPosStatus[0] : Module 0
  - ◇ 0 : Absolut position not available.
  - ◇ 1 : Absolut position available
- ◆ ulAbsPosStatus[1] : Module 1
  - ◇ 0 : Absolut position not available.
  - ◇ 1 : Absolut position available

## MODBUS interface description

### ◆ ulAbsPosStatus[2] : Module 2

◇ 0 : Absolut position not available.

◇ 1 : Absolut position available

### ◆ ulAbsPosStatus[3] : Module 3

◇ 0 : Absolut position not available.

◇ 1 : Absolut position available

[Response frame layout] **ulMeasureErrors** : Array that contain the measure errors.

### ◆ ulMeasureErrors[0] : Module 0

◇ 0 : No error.

◇ 1 : Amplitude error.

◇ 2 : Frequency error (in fast mode is this error not relevant).

◇ 3 : Amplitude and frequency error.

### ◆ ulMeasureErrors[1] : Module 1

◇ 0 : No error.

◇ 1 : Amplitude error.

◇ 2 : Frequency error (in fast mode is this error not relevant).

◇ 3 : Amplitude and frequency error.

### ◆ ulMeasureErrors[2] : Module 2

◇ 0 : No error.

◇ 1 : Amplitude error.

◇ 2 : Frequency error (in fast mode is this error not relevant).

◇ 3 : Amplitude and frequency error.

### ◆ ulMeasureErrors[3] : Module 3

◇ 0 : No error.

◇ 1 : Amplitude error.

◇ 2 : Frequency error (in fast mode is this error not relevant).

◇ 3 : Amplitude and frequency error.

## Returns:

### • Possible return value on the remote system (read them with GetLastCommandStatusEx):

◆ 0 : No error.

◆ -1 : means an system error occurred

◆ -2 : Multifunction sub module index selection error.

◆ -3 : Multifunction sub module is not a SinCos module.

◆ -4 : Sinus / Cosinus module not initialised

◆ -100 : Read Sinus Cosinus kernel function error (see syserrno).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
-------	-----------------	------	-------	-----------------------------	--------------------------



## MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	2200	0x9808	0x0898
word count	2	16-bit integer	32	0x2000	0x0020

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	67	0x4300	0x0043
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	64	0x40	0x40
ulValues	16	32-bit integer array	See the description above	0x????????[4]	0x????????[4]
ulAbsPos	16	32-bit integer array	See the description above	0x????????[4]	0x????????[4]
ulAbsPosStatus	16	32-bit integer array	See the description above	0x????????[4]	0x????????[4]

### Query frame layout

## MODBUS interface description

ulMeasureErrors	16	32-bit integer array	See the description above	0x???????[4]	0x???????[4]
-----------------	----	----------------------	---------------------------	--------------	--------------

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_DigitalIOReadAllChannelsValue

### Description

Read all the digital I/O channel value. If channel is configured as output, then this function return the status of the output

#### Parameters:

[Response frame layout]**ulChannelsValue** : Channels value

#### Returns:

- Possible return value on the remote system (read them with GetLastErrorStatusEx):
  - ◆ 0 : No error.
  - ◆ -1 : Means an system error occurred
  - ◆ -100 : Kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
-------	--------------	------	-------	-----------------------	-----------------------

## MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	7000	0x581B	0x1B58
word count	2	16-bit integer	2	0x0200	0x0002

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	7	0x0700	0x0007
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	4	0x04	0x04
ulChannelsValue	4	32-bit integer	See the description above	0x???????	0x???????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian	big endian (Motorola)
-------	--------------	------	-------	---------------	-----------------------

Query frame layout

## MODBUS interface description

				(Intel)	
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_DigitalIOTestShortCircuit

### Description

#### Parameters:

[Response frame layout] **ulValue** : short circuit status: from 0 to 0xffff, one bit for each output

- ◆ 0 : no short circuit
- ◆ 1 : short circuit

#### Returns:

- Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
  - ◆ 0 : No error.
  - ◆ -1 : Means an system error ocured
  - ◆ -100 : Kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
	2		0	0x0000	0x0000

Exception frame layout

## MODBUS interface description

protocol identifier		16-bit integer			
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	7050	0x8A1B	0x1B8A
word count	2	16-bit integer	2	0x0200	0x0002

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	7	0x0700	0x0007
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	4	0x04	0x04
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
	2		0	0x0000	0x0000

### Query frame layout

## MODBUS interface description

protocol identifier		16-bit integer			
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE17xx\_\_IOWatchdogGetStatusAndValue

### Description

Get watchdog current status and value information

#### Parameters:

[Response frame layout] **ulStatus** : Channels value

- ◆ BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX0: is stopped
- ◆ BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX1: is running
- ◆ BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX0X: is not run down
- ◆ BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX1X: is run down

#### Returns:

- Possible return value on the remote system (read them with GetLastCommandStatusEx):
  - ◆ 0 : No error.
  - ◆ -1 : Means an system error occurred
  - ◆ -100 : Kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000

Exception frame layout

## MODBUS interface description

length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	8000	0x401F	0x1F40
word count	2	16-bit integer	6	0x0600	0x0006

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	12	0x0C	0x0C
ulStatus	4	32-bit integer	See the description above	0x????????	0x????????
ulValue	4	32-bit integer	See the description above	0x????????	0x????????
ulInfo	4	32-bit integer	See the description above	0x????????	0x????????

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
	2			0x0000	0x0000

### Query frame layout

## MODBUS interface description

transaction identifier		16-bit integer	User defined - copied by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??



# FC16 (write multiple register) Functions

[Top](#)

Functions in this group are used to set value on the module.

- MXCommon\_SetHardwareTriggerFilterTime Register: **100**
- MXCommon\_SetHardwareTriggerFilterTimeEx Register: **11000**
- MXCommon\_InitAndStartSynchroTimer Register: **101**
- MXCommon\_InitAndStartSynchroTimerEx Register: **11050**
- MXCommon\_StopAndReleaseSynchroTimer Register: **102**
- MXCommon\_StopAndReleaseSynchroTimerEx Register: **11100**
- MXCommon\_Reboot Register: **103**
- MXCommon\_RebootEx Register: **11150**
- MXCommon\_SetCustomerKey Register: **104**
- MXCommon\_SetCustomerKeyEx Register: **11200**
- MXCommon\_SetFilterChannels Register: **105**
- MXCommon\_SetFilterChannelsEx Register: **11250**
- MSXE17xx\_MFCommonSetInputsFilter Register: **6000**
- MSXE17xx\_MFCommonReferenceVoltageActivation Register: **6050**
- MSXE17xx\_MFCommonSetFIFO0Level Register: **6100**
- MSXE17xx\_MFCommonEnableDisableTriggerGate Register: **6150**
- MSXE17xx\_MFSinCosInit Register: **1250**
- MSXE17xx\_MFSinCosClear Register: **1300**
- MSXE17xx\_MFSinCosRelease Register: **1350**
- MSXE17xx\_MFSinCosInitAndEnableLatch Register: **1400**

## MODBUS interface description

• <u>MSXE17xx MFSinCosDisableAndReleaseLatch</u>	Register: <b>1450</b>
• <u>MSXE17xx MFSinCosInitAndEnableClear</u>	Register: <b>1500</b>
• <u>MSXE17xx MFSinCosDisableAndReleaseClear</u>	Register: <b>1550</b>
• <u>MSXE17xx MFSinCosInitAndEnableCompareLogic</u>	Register: <b>1600</b>
• <u>MSXE17xx MFSinCosDisableAndReleaseCompareLogic</u>	Register: <b>1650</b>
• <u>MSXE17xx MFSinCosInitHardwareTrigger</u>	Register: <b>1700</b>
• <u>MSXE17xx MFSinCosReleaseHardwareTrigger</u>	Register: <b>1750</b>
• <u>MSXE17xx MFSinCosInitIndex</u>	Register: <b>1800</b>
• <u>MSXE17xx MFSinCosReleaseIndex</u>	Register: <b>1850</b>
• <u>MSXE17xx MFSinCosInitAbsPos</u>	Register: <b>1900</b>
• <u>MSXE17xx DigitalIOWriteAllChannelsValue</u>	Register: <b>7100</b>
• <u>MSXE17xx DigitalIORearmShortCircuit</u>	Register: <b>7150</b>
• <u>MSXE17xx DigitalIOInitPort</u>	Register: <b>7200</b>
• <u>MSXE17xx IOWatchdogInitAndStart</u>	Register: <b>8050</b>
• <u>MSXE17xx IOWatchdogStopAndRelease</u>	Register: <b>8100</b>

## Function MXCommon\_\_SetHardwareTriggerFilterTime

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_SetHardwareTriggerFilterTimeEx.

