

## **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>	9
<u>Exception frame layout.....</u>	9
<u>Function MXCommon GetModuleTypeEx.....</u>	10
<u>Description.....</u>	10
<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>	12
<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>	15
<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**

<u>Exception frame layout</u> .....	16
<u>Function MXCommon TestCustomerIDEx</u> .....	17
<u>Description</u> .....	17
<u>Query frame layout</u> .....	17
<u>Response frame layout</u> .....	18
<u>Exception frame layout</u> .....	18
<u>Function MSXE3317 PressureGetNumberOfChannels</u> .....	19
<u>Description</u> .....	19
<u>Query frame layout</u> .....	19
<u>Response frame layout</u> .....	20
<u>Exception frame layout</u> .....	20
<u>Function MSXE3317 PressureCalibrationGetCurrentStatus</u> .....	21
<u>Description</u> .....	21
<u>Query frame layout</u> .....	21
<u>Response frame layout</u> .....	22
<u>Exception frame layout</u> .....	22
<u>Function MSXE3317 AcquisitionGetNumberOfChannels</u> .....	23
<u>Description</u> .....	23
<u>Query frame layout</u> .....	23
<u>Response frame layout</u> .....	24
<u>Exception frame layout</u> .....	24
<u>Function MSXE3317 AcquisitionGetChannelsInfo</u> .....	25
<u>Description</u> .....	25
<u>Query frame layout</u> .....	25
<u>Response frame layout</u> .....	26
<u>Exception frame layout</u> .....	27
<u>Function MSXE3317 AcquisitionAutoRefreshGetValues</u> .....	27
<u>Description</u> .....	27
<u>Query frame layout</u> .....	28
<u>Response frame layout</u> .....	28
<u>Exception frame layout</u> .....	29
<u>Function MSXE3317 AcquisitionAutoRefreshGetBlockingValues</u> .....	29
<u>Description</u> .....	29
<u>Query frame layout</u> .....	30
<u>Response frame layout</u> .....	30
<u>Exception frame layout</u> .....	31
<u>Function MSXE3317 AcquisitionAutoRefreshGetConfiguration</u> .....	31
<u>Description</u> .....	32
<u>Query frame layout</u> .....	33
<u>Response frame layout</u> .....	33
<u>Exception frame layout</u> .....	35
<u>Function MSXE3317 AcquisitionSequenceGetConfiguration</u> .....	35
<u>Description</u> .....	35
<u>Query frame layout</u> .....	37
<u>Response frame layout</u> .....	37
<u>Exception frame layout</u> .....	38
<u>Function MSXE3317 DigitalIOGetNumberOfChannels</u> .....	39

# Table of Contents

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

<u>Description</u> .....	39
<u>Query frame layout</u> .....	39
<u>Response frame layout</u> .....	40
<u>Exception frame layout</u> .....	40
<u>Function MSXE3317 DigitalIOGetNumberOfPorts</u> .....	41
<u>Description</u> .....	41
<u>Query frame layout</u> .....	41
<u>Response frame layout</u> .....	41
<u>Exception frame layout</u> .....	42
<u>Function MSXE3317 DigitalIOGetPortAvailableDirections</u> .....	42
<u>Description</u> .....	42
<u>Query frame layout</u> .....	43
<u>Response frame layout</u> .....	43
<u>Exception frame layout</u> .....	44
<u>Function MSXE3317 DigitalIOGetPortDirections</u> .....	44
<u>Description</u> .....	44
<u>Query frame layout</u> .....	45
<u>Response frame layout</u> .....	45
<u>Exception frame layout</u> .....	46
<u>Function MSXE3317 DigitalIOGetInputsFilterConfiguration</u> .....	46
<u>Description</u> .....	46
<u>Query frame layout</u> .....	47
<u>Response frame layout</u> .....	47
<u>Exception frame layout</u> .....	48
<u>Function MSXE3317 DigitalIOTestOutputsShortCircuit</u> .....	48
<u>Description</u> .....	48
<u>Query frame layout</u> .....	49
<u>Response frame layout</u> .....	49
<u>Exception frame layout</u> .....	50
<u>Function MSXE3317 DigitalIOReadChannel</u> .....	50
<u>Description</u> .....	50
<u>Query frame layout</u> .....	51
<u>Response frame layout</u> .....	51
<u>Exception frame layout</u> .....	52
<u>Function MSXE3317 DigitalIOReadPort</u> .....	52
<u>Description</u> .....	52
<u>Query frame layout</u> .....	52
<u>Response frame layout</u> .....	53
<u>Exception frame layout</u> .....	54
<u>Function MSXE3317 SinCosGetNumberOfChannels</u> .....	54
<u>Description</u> .....	54
<u>Query frame layout</u> .....	54
<u>Response frame layout</u> .....	55
<u>Exception frame layout</u> .....	55
<u>Function MSXE3317 SinCosRead</u> .....	56
<u>Description</u> .....	56
<u>Query frame layout</u> .....	56