### Description

Sets the filter time for the hardware trigger input in **250ns** step (max value : 65535 ).

On the MSX-E3011 system, the step of the hardware trigger filter is **622ns**.

### Parameters

- [Query frame layout] **ulFilterTime** Filter time for the hardware trigger input in 250ns step (max value : 65535 ).
  - ◆ **0**: disable the filter

## MODBUS interface description

- ◆ 1: filter of 250ns
- ◆ 2: filter of 500ns
- ◆ ...
- ◆ 65535: filter of 16ms
- [Query frame layout] **ulOption** Reserved. Set to 0

### Returns

Possible return value on the remote system (read them with GetLastCommandStatus).

- 0 The remote function performed OK
- -1 Internal system error occurred. See value of syserrno

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	16	0x1000	0x0010
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	100	0x6400	0x0064
word count	2	16-bit integer	4	0x0400	0x0004
byte count	2	16-bit integer	8	0x0800	0x0008
ulFilterTime	4	32-bit integer	See the description above	0x????????	0x????????
Reserved	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	100	0x6400	0x0064
word count	2	16-bit integer	4	0x0400	0x0004

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

# Function MXCommon\_\_SetHardwareTriggerFilterTimeEx

## Description

Sets the filter time for the hardware trigger input in **250ns** step (max value : 65535 ).

On the MSX-E3011 system, the step of the hardware trigger filter is **622ns**.

## Parameters

- [Query frame layout] **ulFilterTime** Filter time for the hardware trigger input in 250ns step (max value : 65535 ).
  - ◆ **0**: disable the filter
  - ◆ **1**: filter of 250ns
  - ◆ **2**: filter of 500ns
  - ◆ ...
  - ◆ **65535**: filter of 16ms
- [Query frame layout] **ulOption** Reserved. Set to 0

## Returns

Possible return value on the remote system (read them with GetLastCommandStatusEx).

- **0** The remote function performed OK
- **-1** Internal system error occurred. See value of syserrno

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11000	0xF82A	0x2AF8
word count	2		4	0x0400	0x0004

## MODBUS interface description

		16-bit integer			
byte count	1	8-bit integer	8	0x08	0x08
ulFilterTime	4	32-bit integer	See the description above	0x????????	0x????????
Reserved	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11000	0xF82A	0x2AF8
word count	2	16-bit integer	4	0x0400	0x0004

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit	3	0x0300	0x0003

### Query frame layout

## MODBUS interface description

		integer			
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_InitAndStartSynchroTimer

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_InitAndStartSynchroTimerEx.

### Description

Init and start the synchronisation timer of the module (not already available on all module)

#### Parameters:

[Query frame layout] **ulTimeBase:** Time base of the timer (0 for us, 1 for ms, 2 for s)

[Query frame layout] **ulReloadValue:** Timer reload value (0 to 0xFFFF), minimum reload time is 5 us

[Query frame layout] **ulNbrOfCycle:** Number of timer cycle

- ◆ 0: continuous
- ◆ > 0: defined number of cycle

[Query frame layout] **ulGenerateTriggerMode:**

- ◆ 0: Wait the time overflow to set the synchronisation trigger
- ◆ 1: Set the synchronisation trigger by the start of the timer and after each time overflow

[Query frame layout] **ulOption01:** Define the source of the trigger

- ◆ 0 : Trigger disabled
- ◆ 1 : Enable the hardware figital input trigger

[Query frame layout] **ulOption02:** Define the edge of the hardware trigger who generates a trigger action

- ◆ 1 : rising edge (Only if hardware trigger selected)
- ◆ 2 : falling edge (Only if hardware trigger selected)
- ◆ 3 : Both front (Only if hardware trigger selected)

[Query frame layout] **ulOption03:** Define the number of trigger events before the action occur

- ◆ 1 : all trigger event start the action
- ◆ max value : 65535

[Query frame layout] **ulOption04:** Reserved

**Returns:**

**Possible return value on the remote system (read them with GetLastCommandStatus)**

- ◆ 0 : means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: not available time base
- ◆ -3: timer reload value can not be greater than 65535
- ◆ -4: minimum time reload is 5 us
- ◆ -5: Number of cycle can not be greater than 65535
- ◆ -6: Generate trigger mode error
- ◆ -100: Init timer error
- ◆ -101: Start timer error

**Query frame layout**

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	40	0x2800	0x0028
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	101	0x6500	0x0065
word count	2	16-bit integer	16	0x1000	0x0010
byte count	2	16-bit integer	32	0x2000	0x0020
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulReloadValue	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrOfCycle	4	32-bit integer	See the description above	0x????????	0x????????
ulGenerateTriggerMode	4	32-bit integer	See the description	0x????????	0x????????



## MODBUS interface description

			above		
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????
ulOption03	4	32-bit integer	See the description above	0x????????	0x????????
ulOption04	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	101	0x6500	0x0065
word count	2	16-bit integer	16	0x1000	0x0010

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000

## MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_InitAndStartSynchroTimerEx

### Description

Init and start the synchronisation timer of the module (not already available on all module)

#### Parameters:

[Query frame layout] **ulTimeBase:** Time base of the timer (0 for us, 1 for ms, 2 for s)

[Query frame layout] **ulReloadValue:** Timer reload value (0 to 0xFFFF), minimum reload time is 5 us

[Query frame layout] **ulNbrOfCycle:** Number of timer cycle

- ◆ 0: continuous
- ◆ > 0: defined number of cycle

[Query frame layout] **ulGenerateTriggerMode:**

- ◆ 0: Wait the time overflow to set the synchronisation trigger
- ◆ 1: Set the synchronisation trigger by the start of the timer and after each time overflow

[Query frame layout] **ulOption01:** Define the source of the trigger

- ◆ 0 : Trigger disabled
- ◆ 1 : Enable the hardware figital input trigger

[Query frame layout] **ulOption02:** Define the edge of the hardware trigger who generates a trigger action

- ◆ 1 : rising edge (Only if hardware trigger selected)
- ◆ 2 : falling edge (Only if hardware trigger selected)
- ◆ 3 : Both front (Only if hardware trigger selected)

[Query frame layout] **ulOption03:** Define the number of trigger events before the action occur

- ◆ 1 : all trigger event start the action
- ◆ max value : 65535

[Query frame layout] **ulOption04:** Reserved

## MODBUS interface description

### Returns:

**Possible return value on the remote system (read them with GetLastCommandStatusEx)**

- ◆ 0 : means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: not available time base
- ◆ -3: timer reload value can not be greater than 65535
- ◆ -4: minimum time reload is 5 us
- ◆ -5: Number of cycle can not be greater than 65535
- ◆ -6: Generate trigger mode error
- ◆ -100: Init timer error
- ◆ -101: Start timer error

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	39	0x2700	0x0027
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11050	0x2A2B	0x2B2A
word count	2	16-bit integer	16	0x1000	0x0010
byte count	1	8-bit integer	32	0x20	0x20
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulReloadValue	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrOfCycle	4	32-bit integer	See the description above	0x????????	0x????????
ulGenerateTriggerMode	4	32-bit integer	See the description	0x????????	0x????????