# Table of Contents

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

<u>Response frame layout</u> .....	57
<u>Exception frame layout</u> .....	58

## **FC16 (write multiple register) Functions**.....59

<u>Function MXCommon SetHardwareTriggerFilterTime</u> .....	60
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon SetHardwareTriggerFilterTimeEx</u> .....	60
<u>Description</u> .....	60
<u>Query frame layout</u> .....	60
<u>Response frame layout</u> .....	61
<u>Exception frame layout</u> .....	61
<u>Function MXCommon SetHardwareTriggerFilterTimeEx</u> .....	62
<u>Description</u> .....	62
<u>Query frame layout</u> .....	62
<u>Response frame layout</u> .....	63
<u>Exception frame layout</u> .....	64
<u>Function MXCommon InitAndStartSynchroTimer</u> .....	64
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon InitAndStartSynchroTimerEx</u> .....	64
<u>Description</u> .....	64
<u>Query frame layout</u> .....	65
<u>Response frame layout</u> .....	66
<u>Exception frame layout</u> .....	67
<u>Function MXCommon InitAndStartSynchroTimerEx</u> .....	67
<u>Description</u> .....	67
<u>Query frame layout</u> .....	68
<u>Response frame layout</u> .....	69
<u>Exception frame layout</u> .....	70
<u>Function MXCommon StopAndReleaseSynchroTimer</u> .....	70
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon StopAndReleaseSynchroTimerEx</u> .....	70
<u>Description</u> .....	70
<u>Query frame layout</u> .....	71
<u>Response frame layout</u> .....	71
<u>Exception frame layout</u> .....	72
<u>Function MXCommon StopAndReleaseSynchroTimerEx</u> .....	72
<u>Description</u> .....	72
<u>Query frame layout</u> .....	73
<u>Response frame layout</u> .....	73
<u>Exception frame layout</u> .....	74
<u>Function MXCommon Reboot</u> .....	74
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon RebootEx</u> .....	74
<u>Description</u> .....	74
<u>Query frame layout</u> .....	75
<u>Response frame layout</u> .....	75
<u>Exception frame layout</u> .....	76

# Table of Contents

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

<b><u>Function MXCommon RebootEx</u></b> .....	76
<u>Description</u> .....	76
<u>Query frame layout</u> .....	77
<u>Response frame layout</u> .....	77
<u>Exception frame layout</u> .....	78
<b><u>Function MXCommon SetCustomerKey</u></b> .....	78
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon SetCustomerKeyEx</u> .....	78
<u>Description</u> .....	78
<u>Query frame layout</u> .....	79
<u>Response frame layout</u> .....	79
<u>Exception frame layout</u> .....	80
<b><u>Function MXCommon SetCustomerKeyEx</u></b> .....	80
<u>Description</u> .....	80
<u>Query frame layout</u> .....	81
<u>Response frame layout</u> .....	82
<u>Exception frame layout</u> .....	82
<b><u>Function MSXE3317 SetDataCursor</u></b> .....	83
<u>Description</u> .....	83
<u>Query frame layout</u> .....	83
<u>Response frame layout</u> .....	84
<u>Exception frame layout</u> .....	84
<b><u>Function MSXE3317 PressureSetChannelConfiguration</u></b> .....	85
<u>Description</u> .....	85
<u>Query frame layout</u> .....	85
<u>Response frame layout</u> .....	86
<u>Exception frame layout</u> .....	86
<b><u>Function MSXE3317 PressureSetSamplingRate</u></b> .....	87
<u>Description</u> .....	87
<u>Query frame layout</u> .....	88
<u>Response frame layout</u> .....	89
<u>Exception frame layout</u> .....	89
<b><u>Function MSXE3317 PressureCalibrationStart</u></b> .....	90
<u>Description</u> .....	90
<u>Query frame layout</u> .....	90
<u>Response frame layout</u> .....	91
<u>Exception frame layout</u> .....	92
<b><u>Function MSXE3317 PressureCalibrationNextStep</u></b> .....	92
<u>Description</u> .....	92
<u>Query frame layout</u> .....	92
<u>Response frame layout</u> .....	93
<u>Exception frame layout</u> .....	94
<b><u>Function MSXE3317 PressureCalibrationBreak</u></b> .....	94
<u>Description</u> .....	94
<u>Query frame layout</u> .....	94
<u>Response frame layout</u> .....	95
<u>Exception frame layout</u> .....	96

# Table of Contents

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

<b>Function MSXE3317 AcquisitionAutoRefreshInitAndStart</b> .....	96
<u>Description</u> .....	96
<u>Query frame layout</u> .....	98
<u>Response frame layout</u> .....	100
<u>Exception frame layout</u> .....	100
<b>Function MSXE3317 AcquisitionAutoRefreshStopAndRelease</b> .....	101
<u>Description</u> .....	101
<u>Query frame layout</u> .....	101
<u>Response frame layout</u> .....	102
<u>Exception frame layout</u> .....	102
<b>Function MSXE3317 AcquisitionSequenceInitAndStart</b> .....	103
<u>Description</u> .....	103
<u>Query frame layout</u> .....	105
<u>Response frame layout</u> .....	106
<u>Exception frame layout</u> .....	107
<b>Function MSXE3317 AcquisitionSequenceStopAndRelease</b> .....	107
<u>Description</u> .....	107
<u>Query frame layout</u> .....	108
<u>Response frame layout</u> .....	108
<u>Exception frame layout</u> .....	109
<b>Function MSXE3317 DigitalIOSetPortDirections</b> .....	109
<u>Description</u> .....	109
<u>Query frame layout</u> .....	110
<u>Response frame layout</u> .....	111
<u>Exception frame layout</u> .....	111
<b>Function MSXE3317 DigitalIOSetInputsFilterTime</b> .....	112
<u>Description</u> .....	112
<u>Query frame layout</u> .....	112
<u>Response frame layout</u> .....	113
<u>Exception frame layout</u> .....	113
<b>Function MSXE3317 DigitalIOEnableDisableInputsFilter</b> .....	114
<u>Description</u> .....	114
<u>Query frame layout</u> .....	114
<u>Response frame layout</u> .....	115
<u>Exception frame layout</u> .....	116
<b>Function MSXE3317 DigitalIORearmOutputsShortCircuit</b> .....	116
<u>Description</u> .....	116
<u>Query frame layout</u> .....	116
<u>Response frame layout</u> .....	117
<u>Exception frame layout</u> .....	118
<b>Function MSXE3317 DigitalIOWriteChannel</b> .....	118
<u>Description</u> .....	118
<u>Query frame layout</u> .....	119
<u>Response frame layout</u> .....	119
<u>Exception frame layout</u> .....	120
<b>Function MSXE3317 DigitalIOWritePort</b> .....	120
<u>Description</u> .....	120

# Table of Contents

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

<u>Query frame layout</u> .....	121
<u>Response frame layout</u> .....	122
<u>Exception frame layout</u> .....	122
<u>Function MSXE3317 SinCosInit</u> .....	123
<u>Description</u> .....	123
<u>Query frame layout</u> .....	123
<u>Response frame layout</u> .....	124
<u>Exception frame layout</u> .....	125
<u>Function MSXE3317 SinCosRelease</u> .....	125
<u>Description</u> .....	125
<u>Query frame layout</u> .....	126
<u>Response frame layout</u> .....	126
<u>Exception frame layout</u> .....	127
<u>Function MSXE3317 SinCosClear</u> .....	127
<u>Description</u> .....	128
<u>Query frame layout</u> .....	128
<u>Response frame layout</u> .....	129
<u>Exception frame layout</u> .....	129
<u>Function MSXE3317 SinCosClearError</u> .....	130
<u>Description</u> .....	130
<u>Query frame layout</u> .....	130
<u>Response frame layout</u> .....	131
<u>Exception frame layout</u> .....	132
<u>Function MSXE3317 SinCosInitIndex</u> .....	132
<u>Description</u> .....	132
<u>Query frame layout</u> .....	133
<u>Response frame layout</u> .....	134
<u>Exception frame layout</u> .....	134
<u>Function MSXE3317 SinCosReleaseIndex</u> .....	135
<u>Description</u> .....	135
<u>Query frame layout</u> .....	135
<u>Response frame layout</u> .....	136
<u>Exception frame layout</u> .....	137
<u>Function MSXE3317 SinCosInitAndEnableCompareLogic</u> .....	137
<u>Description</u> .....	137
<u>Query frame layout</u> .....	138
<u>Response frame layout</u> .....	139
<u>Exception frame layout</u> .....	140
<u>Function MSXE3317 SinCosDisableAndReleaseCompareLogic</u> .....	140
<u>Description</u> .....	140
<u>Query frame layout</u> .....	141
<u>Response frame layout</u> .....	141
<u>Exception frame layout</u> .....	142

## **FC23 (read/write registers) Functions**.....143

<u>Query frame layout</u> .....	143
<u>Response frame layout</u> .....	144



# Table of Contents

<b><u>FC23 (read/write registers) Functions</u></b>	
<u>Exception frame layout</u> .....	144
<b><u>Exception code description</u></b> .....	145
<b><u>Siemens Step 7 compatibility information (AWL/SDF code)</u></b> .....	146

# 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>MSXE3317_PressureGetNumberOfChannels</u></a>	Register: <b>21000</b>
• <a href="#"><u>MSXE3317_PressureCalibrationGetCurrentStatus</u></a>	Register: <b>21150</b>
• <a href="#"><u>MSXE3317_AcquisitionGetNumberOfChannels</u></a>	Register: <b>15000</b>
• <a href="#"><u>MSXE3317_AcquisitionGetChannelsInfo</u></a>	Register: <b>15050</b>
• <a href="#"><u>MSXE3317_AcquisitionAutoRefreshGetValues</u></a>	Register: <b>15300</b>
• <a href="#"><u>MSXE3317_AcquisitionAutoRefreshGetBlockingValues</u></a>	Register: <b>15450</b>
• <a href="#"><u>MSXE3317_AcquisitionAutoRefreshGetConfiguration</u></a>	Register: <b>15600</b>
• <a href="#"><u>MSXE3317_AcquisitionSequenceGetConfiguration</u></a>	Register: <b>15650</b>
• <a href="#"><u>MSXE3317_DigitalIOGetNumberOfChannels</u></a>	Register: <b>26000</b>
• <a href="#"><u>MSXE3317_DigitalIOGetNumberOfPorts</u></a>	Register: <b>26050</b>
• <a href="#"><u>MSXE3317_DigitalIOGetPortAvailableDirections</u></a>	Register: <b>26100</b>
• <a href="#"><u>MSXE3317_DigitalIOGetPortDirections</u></a>	Register: <b>26150</b>

## MODBUS interface description

- MSXE3317 DigitalIOGetInputsFilterConfiguration Register: **26200**
- MSXE3317 DigitalIOTestOutputsShortCircuit Register: **26250**
- MSXE3317 DigitalIOReadChannel Register: **26300**
- MSXE3317 DigitalIOReadPort Register: **26350**
- MSXE3317 SinCosGetNumberOfChannels Register: **37000**
- MSXE3317 SinCosRead Register: **37100**

## 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	6	0x0600	0x0006

## 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	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
Return Value	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	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	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

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	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 -	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 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

Exception frame layout

## MODBUS interface description

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)
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
	1			??	??

### Query frame layout

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

## 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	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	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	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
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

## MODBUS interface description

[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
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

## MODBUS interface description

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)
	2			0x0000	0x0000

Response 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	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
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 -	0x0000	0x0000



## MODBUS interface description

			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	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]
bCryptedExceptionArray	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)
	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 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

Exception frame layout

## MODBUS interface description

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

### 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
	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 MSXE3317\_\_PressureGetNumberOfChannels

### Description

Return the number of pressure channels.