## MODBUS interface description

			above		
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????
ulOption03	4	32-bit integer	See the description above	0x????????	0x????????
ulOption04	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11050	0x2A2B	0x2B2A
word count	2	16-bit integer	16	0x1000	0x0010

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000

### Query frame layout

## MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_StopAndReleaseSynchroTimer

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_StopAndReleaseSynchroTimerEx.

### Description

stop the synchronisation timer (not already available on all module)

#### Parameters:

[Query frame layout] **ulOption01** : Reserved

#### Returns:

Possible return value on the remote system (read them with GetLastCommandStatus)

- ◆ 0 : means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -100: Start/Stop timer error

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	12	0x0C00	0x000C

Exception frame layout

## MODBUS interface description

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	102	0x6600	0x0066
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	102	0x6600	0x0066
word count	2	16-bit integer	2	0x0200	0x0002

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by	0x0000	0x0000

### Query frame layout

## MODBUS interface description

			server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_StopAndReleaseSynchroTimerEx

### Description

stop the synchronisation timer (not already available on all module)

#### Parameters:

[Query frame layout] ***ulOption01*** : Reserved

#### Returns:

Possible return value on the remote system (read them with **GetLastCommandStatusEx**)

- ◆ 0 : means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -100: Start/Stop timer error

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit	1	8-bit	0 or 1	0x00 or	0x00 or

Exception frame layout

## MODBUS interface description

identifier		integer		0x01	0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11100	0x5C2B	0x2B5C
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11100	0x5C2B	0x2B5C
word count	2	16-bit integer	2	0x0200	0x0002

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server -	0x0000	0x0000

### Query frame layout



## MODBUS interface description

			usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_Reboot

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_RebootEx.

### Description

Ask the MSX-E module to reboot

#### Parameters:

[Query frame layout] **Dummy** : Reserved

#### Returns:

Possible return value on the remote system (read them with GetLastCommandStatus)

- ◆ 0 : means the remote function performed OK
- ◆ -1: means an system error occured (probably EPERM)

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	12	0x0C00	0x000C

Exception frame layout

## MODBUS interface description

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	103	0x6700	0x0067
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
Dummy	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	103	0x6700	0x0067
word count	2	16-bit integer	2	0x0200	0x0002

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by	0x0000	0x0000

### Query frame layout

## MODBUS interface description

			server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_RebootEx

### Description

Ask the MSX-E module to reboot

#### Parameters:

[Query frame layout] **Dummy** : Reserved

#### Returns:

Possible return value on the remote system (read them with GetLastCommandStatusEx)

- ◆ 0 : means the remote function performed OK
- ◆ -1: means an system error occurred (probably EPERM)

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01

Exception frame layout

## MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11150	0x8E2B	0x2B8E
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
Dummy	4	32-bit integer	See the description above	0x???????	0x???????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11150	0x8E2B	0x2B8E
word count	2	16-bit integer	2	0x0200	0x0002

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000

## MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_SetCustomerKey

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_SetCustomerKeyEx.

### Description

Permit to set the Customer key

#### Parameters:

[Query frame layout] **bKey** : Customer key (only writable on the module) [32 bytes containing a AES key]

[Query frame layout] **bPublicKey** : IV (Initialisation vector) for the AES cryptography [16 bytes containing a AES key]

#### Returns:

Possible return value on the remote system (read them with GetLastCommandStatus)

- ◆ 0 : means the remote function performed OK
- ◆ -1: means an system error occured (probably EPERM)

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
	2		0	0x0000	0x0000

Exception frame layout

## MODBUS interface description

protocol identifier		16-bit integer			
length	2	16-bit integer	56	0x3800	0x0038
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	104	0x6800	0x0068
word count	2	16-bit integer	24	0x1800	0x0018
byte count	2	16-bit integer	48	0x3000	0x0030
bKey	32	8-bit integer array	See the description above	0x??[32]	0x??[32]
bPublicKey	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	104	0x6800	0x0068
word count	2	16-bit integer	24	0x1800	0x0018

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_SetCustomerKeyEx

### Description

Permit to set the Customer key

#### Parameters:

[Query frame layout] **bKey** : Customer key (only writable on the module) [32 bytes containing a AES key]

[Query frame layout] **bPublicKey** : IV (Initialisation vector) for the AES cryptography [16 bytes containing a AES key]

#### Returns:

Possible return value on the remote system (read them with `GetLastCommandStatusEx`)

- ◆ 0 : means the remote function performed OK
- ◆ -1: means an system error occurred (probably EPERM)

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
-------	--------------	------	-------	-----------------------	-----------------------

## MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	55	0x3700	0x0037
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11200	0xC02B	0x2BC0
word count	2	16-bit integer	24	0x1800	0x0018
byte count	1	8-bit integer	48	0x30	0x30
bKey	32	8-bit integer array	See the description above	0x??[32]	0x??[32]
bPublicKey	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
	2		11200	0xC02B	0x2BC0

### Query frame layout



## MODBUS interface description

Reference number (=register)		16-bit integer			
word count	2	16-bit integer	24	0x1800	0x0018

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_SetFilterChannels

For new application(s) or automate communication it is recommended to use the function MXCommon\_\_SetFilterChannelsEx.

### Description

Permit to set a filter per channel

#### Parameters:

[Query frame layout] **ChannelList** : Each index of the array is representing a channel. To set a filter on a channel, enter the filter ID. By default the value ist 0 (No filter).

#### Returns:

Possible return value on the remote system (read them with GetLastCommandStatus)

- ◆ 0 : means the remote function performed OK
- ◆ -1: means a system error occurred (probably EPERM)

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	24	0x1800	0x0018
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	105	0x6900	0x0069
word count	2	16-bit integer	8	0x0800	0x0008
byte count	2	16-bit integer	16	0x1000	0x0010
ChannelList	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS	1	8-bit	0x10	0x10	0x10

## MODBUS interface description

Function code		integer			
Reference number (=register)	2	16-bit integer	105	0x6900	0x0069
word count	2	16-bit integer	8	0x0800	0x0008

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MXCommon\_\_SetFilterChannelsEx

### Description

Permit to set a filter per channel

#### Parameters:

[Query frame layout] **Channellist** : Each index of the array is representing a channel. To set a filter on a channel, enter the filter ID. By default the value is 0 (No filter).

#### Returns:

**Possible return value on the remote system (read them with GetLastErrorStatusEx)**

- ◆ 0 : means the remote function performed OK
- ◆ -1: means a system error occurred (probably EPERM)

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11250	0xF22B	0x2BF2
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ChannelList	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS	1	8-bit	0x10	0x10	0x10

## MODBUS interface description

Function code		integer			
Reference number (=register)	2	16-bit integer	11250	0xF22B	0x2BF2
word count	2	16-bit integer	8	0x0800	0x0008

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFCommonSetInputsFilter

### Description

Set a filter to the input of a multifunction sub module.

#### Parameters:

[Query frame layout] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3)

[Query frame layout] **ulInputAFilterValue** : Filter value for input A (0 to 262143).

- ◆ 0: Filter nicht benutzt
- ◆ 1: 100 ns
- ◆ 2: 200 ns
- ◆ 3: 300 ns ...
- ◆ 262143 : 26,2143 ms

[Query frame layout] **ulInputBFilterValue** : Filter value for input B (0 to 262143).

- ◆ 0: Filter nicht benutzt

### Response frame layout

## MODBUS interface description

- ◆ 1: 100 ns
- ◆ 2: 200 ns
- ◆ 3: 300 ns ...
- ◆ 262143 : 26,2143 ms

[Query frame layout] **ullInputCFilterValue** : Filter value for input C (0 to 262143).

- ◆ 0: Filter nicht benutzt
- ◆ 1: 100 ns
- ◆ 2: 200 ns
- ◆ 3: 300 ns ...
- ◆ 262143 : 26,2143 ms

[Query frame layout] **ullInputDFilterValue** : Filter value for input D (0 to 262143).