### Parameters:

[Response frame layout] **ulNumber** Return the number of available pressure channels

### Returns:

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

### 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	21000	0x0852	0x5208
word count	2	16-bit	2	0x0200	0x0002

Exception frame layout

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

## 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
ulNumber	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 MSXE3317\_\_PressureCalibrationGetCurrentStatus

## Description

Return the type of the analog input channels

### Parameters:

[Response frame layout] **ulStatus** Calibration current status

- ◆ 0: No calibration in progress
- ◆ 1: Wait to setting the reference value
- ◆ 2: Selected reference value calibration in progress
- ◆ 3: Calibration finished
- ◆ 4: User break occur
- ◆ 5: Can not calibrate the selected reference value

[Response frame layout] **ulChannel** : Current calibration channel

[Response frame layout] **fRefValue** : Current selected calibration reference value

[Response frame layout] **ulDigitalValue** : Last measured digital value

### Returns:

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

- ◆ **syserrno** : system-error code (the value of the libc "errno" code) EPERM means a autorefresh acquisition was not started.

## 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	1	8-bit integer	0x03	0x03	0x03

## MODBUS interface description

code					
Reference number (=register)	2	16-bit integer	21150	0x9E52	0x529E
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
ulStatus	4	32-bit integer	See the description above	0x????????	0x????????
ulChannel	4	32-bit integer	See the description above	0x????????	0x????????
fRefValue	4	32-bit floating point	See the description above	0x????????	0x????????
ulDigitalValue	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 MSXE3317\_\_AcquisitionGetNumberOfChannels

### Description

Return the number of acquisition channels.

### Parameters

- [Response frame layout] **ulNumber** Return the number of available acquisition channels

### Returns

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

- **0** The remote function performed OK
- **-100** 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	6	0x0600	0x0006
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	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	15000	0x983A	0x3A98
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
ulNumber	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
	1		0x83	0x83	0x83

### Query frame layout

## MODBUS interface description

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

## Function MSXE3317\_\_AcquisitionGetChannelsInfo

### Description

Return the acquisition channels type and hardware position.

### Parameters:

- [Response frame layout] **ulType** Array that contains the channels type.
  - ◆ 0 : Not available
  - ◆ 1 : Temperature channel
  - ◆ 2 : Pressure channel
  - ◆ 3 : Transducer channel
  - ◆ 4 : Analog input channel
  - ◆ 5 : Analog input ICP channel
  - ◆ 6 : Digital I/O port
  - ◆ ulType [0] : Channel 0 type
  - ◆ ...
  - ◆ ulType [15] : Channel 15 type
- [Response frame layout] **ulHwPosition** Array that contain the hardware position index (0 to 7)
  - ◆ ulType [0] : Channel 0 hardware position index
  - ◆ ...
  - ◆ ulType [15] : Channel 15 hardware position index
- [Response frame layout] **ulChannelIndex** Array that contain the functionality channel index.
  - ◆ ulType [0] : Channel 0 hardware position index
  - ◆ ...
  - ◆ ulType [15] : Channel 15 functionality channel index

### Returns

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

- **0** The remote function performed OK
- **-2** Channel selection wrong
- **-100** Internal system error occurred. See value of syserrno

### Query frame layout

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

Exception 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	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	15050	0xCA3A	0x3ACA
word count	2	16-bit integer	96	0x6000	0x0060

### 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	195	0xC300	0x00C3
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	192	0xC0	0xC0
ulType	64	32-bit integer array	See the description above	0x???????[16]	0x???????[16]
ulHwPosition	64	32-bit integer array	See the description above	0x???????[16]	0x???????[16]
ulChannelIndex	64	32-bit integer array	See the description above	0x???????[16]	0x???????[16]

### Query frame layout

## 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 MSXE3317\_\_AcquisitionAutoRefreshGetValues

### Description

Returns the values acquired in auto refresh mode

#### Parameters:

- [Response frame layout] ***ulTimeStampLow*** Number of microseconds since epoch
- [Response frame layout] ***ulTimeStampHigh*** Number of seconds since epoch
- [Response frame layout] ***ulAutoRefreshCounter*** Number of sequence acquisition since the start
- [Response frame layout] ***fValues*** Array that contains the channels values
  - ◆ pdValues [0] : Channel 0 value
  - ◆ ...
  - ◆ pdValues [15] : Channel 15 value

#### Returns:

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

- ◆ **0** The remote function performed OK
- ◆ **-2** No Acquisition in progress
- ◆ **-3** 2s timeout occur (If you have enabled the blocking mode).
- ◆ **-100** 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	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	15300	0xC43B	0x3BC4
word count	2	16-bit integer	38	0x2600	0x0026

## 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	79	0x4F00	0x004F
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	76	0x4C	0x4C
ulTimeStampLow	4	32-bit integer	See the description above	0x???????	0x???????

## MODBUS interface description

ulTimeStampHigh	4	32-bit integer	See the description above	0x????????	0x????????
ulCounterValue	4	32-bit integer	See the description above	0x????????	0x????????
fValues	64	32-bit floating point 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

### MSXE3317\_\_AcquisitionAutoRefreshGetBlockingValues

## Description

Returns the values acquired in auto refresh mode after a new cycle occur (wait a new auto refresh value cycle)

### Parameters:

- ◆ [Response frame layout] **ulTimeStampLow** Number of microseconds since epoch
- ◆ [Response frame layout] **ulTimeStampHigh** Number of seconds since epoch
- ◆ [Response frame layout] **ulAutoRefreshCounter** Number of sequence acquisition since the start
- ◆ [Response frame layout] **fValues** Array that contains the channels values
  - ◊ pdValues [0] : Channel 0 value

## MODBUS interface description

- ◇ ...
- ◇ pdValues [15] : Channel 15 value

### Returns:

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

- ◇ **0** The remote function performed OK
- ◇ **-2** No Acquisition in progress
- ◇ **-3** 2s timeout occur (If you have enabled the blocking mode).
- ◇ **-100** 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	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	15450	0x5A3C	0x3C5A
word count	2	16-bit integer	38	0x2600	0x0026

### 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		79	0x4F00	0x004F

## 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	0x03	0x03	0x03
Byte count	1	8-bit integer	76	0x4C	0x4C
ulTimeStampLow	4	32-bit integer	See the description above	0x????????	0x????????
ulTimeStampHigh	4	32-bit integer	See the description above	0x????????	0x????????
ulCounterValue	4	32-bit integer	See the description above	0x????????	0x????????
fValues	64	32-bit floating point 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

## MSXE3317\_\_AcquisitionAutoRefreshGetConfiguration



## Description

Get the current auto refresh acquisition configuration.

### Parameters:

- ◇ [Response frame layout] ***ulChannelMask*** Mask of the channel acquired by the auto refresh (1 bit = 1 Channel). If the value returned is 0, then the auto refresh acquisition is not initialised.
- ◇ [Response frame layout] ***ulAverageValue*** Average value
  - 1 : not used
  - max value : 255
- ◇ [Response frame layout] ***ulRefreshTimeUnit*** Refresh Time Unit
  - 0 : microsecond
  - 1 : millisecond
  - 2 : second
- ◇ [Response frame layout] ***ulRefreshTime*** Refresh Time
  - range from min 1000 to 65535 when the unit is the microsecond
  - range from min 1 to 65535 when the unit is the millisecond
  - range from min 1 to 65535 when the unit is the second
- ◇ [Response frame layout] ***ulTriggerMask*** Define the source of the trigger
  - 0 : trigger disabled
  - 1 : Enable Hardware Digital Input Trigger
  - 2 : Enable Synchro Trigger
  - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/Cosine input)
  - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- ◇ [Response frame layout] ***ulTriggerMode*** Define the trigger mode
  - 1 : One shot trigger
  - 2 : Sequence trigger
- ◇ [Response frame layout] ***ulHardwareTriggerEdge*** 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)
- ◇ [Response frame layout] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
  - 1 : all trigger event start the action
  - max value : 65535
- ◇ [Response frame layout] ***ulByTriggerNbrOfSeqToAcquire*** Define the number of sequence to acquire by each trigger event
  - 0 : continuous mode
  - 0 : number of sequence : (1..0xFFFFFFFF)
- ◇ [Response frame layout] ***ulDataFormat***
  - D0 : Absolute time stamp information
    - 0: no time stamp information
    - 1: time stamp information
  - D1 : Relative time stamp information
    - 0: no time stamp information
    - 1: time stamp information

## MODBUS interface description

- D2 : Auto refresh counter information
  - 0 : No auto refresh counter information
  - 1 : Auto refresh counter information
- D3 : System status information
  - 0 : No system status information required
  - 1 : System status information required
- D4 : Data format
  - 0: Digital value
  - 1: Analog value

◇ [Response frame layout] **ulRunning** Auto refresh acquisition running state

- 0 : Not running
- 1 : Auto refresh acquisition running

### Returns:

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

◇ 0 The remote function performed OK

◇ -100 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	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	15600	0xF03C	0x3CF0
word count	2	16-bit integer	22	0x1600	0x0016

### 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	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	0x03	0x03	0x03
Byte count	1	8-bit integer	44	0x2C	0x2C
ulChannelMask	4	32-bit integer	See the description above	0x????????	0x????????
ulAverageValue	4	32-bit integer	See the description above	0x????????	0x????????
ulRefreshTime	4	32-bit integer	See the description above	0x????????	0x????????
ulRefreshTimeUnit	4	32-bit integer	See the description above	0x????????	0x????????
ulTriggerMask	4	32-bit integer	See the description above	0x????????	0x????????
ulTriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
ulHardwareTriggerEdge	4	32-bit integer	See the description above	0x????????	0x????????
ulHardwareTriggerCount	4	32-bit integer	See the description above	0x????????	0x????????
ulByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????	0x????????
ulDataFormat	4	32-bit integer	See the description above	0x????????	0x????????
ulRunning	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

### MSXE3317\_\_AcquisitionSequenceGetConfiguration

## Description

Get the current sequence acquisition configuration.

### Parameters

- ◇ [Response frame layout] ***ulNbrOfChannel*** Nbr of channel in the sequence. If 0 no sequence initialised
- ◇ [Response frame layout] ***psChannelList*** List of the channel who compose the sequence.
- ◇ [Response frame layout] ***ulAcquisitionTime*** Acquisition Time
  - range from min 1000 to 65535 when the unit is the microsecond
  - range from min 1 to 65535 when the unit is the millisecond
  - range from min 1 to 65535 when the unit is the second
- ◇ [Response frame layout] ***ulAcquisitionTimeUnit*** Acquisition Time Unit
  - 0 : us
  - 1 : ms
  - 2 : s
- ◇ [Response frame layout] ***ulNbrOfSequence*** Number of sequence to acquire
  - 0 : continuous mode
  - Superior to 0 : number of sequence
- ◇ [Response frame layout] ***ulNbrMaxSequenceToTransfer*** Max number of sequence to acquire before a data transfer. (from 1 to 65535)