- ◆ 0: Filter nicht benutzt
- ◆ 1: 100 ns
- ◆ 2: 200 ns
- ◆ 3: 300 ns ...
- ◆ 262143 : 26,2143 ms

### Returns:

#### • Possible return value on the remote system (read them with GetLastCommandStatusEx):

- ◆ 0 : No error.
- ◆ -1 : Means an system error occurred
- ◆ -2 : Multifunction sub module index selection error
- ◆ -3 : Input A filter value selection error
- ◆ -4 : Input B filter value selection error
- ◆ -5 : Input C filter value selection error
- ◆ -6 : Input D filter value selection error
- ◆ -100 : Kernel function error (see syserrno).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	27	0x1B00	0x001B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
	2		6000	0x7017	0x1770

## MODBUS interface description

Reference number (=register)		16-bit integer			
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulInputAFilterValue	4	32-bit integer	See the description above	0x????????	0x????????
ulInputBFilterValue	4	32-bit integer	See the description above	0x????????	0x????????
ulInputCFilterValue	4	32-bit integer	See the description above	0x????????	0x????????
ulInputDFilterValue	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6000	0x7017	0x1770
word count	2	16-bit integer	10	0x0A00	0x000A

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function

### MSXE17xx\_\_MFCommonReferenceVoltageActivation

## Description

Permit to activate the reference voltage to pin D-

### Parameters:

[Query frame layout] ***ulMFModuleIndex*** : index of the multifunction sub module (0 to 3)

[Query frame layout] ***ulActivationFlag*** : Filter value for input A (0 to 262143).

◆ 0: normal mode from D- (Default mode)

◆ 1: activate the reference voltage to pin D-

[Query frame layout] ***ulOption01*** : Reserved. Set it to 0.

[Query frame layout] ***ulOption02*** : Reserved. Set it to 0.

### Returns:

#### • Possible return value on the remote system (read them with GetLastCommandStatusEx):

◆ 0 : No error.

◆ -1 : Means an system error occurred

◆ -2 : Multifunction sub module index selection error

◆ -3 : Activation flag selection error



## MODBUS interface description

◆ -100 : Kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6050	0xA217	0x17A2
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulActivationFlag	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually	0x0000	0x0000

Description

## MODBUS interface description

			0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6050	0xA217	0x17A2
word count	2	16-bit integer	8	0x0800	0x0008

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFCommonSetFIFO0Level

### Description

Define the number of data bloc in the first FIFO before transmit the datas

### Parameters:

[Query frame layout] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3)

[Query frame layout] **ulFIFOLevel** : Define the FIFO level (1 to 200).

Response frame layout

## MODBUS interface description

[Query frame layout] **ulTimeOutTimeBase** : Define a Time out : permit to receive the data from the FIFO before the FIFO level is reached.

Time base of the timer (0: disabled, 1 for us, 2 for ms, 3 for s)

[Query frame layout] **ulReloadValue** : Time out reload value (1 to 0xFFFF)

[Query frame layout] **ulOption01** : Reserved. Set it to 0.

[Query frame layout] **ulOption02** : Reserved. Set it to 0.

### Returns:

- Possible return value on the remote system (read them with GetLastCommandStatusEx):

- ◆ 0 : No error.
- ◆ -1 : Means an system error occurred
- ◆ -2 : Multifunction sub module index selection error
- ◆ -3 : FIFO level value is wrong
- ◆ -4 : Time out time base selection error
- ◆ -5 : Time out value can not be null, if a time base is selected
- ◆ -100 : Kernel function error (see syserrno).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6100	0xD417	0x17D4
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulFIFOLevel	4			0x???????	0x???????

## MODBUS interface description

		32-bit integer	See the description above		
ulTimeOutTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulReloadValue	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6100	0xD417	0x17D4
word count	2	16-bit integer	12	0x0C00	0x000C

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server -	0x0000	0x0000

## Query frame layout

## MODBUS interface description

			usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFCommonEnableDisableTriggerGate

### Description

Define the number of data bloc in the first FIFO before transmit the datas

#### Parameters:

[Query frame layout] ***ulTriggerConfiguration*** : Trigger configuration:

- ◆ Bit 0: Hardware trigger
  - ◇ 0 : Disable hardware trigger gate
  - ◇ 1 : Enable hardware trigger gate

[Query frame layout] ***ulOption01*** : Reserved. Set it to 0.

[Query frame layout] ***ulOption02*** : Reserved. Set it to 0.

#### Returns:

- Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
  - ◆ 0 : No error.
  - ◆ -1 : Means an system error occurred
  - ◆ -2 : **ulTriggerConfiguration** parameter is wrong
  - ◆ -100 : Kernel function error (see **syserrno**).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by	0x0000	0x0000

Exception frame layout

## MODBUS interface description

			server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6150	0x0618	0x1806
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulTriggerConfiguration	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number	2	16-bit integer	6150	0x0618	0x1806

## Query frame layout

## MODBUS interface description

(=register)					
word count	2	16-bit integer	6	0x0600	0x0006

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosInit

### Description

Initialize the selected Sinus / Cosinus module.

#### Parameters:

[Query frame layout] **ulModuleIndex** : The module to initialize (0 to 3).

[Query frame layout] **ulMode** : Measure mode

- ◆ 0 : Fast measure. The measure is fast, always 250 kHz but the measure range is smaller.
- ◆ 1 : Full range. The measure is slow but the maximal range is used.  
See in the table called  
"Max. input frequency in corresponding with the input ulResolution"  
in order to set the measure frequency.

[Query frame layout] **ulSignalPeriod** : Signal period.

[Query frame layout] **ulResolution** : Reslution to use for the measure (binary value)

**Max. input frequency in corresponding with the input ulResolution:**

## MODBUS interface description

Resolution	Max. Freq. Hz	Compatible with
16	250000	fast mode and full range mode
25	26000	fast mode
32	162500	fast mode and full range mode
40	16300	fast mode and full range mode
50	26000	fast mode
64	81300	fast mode and full range mode
80	16300	fast mode and full range mode
100	26000	fast mode and full range mode
125	20800	fast mode
128	40600	fast mode and full range mode
160	16300	fast mode and full range mode
200	26000	fast mode and full range mode
250	20800	fast mode
256	20300	fast mode and full range mode
320	16300	fast mode and full range mode
400	13000	fast mode and full range mode
500	10400	fast mode and full range mode
512	10200	fast mode and full range mode
800	6500	fast mode and full range mode
1000	5200	fast mode and full range mode
1024	5100	fast mode and full range mode
1600	3300	fast mode and full range mode
2000	2600	fast mode and full range mode
2048	2540	fast mode and full range mode
4096	1270	fast mode and full range mode
8192	635	fast mode and full range mode

[Query frame layout] ***ulValueFormat*** : Output format of the measure

- ◆ 0 : Raw data.
- ◆ 1 : Standardized in mm.

[Query frame layout] ***ulOption01*** : Reserved. Set it to 0.

[Query frame layout] ***ulOption02*** : Reserved. Set it to 0.

[Query frame layout] ***ulOption03*** : Reserved. Set it to 0.

[Query frame layout] ***ulOption04*** : Reserved. Set it to 0.

### Returns:

- **Possible return value on the remote system (read them with `GetLastCommandStatusEx`):**
  - ◆ 0 : No error.
  - ◆ -1 : means an system error occurred
  - ◆ -2 : Multifunction sub module index selection error.



## MODBUS interface description

- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Wrong mode.
- ◆ -5 : Wrong signal period.
- ◆ -6 : Wrong resolution.
- ◆ -7 : The resolution is not supported by the selected mode.
- ◆ -8 : Wrong format.
- ◆ -9 : Auto gain calibration error.
- ◆ -100 : Kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	43	0x2B00	0x002B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1250	0xE204	0x04E2
word count	2	16-bit integer	18	0x1200	0x0012
byte count	1	8-bit integer	36	0x24	0x24
ulModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulMode	4	32-bit integer	See the description above	0x????????	0x????????
ulSignalPeriod	4	32-bit integer	See the description above	0x????????	0x????????
ulResolution	4	32-bit integer	See the description above	0x????????	0x????????
ulValueFormat	4	32-bit integer	See the description above	0x????????	0x????????