## MODBUS interface description

- ◇ [Response frame layout] **ulTriggerMask** Define the source of the trigger
  - 0 : trigger disabled
  - 1 : Enable Hardware Digital Input Trigger
  - 2 : Enable Synchro Trigger
  - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/Cosine input)
  - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- ◇ [Response frame layout] **ulTriggerMode** Define the trigger mode
  - 1 : One shot trigger
  - 2 : Sequence trigger
- ◇ [Response frame layout] **ulHardwareTriggerEdge** Define the edge of the hardware trigger who generate a trigger action
  - 1 : rising front (Only if hardware trigger selected)
  - 2 : falling front (Only if hardware trigger selected)
  - 3 : Both front (Only if hardware trigger selected)
- ◇ [Response frame layout] **ulHardwareTriggerCount** Define the number of trigger events before the action occur
  - 1 : all trigger event start the action
  - max value : 65535
- ◇ [Response frame layout] **ulByTriggerNbrOfSeqToAcquire** define the number of sequence to acquire by each trigger event
  - 0 : continuous mode
  - Superior to 0 : number of sequence : (1..0xFFFFFFFF)
- ◇ [Response frame layout] **ulDataFormat** Data format option
  - D0 : Absolute time stamp information
    - 0: no time stamp information
    - 1: time stamp information
  - D1 : Relative time stamp information
    - 0: no time stamp information
    - 1: time stamp information
  - D2 : Sequence counter information
    - 0 : No sequence counter information
    - 1 : Sequence counter information
  - D3 : System status information
    - 0 : No system status information required
    - 1 : System status information required
  - D4 : Data format
    - 0: Digital value
    - 1: Analog value
- ◇ [Response frame layout] **ulRunning** Sequence acquisition running state
  - 0 : Not running
  - 1 : Sequence acquisition running

### Returns:

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

- ◇ **0** The remote function performed OK
- ◇ **-100** 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	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	15650	0x223D	0x3D22
word count	2	16-bit integer	56	0x3800	0x0038

## 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	115	0x7300	0x0073
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	112	0x70	0x70
ulNbrOfChannel	4	32-bit integer	See the description above	0x???????	0x???????

## MODBUS interface description

ulChannelList	64	32-bit integer array	See the description above	0x????????[16]	0x????????[16]
ulAcquisitionTime	4	32-bit integer	See the description above	0x????????	0x????????
ulAcquisitionTimeUnit	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrOfSequence	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrMaxSequenceToTransfer	4	32-bit integer	See the description above	0x????????	0x????????
ulTriggerMask	4	32-bit integer	See the description above	0x????????	0x????????
ulTriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
ulHardwareTriggerEdge	4	32-bit integer	See the description above	0x????????	0x????????
ulHardwareTriggerCount	4	32-bit integer	See the description above	0x????????	0x????????
ulByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????	0x????????
ulDataFormat	4	32-bit integer	See the description above	0x????????	0x????????
ulRunning	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

## 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	0x83	0x83	0x83
Exception code	1	8-bit integer	See corresponding chapter	??	??

## Function MSXE3317\_\_DigitalIOGetNumberOfChannels

### Description

Returns the number of digital I/O channels.

### Parameters

◇ [Response frame layout] **ulNumber** Number of available digital I/O channels

### Returns

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

◇ **0** The remote function performed OK

◇ **-100** 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	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	2	16-bit integer	26000	0x9065	0x6590



## MODBUS interface description

(=register)					
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
ulNumber	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 MSXE3317\_\_DigitalIOGetNumberOfPorts

### Description

Returns the number of digital I/O ports. A port is a set of consecutive digital I/O channels, whose status can be written or read at the same time.

### Parameters

◇ [Response frame layout] **ulNumber** Number of available digital I/O ports

### Returns

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

◇ **0** The remote function performed OK

◇ **-100** 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	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	26050	0xC265	0x65C2
word count	2	16-bit integer	2	0x0200	0x0002

### 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	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
ulNumber	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

### MSXE3317\_\_DigitalIOGetPortAvailableDirections

## Description

Returns the available directions for the selected port (input or output). The port is selected via MSXE3317\_\_SetDataCursor.

## MODBUS interface description

### Parameters

- ◇ [Response frame layout] **ullInputs** Digital inputs availability. Each bit indicates if the channel can be used as an input.
- ◇ [Response frame layout] **ulOutputs** Digital outputs availability. Each bit indicates if the channel can be used as an output.

### Returns

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The selected port (selected via MSXE3317\_\_SetDataCursor) is wrong
- ◇ **-100** 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	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	26100	0xF465	0x65F4
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
ulInputs	4	32-bit integer	See the description above	0x????????	0x????????
ulOutputs	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 MSXE3317\_\_DigitalIOGetPortDirections

### Description

Reads the current digital I/O direction for the selected port. The port is selected via MSXE3317\_\_SetDataCursor.

### Parameters

## MODBUS interface description

◇ [Response frame layout] ***ulDirection*** Current digital I/O direction. Each bit indicates if the channel is used as an input or an output (0: input, 1: output)

### Returns

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

◇ **0** The remote function performed OK

◇ **-2** The selected port (selected via MSXE3317\_\_SetDataCursor) is wrong

◇ **-100** 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	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	26150	0x2666	0x6626
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

## 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
Byte count	1	8-bit integer	4	0x04	0x04
ulDirection	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

## MSXE3317\_\_DigitalIOGetInputsFilterConfiguration

## Description

Reads the digital inputs filter configuration for the selected port. The port is selected via MSXE3317\_\_SetDataCursor.

### Parameters

- ◇ [Response frame layout] **ulFilterTime** Filter time value (from 1 to 16777215) 1 corresponds to 250 ns, 2 corresponds to 500 ns, ...
- ◇ [Response frame layout] **ulFilter** Digital inputs filter selection. Each bit indicate the filter state for one digital input channel.