## MODBUS interface description

ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????
ulOption03	4	32-bit integer	See the description above	0x????????	0x????????
ulOption04	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1250	0xE204	0x04E2
word count	2	16-bit integer	18	0x1200	0x0012

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000

## MODBUS interface description

length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosClear

### Description

Clear the selected Sinus / Cosinus module.

### Parameters:

[Query frame layout] **ulModuleIndex** : The module to clear (0 to 3).

[Query frame layout] **ulOption01** : Reserved. Set to 0.

[Query frame layout] **ulOption02** : Reserved. Set to 0.

### Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Sinus / Cosinus module not initialised
- ◆ -100 : Clear Sinus Cosinus kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013

Exception frame layout

## MODBUS interface description

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1300	0x1405	0x0514
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1300	0x1405	0x0514
word count	2	16-bit integer	6	0x0600	0x0006

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosRelease

### Description

Release the selected Sinus / Cosinus module.

#### Parameters:

[Query frame layout] ***ulModuleIndex*** : The module to clear (0 to 3).

[Query frame layout] ***ulOption01*** : Reserved. Set to 0.

[Query frame layout] ***ulOption02*** : Reserved. Set to 0.

#### Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - ◆ 0 : No error.
  - ◆ -1 : means an system error occurred
  - ◆ -2 : Multifunction sub module index selection error.
  - ◆ -3 : Multifunction sub module is not a SinCos module.
  - ◆ -4 : Sinus / Cosinus module not initialised
  - ◆ -100 : Release Sinus Cosinus kernel function error (see syserrno).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1350	0x4605	0x0546
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000

## MODBUS interface description

length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1350	0x4605	0x0546
word count	2	16-bit integer	6	0x0600	0x0006

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosInitAndEnableLatch

### Description

Init and enable a counter latch logic

For each latch the data server send a 5 DWORD frame with following informations:

- DWORD 0 : Time stamp micro s
- DWORD 1 : Time stamp s
- DWORD 2 :
  - ◆ D1-D0 : Sub module index (0 to 3)
  - ◆ D31-D16 : Sub module functionality (2)
- DWORD 3 : Event mask

Response frame layout

## MODBUS interface description

- ◆ D30-D0 :
  - 2: Hardware trigger latch occur
  - 3: Synchro input latch occur
  - 4: Index input latch occur
- ◆ D31 :
  - 0: No error occur
  - 1: Amplitude or Frequency error occur.
- DWORD 4 :
  - ◆ D31-D0: Counter value (DWORD) if the selected output format of the measure is raw data
  - ◆ D31-D0: Current position in mm (FLOAT) if the selected output format of the measure is standardized in mm.

### Parameters:

[Query frame layout] **ulMFModuleIndex** : Index of the multifunction sub module (0 to 3).

[Query frame layout] **ulLatchSource** : Latch source.

- ◆ 0: Index input
- ◆ 1: Hardware trigger
- ◆ 2: Synchro input

[Query frame layout] **ulCondition** : Previously condition for accept the latch source

- ◆ 0: No previously condition required
- ◆ 1: Index input condition required (Only if index input not selected selected for the latch source)
- ◆ 2: Hardware trigger condition required (Only if hardware trigger not selected selected for the latch source)
- ◆ 3: Synchro input condition required (Only if synchro input not selected selected for the latch source)

[Query frame layout] **ulAutoMode** : Action mode

- ◆ 0: Do not use auto mode (action is done only once)
- ◆ 1: Use auto mode (action is done continuously)

[Query frame layout] **ulOption02** : set it to 0

### Returns:

- **Possible return value on the remote system (read them with GetLastErrorStatusEx):**
  - ◆ 0: means the remote function performed OK
  - ◆ -1: means an system error occurred
  - ◆ -2: Multifunction sub module index selection error
  - ◆ -3: Multifunction sub module is not a SinCos module.
  - ◆ -4: Sinus / Cosinus module not initialised
  - ◆ -5: Latch logic already initialised
  - ◆ -6: Latch source selection error
  - ◆ -7: Previously condition selection error
  - ◆ -8: Auto mode selection error
  - ◆ -9: Hardware trigger not initialised. Refer to MSXE17xx\_\_MFSinCosInitHardwareTrigger
  - ◆ -10: Index input not initialised. Refer to MSXE17xx\_\_MFSinCosInitIndex
  - ◆ -11: Can not be used for the "fast measure" mode



## MODBUS interface description

◆ -100: Init and enable counter latch kernel function error

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1400	0x7805	0x0578
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulLatchSource	4	32-bit integer	See the description above	0x????????	0x????????
ulCondition	4	32-bit integer	See the description above	0x????????	0x????????
ulAutoMode	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian	big endian (Motorola)
-------	--------------	------	-------	---------------	-----------------------

## MODBUS interface description

				(Intel)	
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1400	0x7805	0x0578
word count	2	16-bit integer	12	0x0C00	0x000C

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosDisableAndReleaseLatch

## Description

Disable and Release a counter latch logic

### Parameters:

[Query frame layout] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout] **ulLatchSource** : Latch source to disable and release.

- ◆ 0: Index input
- ◆ 1: Hardware trigger
- ◆ 2: Synchro input

[Query frame layout] **ulOption01** : set it to 0

### Returns:

• **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Multifunction sub module index selection error
- ◆ -3: Multifunction sub module is not a SinCos module.
- ◆ -4: Sinus / Cosinus module not initialised
- ◆ -5: Latch logic not initialised
- ◆ -6: Latch source selection error
- ◆ -100: Disable and release counter latch register kernel function error

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1450	0xAA05	0x05AA
word count	2	16-bit integer	6	0x0600	0x0006

## MODBUS interface description

byte count	1	8-bit integer	12	0x0C	0x0C
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulLatchSource	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1450	0xAA05	0x05AA
word count	2	16-bit integer	6	0x0600	0x0006

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000

## MODBUS interface description

length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosInitAndEnableClear

### Description

Init and enable a counter clear logic

#### Parameters:

[Query frame layout] **ulMFModuleIndex** : Index of the multifunction sub module (0 to 3).

[Query frame layout] **ulClearSource** : Clear source.

- ◆ 0: Index input
- ◆ 1: Hardware trigger
- ◆ 2: Synchro input

[Query frame layout] **ulCondition** : Previously condition for accept the clear source

- ◆ 0: No previously condition required
- ◆ 1: Index input condition required (Only if index input not selected selected for the clear source)
- ◆ 2: Hardware trigger condition required (Only if hardware trigger not selected selected for the clear source)
- ◆ 3: Synchro input condition required (Only if synchro input not selected selected for the clear source)

[Query frame layout] **ulAutoMode** : Action mode

- ◆ 0: Do not use auto mode (action is done only once)
- ◆ 1: Use auto mode (action is done continuously)

[Query frame layout] **ulOption01** : set it to 0

[Query frame layout] **ulOption02** : set it to 0

#### Returns:

- **Possible return value on the remote system (read them with GetLastErrorStatusEx):**
  - ◆ 0: means the remote function performed OK
  - ◆ -1: means an system error occurred
  - ◆ -2: Multifunction sub module index selection error

## MODBUS interface description

- ◆ -3: Multifunction sub module is not a SinCos module.
- ◆ -4: Sinus / Cosinus module not initialised
- ◆ -5: Clear logic already initialised
- ◆ -6: Clear source selection error
- ◆ -7: Previously condition selection error
- ◆ -8: Auto mode selection error
- ◆ -9: Hardware trigger not initialised. Refer to MSXE17xx\_\_MFSinCosInitHardwareTrigger
- ◆ -10: Index input not initialised. Refer to MSXE17xx\_\_MFSinCosInitIndex
- ◆ -11: Can not be used for the "fast measure" mode
- ◆ -100: Init and enable counter clear kernel function error

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1500	0xDC05	0x05DC
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulClearSource	4	32-bit integer	See the description above	0x???????	0x???????
ulCondition	4	32-bit integer	See the description above	0x???????	0x???????
ulAutoMode	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description	0x???????	0x???????

## MODBUS interface description

			above		
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1500	0xDC05	0x05DC
word count	2	16-bit integer	12	0x0C00	0x000C

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90

Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??
----------------	---	---------------	---------------------------	------	------

## Function MSXE17xx\_\_MFSinCosDisableAndReleaseClear

### Description

Disable and Release a counter clear logic

#### Parameters:

[Query frame layout] ***ulMFModuleIndex*** : index of the multifunction sub module (0 to 3).

[Query frame layout] ***ulClearSource*** : Clear source to disable and release.

- ◆ 0: Index input
- ◆ 1: Hardware trigger
- ◆ 2: Synchro input

[Query frame layout] ***ulOption01*** : set it to 0

#### Returns:

• **Possible return value on the remote system (read them with `GetLastCommandStatusEx`):**

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Multifunction sub module index selection error
- ◆ -3: Multifunction sub module is not a SinCos module.
- ◆ -4: Sinus / Cosinus module not initialised
- ◆ -5: Clear logic not initialised
- ◆ -6: Clear source selection error
- ◆ -100: Disable and release counter clear register kernel function error

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01



## MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1550	0x0E06	0x060E
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulClearSource	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1550	0x0E06	0x060E
word count	2	16-bit integer	6	0x0600	0x0006

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
-------	--------------	------	-------	-----------------------	-----------------------

## MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function

### MSXE17xx\_\_MFSinCosInitAndEnableCompareLogic

## Description

Init and enable a counter compare value

For each latch the data server send a 5 DWORD frame with following informations:

- DWORD 0 : Time stamp micro s
- DWORD 1 : Time stamp s
- DWORD 2 :
  - ◆ D1-D0 : Sub module index (0 to 3)
  - ◆ D31-D16 : Sub module functionality (2)
- DWORD 3 : Event mask
  - ◆ D30-D0 :
    - 0: Compare occur
  - ◆ D31 :
    - 0: No error occur
    - 1: Amplitude or Frequency error occur.
- DWORD 4 :
  - ◆ D31-D0: Counter value (DWORD) if the selected output format of the measure is raw data
  - ◆ D31-D0: Current position in mm (FLOAT) if the selected output format of the measure is standardized in mm.

## Parameters:

[Query frame layout] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout] **ulValueLow / ulValueLow** : compare value (double). ulValueLow : low 32-bit from double. ulValueHigh : high 32-bit from double

## MODBUS interface description

- ◆ 0 to 0xFFFFFFFF if raw data selected
- ◆ Position in mm if the measure is standardized in mm

[Query frame layout] **ulMode** : compare mode

- ◆ 0: condition true when counter equals compare value
- ◆ 1: condition true when counter equals a multiple of the compare value

[Query frame layout] **ulSynchroTrigger**

- ◆ 0 : no synchro trigger
- ◆ 1 : generates a synchro trigger when condition is true

[Query frame layout] **ulOption02** : set it to 0

### Returns:

#### • Possible return value on the remote system (read them with **GetLastCommandStatusEx**):

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Multifunction sub module index selection error
- ◆ -3: Compare value error
- ◆ -4: Compare mode error
- ◆ -5: Synchro trigger error
- ◆ -6: Multifunction sub module is not a SinCos module.
- ◆ -7: Sinus / Cosinus module not initialised
- ◆ -8: Compare logic already initialised
- ◆ -9: Can not be used for the "fast measure" mode
- ◆ -100: Init and enable counter compare kernel function error

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	35	0x2300	0x0023
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1600	0x4006	0x0640
word count	2	16-bit integer	14	0x0E00	0x000E

## MODBUS interface description

byte count	1	8-bit integer	28	0x1C	0x1C
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulValueLow	4	32-bit integer	See the description above	0x????????	0x????????
ulValueHigh	4	32-bit integer	See the description above	0x????????	0x????????
ulMode	4	32-bit integer	See the description above	0x????????	0x????????
ulSynchroTrigger	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1600	0x4006	0x0640
word count	2	16-bit integer	14	0x0E00	0x000E

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function

### MSXE17xx\_\_MFSinCosDisableAndReleaseCompareLogic

## Description

Disable and Release a counter compare value

### Parameters:

[Query frame layout] ***uIMFModuleIndex*** : index of the multifunction sub module (0 to 3).

### Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - ◆ 0: means the remote function performed OK
  - ◆ -1: means an system error occurred
  - ◆ -2: Multifunction sub module index selection error
  - ◆ -3: Multifunction sub module is not a SinCos module.
  - ◆ -4: Sinus / Cosinus module not initialised
  - ◆ -5: Compare logic not initialised
  - ◆ -100: Disable counter compare value kernel function error

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian	big endian (Motorola)
-------	--------------	------	-------	---------------	-----------------------

## MODBUS interface description

				(Intel)	
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1650	0x7206	0x0672
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1650	0x7206	0x0672
word count	2	16-bit	2	0x0200	0x0002

		integer			
--	--	---------	--	--	--

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosInitHardwareTrigger

### Description

Init the hardware trigger configuration

#### Parameters:

[Query frame layout] ***ulMFModuleIndex*** : index of the multifunction sub module (0 to 3).

[Query frame layout] ***ulEdgeSelection*** : Front selection

- ◆ 01 : rising front
- ◆ 10 : falling front
- ◆ 11 : Both front

[Query frame layout] ***ulCount*** : Define the number of trigger events before the action occur

- ◆ 1 : all trigger event start the action
- ◆ max value : 65535

[Query frame layout] ***ulOption01*** : Hardware trigger gate

- ◆ 0 : Don't use hardware trigger gate
- ◆ 1 : Use hardware trigger gate

[Query frame layout] ***ulOption02*** : set it to 0

#### Returns:

Response frame layout

## MODBUS interface description

### • Possible return value on the remote system (read them with `GetLastCommandStatusEx`):

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Multifunction sub module index selection error
- ◆ -3: Multifunction sub module is not a SinCos module.
- ◆ -4: Trigger edge selection error
- ◆ -5: Trigger count selection error
- ◆ -6: Sinus / Cosinus module not initialised
- ◆ -7: Hardware trigger already initialised
- ◆ -100: Init hardware trigger kernel function error

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	27	0x1B00	0x001B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1700	0xA406	0x06A4
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulEdgeSelection	4	32-bit integer	See the description above	0x???????	0x???????
ulCount	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????
ulOption02	4	32-bit integer	See the description	0x???????	0x???????



			above		
--	--	--	-------	--	--

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1700	0xA406	0x06A4
word count	2	16-bit integer	10	0x0A00	0x000A

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

# Function MSXE17xx\_\_MFSinCosReleaseHardwareTrigger

## Description

Release the hardware trigger

### Parameters:

[Query frame layout] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout] **ulOption01** : set it to 0

### Returns:

• **Possible return value on the remote system (read them with GetLastCommandStatusEx):**

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Multifunction sub module index selection error
- ◆ -3: Multifunction sub module is not a SinCos module.
- ◆ -4: Sinus / Cosinus module not initialised
- ◆ -5: Hardware trigger not initialised
- ◆ -6: Hardware trigger used and can not released
- ◆ -100: Release hardware trigger kernel function error

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1750	0xD606	0x06D6
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulMFModuleIndex	4			0x???????	0x???????