### Returns

## MODBUS interface description

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The selected port (selected via MSXE3317\_\_SetDataCursor) is wrong
- ◇ **-100** 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	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	26200	0x5866	0x6658
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	1	8-bit integer	0x03	0x03	0x03



## MODBUS interface description

code					
Byte count	1	8-bit integer	8	0x08	0x08
ulFilterTime	4	32-bit integer	See the description above	0x????????	0x????????
ulFilter	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

### MSXE3317\_\_DigitalIOTestOutputsShortCircuit

## Description

Get the short-circuit status of the outputs of the selected port. The port is selected via MSXE3317\_\_SetDataCursor.

The function returns a mask of bits (32 bits). Each bit represents the short-circuit state of an output.

If you detect a short circuit, first solve it, and then, call the MSXE3317\_\_DigitalIORearmOutputsShortCircuit function.

## Parameters

## MODBUS interface description

◇ [Response frame layout] **ulState** Digital outputs short circuit state. Each bit represents the short-circuit state of one digital output channel.

### Returns

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

◇ **0** The remote function performed OK

◇ **-2** The selected port (selected via MSXE3317\_\_SetDataCursor) is wrong

◇ **-100** 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	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	26250	0x8A66	0x668A
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

## 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
Byte count	1	8-bit integer	4	0x04	0x04
ulState	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 MSXE3317\_\_DigitalIOReadChannel

### Description

Read the selected digital I/O channel. The channel is selected via MSXE3317\_\_SetDataCursor.

If the selected channel is an output, then this function returns the current output state.

### Parameters

◇ [Response frame layout] **ulState** Digital I/O channel state (0: off, 1: on)

### Returns

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

## MODBUS interface description

- ◇ 0 The remote function performed OK
- ◇ -2 The selected channel (selected via MSXE3317\_\_SetDataCursor) is wrong
- ◇ -100 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	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	26300	0xBC66	0x66BC
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		4	0x04	0x04

## MODBUS interface description

		8-bit integer			
ulState	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 MSXE3317\_\_DigitalIOReadPort

### Description

Read the selected digital I/O port. The port is selected via MSXE3317\_\_SetDataCursor.

### Parameters

- ◇ [Response frame layout] **ulState** Digital I/O state. Each bit indicates the state of one digital I/O channel.

### Returns

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The selected port (selected via MSXE3317\_\_SetDataCursor) is wrong
- ◇ **-100** Internal system error occurred. See value of syserrno

### 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	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	26350	0xEE66	0x66EE
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
ulState	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

### MSXE3317\_\_SinCosGetNumberOfChannels

## Description

Return the number of Sine/Cosine channels.

## Parameters

◇ [Response frame layout] **ulNumber** Number of available Sine / Cosine channels

## Returns

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

◇ **0** The remote function performed OK

◇ **-100** 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 -	0x0000	0x0000

## MODBUS interface description

			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	37000	0x8890	0x9088
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
ulNumber	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



## 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 MSXE3317\_\_SinCosRead

### Description

Read the measured value of the selected Sine/Cosine channel. The channel is selected via MSXE3317\_\_SetDataCursor.

### Parameters

- ◇ [Response frame layout] **fValue** Measured value (Unit)
- ◇ [Response frame layout] **ulValue** Measured value (Raw value)
- ◇ [Response frame layout] **ulMeasureError** Status of the Sine/Cosine measurement.  
If the value is 1, then an error occurred. When an error occurred, the reference point is corrupted. To correct it, you must first reset the error flag using the function SinCosClearError, and then reset your reference point.
- ◇ [Response frame layout] **ulInfo01** Reserved
- ◇ [Response frame layout] **ulInfo02** Reserved

### Returns

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The PLD is not working
- ◇ **-3** The selected channel (using MSXE3317\_\_SetDataCursor) is wrong
- ◇ **-4** The current status is not correct. The selected Sine/Cosine channel is not initialised
- ◇ **-100** 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	0x0000	0x0000

## MODBUS interface description

			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	37100	0xEC90	0x90EC
word count	2	16-bit integer	10	0x0A00	0x000A

## 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	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	0x03	0x03	0x03
Byte count	1	8-bit integer	20	0x14	0x14
fValue	4	32-bit floating point	See the description above	0x????????	0x????????
ulValue	4	32-bit integer	See the description above	0x????????	0x????????
ulMeasureError	4	32-bit integer	See the description above	0x????????	0x????????
ulInfo01	4	32-bit integer	See the description	0x????????	0x????????

## MODBUS interface description

			above		
ulInfo02	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	??	??

# FC16 (write multiple register) Functions

[Top](#)