## MODBUS interface description

		32-bit integer	See the description above		
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1750	0xD606	0x06D6
word count	2	16-bit integer	4	0x0400	0x0004

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS	1	8-bit	0x90	0x90	0x90

### Query frame layout

## MODBUS interface description

Function code		integer			
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosInitIndex

### Description

Init the index configuration

### Parameters:

[Query frame layout] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout] **ulEdgeSelection** : Front selection

- ◆ 01 : rising front
- ◆ 10 : falling front
- ◆ 11 : Both front

[Query frame layout] **ulOption01** : set it to 0

[Query frame layout] **ulOption02** : set it to 0

### Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - ◆ 0: means the remote function performed OK
  - ◆ -1: means an system error occurred
  - ◆ -2: Multifunction sub module index selection error
  - ◆ -3: Multifunction sub module is not a SinCos module.
  - ◆ -4: Index edge selection error
  - ◆ -5: Sinus / Cosinus module not initialised
  - ◆ -6: Index already initialised

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2		23	0x1700	0x0017

Exception frame layout

## MODBUS interface description

		16-bit integer			
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1800	0x0807	0x0708
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulEdgeSelection	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1800	0x0807	0x0708

## MODBUS interface description

word count	2	16-bit integer	8	0x0800	0x0008
------------	---	----------------	---	--------	--------

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosReleaseIndex

### Description

Release the index

#### Parameters:

[Query frame layout] ***ulMFModuleIndex*** : index of the multifunction sub module (0 to 3).

[Query frame layout] ***ulOption01*** : set it to 0

#### Returns:

#### • Possible return value on the remote system (read them with GetLastCommandStatusEx):

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Multifunction sub module index selection error
- ◆ -3: Multifunction sub module is not a SinCos module.
- ◆ -4: Sinus / Cosinus module not initialised
- ◆ -5: Index not initialised
- ◆ -6: Index used and can not released
- ◆ -100: Release Index kernel function error

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1850	0x3A07	0x073A
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01

## MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1850	0x3A07	0x073A
word count	2	16-bit integer	4	0x0400	0x0004

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_MFSinCosInitAbsPos

### Description

Initialize the selected Sinus / Cosinus absolute position module.

#### Parameters:

[Query frame layout] ***ulModuleIndex*** : The module to initialize (0 to 3).

[Query frame layout] ***ulSignalPeriod*** : Signal period.

[Query frame layout] ***ulNominalIncrement*** : Nominal incremenet in signal periods.

[Query frame layout] ***ulResolution*** : Reslution to use for the measure (binary value)

**Max. input frequency in corresponding with the input ulResolution:**



## MODBUS interface description

Resolution	Max. Freq. Hz	Compatible with
16	250000	fast mode and full range mode
25	26000	fast mode
32	162500	fast mode and full range mode
40	16300	fast mode and full range mode
50	26000	fast mode
64	81300	fast mode and full range mode
80	16300	fast mode and full range mode
100	26000	fast mode and full range mode
125	20800	fast mode
128	40600	fast mode and full range mode
160	16300	fast mode and full range mode
200	26000	fast mode and full range mode
250	20800	fast mode
256	20300	fast mode and full range mode
320	16300	fast mode and full range mode
400	13000	fast mode and full range mode
500	10400	fast mode and full range mode
512	10200	fast mode and full range mode
800	6500	fast mode and full range mode
1000	5200	fast mode and full range mode
1024	5100	fast mode and full range mode
1600	3300	fast mode and full range mode
2000	2600	fast mode and full range mode
2048	2540	fast mode and full range mode
4096	1270	fast mode and full range mode
8192	635	fast mode and full range mode

[Query frame layout] ***ulValueFormat*** : Output format of the measure

- ◆ 0 : Raw data.
- ◆ 1 : Standardized in mm.

[Query frame layout] ***ulInvert*** : Direction inversion

- ◆ 0 : No inversion.
- ◆ 1 : Invert the direction.

[Query frame layout] ***ulOption01*** : Reserved. Set it to 0.

[Query frame layout] ***ulOption02*** : Reserved. Set it to 0.

[Query frame layout] ***ulOption03*** : Reserved. Set it to 0.

[Query frame layout] ***ulOption04*** : Reserved. Set it to 0.

**Returns:**

## MODBUS interface description

### • Possible return value on the remote system (read them with GetLastCommandStatusEx):

- ◆ 0 : No error.
- ◆ -1 : means an system error occurred
- ◆ -2 : Multifunction sub module index selection error.
- ◆ -3 : Multifunction sub module is not a SinCos module.
- ◆ -4 : Wrong nominal increment.
- ◆ -5 : Wrong signal period.
- ◆ -6 : Wrong resolution.
- ◆ -7 : The resolution is not supported by the selected mode.
- ◆ -8 : Wrong format.
- ◆ -9 : Auto gain calibration error.
- ◆ -10 : Index already initialised.
- ◆ -11 : Enable latch initialisation error.
- ◆ -100 : Kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	47	0x2F00	0x002F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1900	0x6C07	0x076C
word count	2	16-bit integer	20	0x1400	0x0014
byte count	1	8-bit integer	40	0x28	0x28
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulSignalPeriod	4	32-bit integer	See the description above	0x????????	0x????????
ulNominalIncrement	4	32-bit integer	See the description above	0x????????	0x????????
ulResolution	4			0x????????	0x????????

## MODBUS interface description

		32-bit integer	See the description above		
ulValueFormat	4	32-bit integer	See the description above	0x????????	0x????????
ulInvert	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????
ulOption03	4	32-bit integer	See the description above	0x????????	0x????????
ulOption04	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1900	0x6C07	0x076C
word count	2	16-bit integer	20	0x1400	0x0014

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_DigitalIOWriteAllChannelsValue

### Description

Write all digital i/o channels value. if the channel is define as input, nothing append on this channel.

#### Parameters:

[Query frame layout]**ulValue** : Channels value

#### Returns:

- Possible return value on the remote system (read them with `GetLastCommandStatusEx`):
  - ◆ 0: means the remote function performed OK
  - ◆ -1: means an system error occurred
  - ◆ -100: Write digital I/O kernel function error

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server -	0x0000	0x0000

## MODBUS interface description

			usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7100	0xBC1B	0x1BBC
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7100	0xBC1B	0x1BBC
word count	2	16-bit integer	2	0x0200	0x0002

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_DigitalIORearmShortCircuit

### Description

Rearm digital outputs after a short circuit happened.

#### Parameters:

[Query frame layout] **ulOption** : Reserved. Set to 0.

#### Returns:

- Possible return value on the remote system (read them with `GetLastCommandStatusEx`):
  - ◆ 0 : No error.
  - ◆ -1 : means an system error occurred
  - ◆ -100 : Kernel function error (see `syserrno`).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server -	0x0000	0x0000

## MODBUS interface description

			usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7150	0xEE1B	0x1BEE
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption	4	32-bit integer	See the description above	0x???????	0x???????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7150	0xEE1B	0x1BEE
word count	2	16-bit integer	2	0x0200	0x0002

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_DigitalIOInitPort

### Description

Initialise a digital i/o port (2 channels).

#### Parameters:

[Query frame layout] ***ulPort*** : Index of the digital i/o port (0 to 7).

[Query frame layout] ***ulPortConfiguration*** : Define the port configuration

- ◆ 0 : input
- ◆ 1 : output

#### Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - ◆ 0 : No error.
  - ◆ -1 : means an system error occurred
  - ◆ -2: Digital i/o port selection error
  - ◆ -3: Port configuration selection error
  - ◆ -100 : Kernel function error (see syserrno).



## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7200	0x201C	0x1C20
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulPort	4	32-bit integer	See the description above	0x???????	0x???????
ulPortConfiguration	4	32-bit integer	See the description above	0x???????	0x???????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01

## MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7200	0x201C	0x1C20
word count	2	16-bit integer	4	0x0400	0x0004

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_IOWatchdogInitAndStart

### Description

Init and start the digital output IO watchdog.

#### Parameters:

[Query frame layout] **ulTimeBase** : Time base of the watchdog delay (0 for mus, 1 for ms, 2 for s)

[Query frame layout] **ulTimeValue** : Time base of the watchdog delay (0 to 0xFFFF).

[Query frame layout] **ulOption01** : Reserved. Set to 0.

[Query frame layout] **ulOption02** : Reserved. Set to 0.

#### Returns:

- Possible return value on the remote system (read them with `GetLastCommandStatusEx`):

Response frame layout

## MODBUS interface description

- ◆ 0 : No error.
- ◆ -1 : Means an system error occured
- ◆ -2:: Time base selection error
- ◆ -3:: Time value selection error
- ◆ -100 : Kernel function error (see syserrno).

### Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	8050	0x721F	0x1F72
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulTimeValue	4	32-bit integer	See the description above	0x????????	0x????????
ulOption1	4	32-bit integer	See the description above	0x????????	0x????????
ulOption2	4	32-bit integer	See the description above	0x????????	0x????????

### Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
-------	--------------	------	-------	-----------------------	-----------------------

Description

## MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	8050	0x721F	0x1F72
word count	2	16-bit integer	8	0x0800	0x0008

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE17xx\_\_IOWatchdogStopAndRelease

## Description

Stop and release the digital output watchdog.

### Parameters:

[Query frame layout] **ulOption** : Reserved. Set to 0.

### Returns:

- Possible return value on the remote system (read them with `GetLastCommandStatusEx`):
  - ◆ 0 : No error.
  - ◆ -1 : Means an system error occurred
  - ◆ -100 : Kernel function error (see `syserrno`).

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	8100	0xA41F	0x1FA4
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption	4	32-bit integer	See the description above	0x????????	0x????????

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
-------	--------------	------	-------	-----------------------	-----------------------

## MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	8100	0xA41F	0x1FA4
word count	2	16-bit integer	2	0x0200	0x0002

### Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

# FC23 (read/write registers) Functions

[Top](#)

Functions in this group are used to read/write values on the module.  
This functions permits to call a write (FC16) and then a read(FC3) function in one command.

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Motorola)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	Depends to the FC16 function called	?	?
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x17	0x17	0x17
Reference number for read (=register)	2	16-bit integer	FC3 reference	?	?
Word count for read	2	16-bit integer	See the corresponding FC3 function	?	?
Reference number for write (=register)	2	16-bit integer	FC16 reference	?	?
Word count for write	2	16-bit integer	See the corresponding FC16 function	?	?
Byte count	1	8-bit integer	(= 2xWord count for write)	?	?
Register values	?	?	See the corresponding FC16 function	?	?

## Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Motorola)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	Depends to the FC3 function called	?	?
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x17	0x17	0x17
Byte count	1	8-bit integer	(= 2x word count for read)	?	?
Register values	?	?	See the corresponding FC3 function	?	?

## Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Motorola)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	3	0x0300	0x0003
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x97	0x97	0x97
Exception code	1	8-bit integer	See corresponding chapter	??	??



# Exception code description

[Top](#)

Name	Value	Description
MODBUS_ILLEGAL_FUNCTION	0x1	function code is not allowable action for the slave
MODBUS_ILLEGAL_DATA_ADDRESS	0x2	data address received in query is not allowable
MODBUS_ILLEGAL_DATA_VALUE	0x3	incorrect value in the query data field or the length is incorrect
MODBUS_ILLEGAL_DATA_RESPONSE_LENGTH	0x4	the request as framed would generate a response whose size exceeds the available MODBUS datasize.
MODBUS_ACKNOWLEDGE	0x5	specialized use in conjunction with programming commands
MODBUS_DSLAVE_DEVICE_BUSY	0x6	specialized use in conjunction with programming commands
MODBUS_NEGATIVE_ACKNOWLEDGE	0x07	specialized use in conjunction with programming commands
MODBUS_MEMORY_PARITY_ERROR	0x08	the extended file area failed to pass a consistency check
MODBUS_REMOTE_EXECUTION_ERROR	0x09	the remote function performed incorrectly (use function GetLastCommandStatus to know why)
MODBUS_GATEWAY_PATH_UNAVAILABLE	0x0A	used with modbus plus gateway
MODBUS_GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND	0x0B	used with modbus plus gateway

# Siemens Step 7 compatibility information (AWL/SDF code)

[Top](#)

Due to limitations of the S7 platform, some names of function and parameter have been shortened in the AWL and S7 code. This table summarizes the changes against the standard version as described above.

Function/Parameter	Renamed as
MXCommon__GetModuleType	GetModuleType
MXCommon__GetTime	GetTime
MXCommon__TestCustomerID	TestCustomerID
MSXE17xx__MFSinCosReadAll	17xx_MFSinCosReadAll
MSXE17xx__MFSinCosRead0	17xx_MFSinCosRead0
MSXE17xx__MFSinCosRead1	17xx_MFSinCosRead1
MSXE17xx__MFSinCosRead2	17xx_MFSinCosRead2
MSXE17xx__MFSinCosRead3	17xx_MFSinCosRead3
MSXE17xx__MFSinCosReadAbsPos0	17xx_MFSinCosRdAbsPos0
MSXE17xx__MFSinCosReadAbsPos1	17xx_MFSinCosRdAbsPos1
MSXE17xx__MFSinCosReadAbsPos2	17xx_MFSinCosRdAbsPos2
MSXE17xx__MFSinCosReadAbsPos3	17xx_MFSinCosRdAbsPos3
MSXE17xx__MFSinCosReadAbsPosAll	17xx_MFSinCosRdAbsPos
MSXE17xx__DigitalIOReadAllChannelsValue	17xx_DigIOReadAll
MSXE17xx__DigitalIOTestShortCircuit	17xx_DigIOTestShortC
MSXE17xx__IOWatchdogGetStatusAndValue	17xx_IOWatchdogGet
MXCommon__SetHardwareTriggerFilterTime	SetHwTrigFiltTime
MXCommon__InitAndStartSynchroTimer	InitStartSyncTimer
MXCommon__StopAndReleaseSynchroTimer	StopRelSyncTimer
MXCommon__Reboot	Reboot
MXCommon__SetCustomerKey	SetCustomerKey
MXCommon__SetFilterChannels	SetFilterChannels
MSXE17xx__MFCCommonSetInputsFilter	17xx_MFCSetInputFilter
MSXE17xx__MFCCommonReferenceVoltageActivation	17xx_MFCRefVoltActiv
MSXE17xx__MFCCommonSetFIFO0Level	17xx_MFCSetFIFO0Level
MSXE17xx__MFCCommonEnableDisableTriggerGate	17xx_MFEDTriggerGate
MSXE17xx__MFSinCosInit	17xx_MFSinCosInit
MSXE17xx__MFSinCosClear	17xx_MFSinCosClear
MSXE17xx__MFSinCosRelease	17xx_MFSinCosRelease
MSXE17xx__MFSinCosInitAndEnableLatch	17xx_MFSinCosInitLatch
MSXE17xx__MFSinCosDisableAndReleaseLatch	17xx_MFSinCosRelLatch
MSXE17xx__MFSinCosInitAndEnableClear	17xx_MFSinCosInitClear
MSXE17xx__MFSinCosDisableAndReleaseClear	17xx_MFSinCosRelClear
MSXE17xx__MFSinCosInitAndEnableCompareLogic	17xx_MFSinCosInitCL
MSXE17xx__MFSinCosDisableAndReleaseCompareLogic	17xx_MFSinCosRelCL
MSXE17xx__MFSinCosInitHardwareTrigger	17xx_MFSinCosInitHT

## MODBUS interface description

MSXE17xx__MFSinCosReleaseHardwareTrigger	17xx_MFSinCosRelHT
MSXE17xx__MFSinCosInitIndex	17xx_MFSinCosInitIndex
MSXE17xx__MFSinCosReleaseIndex	17xx_MFSinCosRelIndex
MSXE17xx__MFSinCosInitAbsPos	17xx_MFSinCosInitAP
MSXE17xx__DigitalIOWriteAllChannelsValue	17xx_DigIOWriteAll
MSXE17xx__DigitalIORearmShortCircuit	17xx_DigIORearm
MSXE17xx__DigitalIOInitPort	17xx_DigIOInitPort
MSXE17xx__IOWatchdogInitAndStart	17xx_IOWatchdogStart
MSXE17xx__IOWatchdogStopAndRelease	17xx_IOWatchdogStop