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

◇ <a href="#">MXCommon_SetHardwareTriggerFilterTime</a>	Register: <b>100</b>
◇ <a href="#">MXCommon_SetHardwareTriggerFilterTimeEx</a>	Register: <b>11000</b>
◇ <a href="#">MXCommon_InitAndStartSynchroTimer</a>	Register: <b>101</b>
◇ <a href="#">MXCommon_InitAndStartSynchroTimerEx</a>	Register: <b>11050</b>
◇ <a href="#">MXCommon_StopAndReleaseSynchroTimer</a>	Register: <b>102</b>
◇ <a href="#">MXCommon_StopAndReleaseSynchroTimerEx</a>	Register: <b>11100</b>
◇ <a href="#">MXCommon_Reboot</a>	Register: <b>103</b>
◇ <a href="#">MXCommon_RebootEx</a>	Register: <b>11150</b>
◇ <a href="#">MXCommon_SetCustomerKey</a>	Register: <b>104</b>
◇ <a href="#">MXCommon_SetCustomerKeyEx</a>	Register: <b>11200</b>
◇ <a href="#">MSXE3317_SetDataCursor</a>	Register: <b>65530</b>
◇ <a href="#">MSXE3317_PressureSetChannelConfiguration</a>	Register: <b>21200</b>
◇ <a href="#">MSXE3317_PressureSetSamplingRate</a>	Register: <b>21250</b>
◇ <a href="#">MSXE3317_PressureCalibrationStart</a>	Register: <b>21300</b>
◇ <a href="#">MSXE3317_PressureCalibrationNextStep</a>	Register: <b>21350</b>
◇ <a href="#">MSXE3317_PressureCalibrationBreak</a>	Register: <b>21400</b>
◇ <a href="#">MSXE3317_AcquisitionAutoRefreshInitAndStart</a>	Register: <b>15750</b>
◇ <a href="#">MSXE3317_AcquisitionAutoRefreshStopAndRelease</a>	Register: <b>15850</b>
◇ <a href="#">MSXE3317_AcquisitionSequenceInitAndStart</a>	Register: <b>15900</b>
◇ <a href="#">MSXE3317_AcquisitionSequenceStopAndRelease</a>	Register: <b>15950</b>
◇ <a href="#">MSXE3317_DigitalIOSetPortDirections</a>	Register: <b>26500</b>
◇ <a href="#">MSXE3317_DigitalIOSetInputsFilterTime</a>	Register: <b>26550</b>
◇ <a href="#">MSXE3317_DigitalIOEnableDisableInputsFilter</a>	Register: <b>26600</b>
◇ <a href="#">MSXE3317_DigitalIORearmOutputsShortCircuit</a>	Register: <b>26650</b>
◇ <a href="#">MSXE3317_DigitalIOWriteChannel</a>	Register: <b>26700</b>
◇ <a href="#">MSXE3317_DigitalIOWritePort</a>	Register: <b>26750</b>
◇ <a href="#">MSXE3317_SinCosInit</a>	Register: <b>37200</b>
◇ <a href="#">MSXE3317_SinCosRelease</a>	Register: <b>37300</b>
◇ <a href="#">MSXE3317_SinCosClear</a>	Register: <b>37400</b>
◇ <a href="#">MSXE3317_SinCosClearError</a>	Register: <b>37500</b>
◇ <a href="#">MSXE3317_SinCosInitIndex</a>	Register: <b>37600</b>
◇ <a href="#">MSXE3317_SinCosReleaseIndex</a>	Register: <b>37700</b>
◇ <a href="#">MSXE3317_SinCosInitAndEnableCompareLogic</a>	Register: <b>37800</b>
◇ <a href="#">MSXE3317_SinCosDisableAndReleaseCompareLogic</a>	Register: <b>37900</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
  - **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

## MODBUS interface description

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

## 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\_\_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 GetLastErrorStatusEx).

- ◇ **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

## MODBUS interface description

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	16-bit integer	4	0x0400	0x0004
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 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\_\_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

## MODBUS interface description

- 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 occured
- -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
	1		0x10	0x10	0x10

## MODBUS interface description

MODBUS Function code		8-bit integer			
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 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
	1		0 or 1		

## MODBUS interface description

unit identifier		8-bit integer		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
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

## MODBUS interface description

- 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 GetLastCommandStatusEx)**

- 0 : means the remote function performed OK
- -1: means an system error occured
- -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

## MODBUS interface description

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 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
	2		0	0x0000	0x0000

## 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	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
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)

## MODBUS interface description

### 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
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	0x0000	0x0000



## 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	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 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\_\_StopAndReleaseSynchroTimerEx

## Description

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

### Parameters:

## MODBUS interface description

[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 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
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 -	0x0000	0x0000

## MODBUS interface description

			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	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 - 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\_\_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:

Response frame layout

## MODBUS interface description

[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
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	0x0000	0x0000

## 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	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 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\_\_RebootEx

### Description

Ask the MSX-E module to reboot

### Parameters:

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

### 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 (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
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		6	0x0600	0x0006

## 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	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
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]

## MODBUS interface description

[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 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	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	0x0000	0x0000



## MODBUS interface description

			- copied 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	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:

## MODBUS interface description

[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 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	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
Reference number (=register)	2	16-bit integer	11200	0xC02B	0x2BC0
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 MSXE3317\_\_SetDataCursor

### Description

Change the active data cursor cursor

### Parameters:

[Query frame layout] **ulCursor**: New cursor value

### Returns:

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

- 0: success
- -1: otherwise : internal 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 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	65530	0xFAFF	0xFFFFA
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulCursor	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	65530	0xFAFF	0xFFFFA
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

## MSXE3317\_\_PressureSetChannelConfiguration

### Description

Pressure sensor configuration for the selected channel

#### Parameters:

[Query frame layout] ***ulChannel*** Channel selection. 0 to 15

[Query frame layout] ***fSensorSensibility*** Sensor sensibility (mV/V/bar or mV/V/mbar or mV/V/Pa, ...). Refer to sensor documentation

[Query frame layout] ***fSensorOffset*** Sensor V offset for 0 mV/V/bar, 0 mV/V/mbar, ... Refer to sensor documentation

[Query frame layout] ***ulOption1*** Reserved. Set to 0

#### Returns:

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

◇ ***syserrno*** : system-error code (the value of the libc "errno" code) EPERM means a autorefresh acquisition was not started.

### 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	21200	0xD052	0x52D0
word count	2		8	0x0800	0x0008

## MODBUS interface description

		16-bit integer			
byte count	1	8-bit integer	16	0x10	0x10
ulChannel	4	32-bit integer	See the description above	0x????????	0x????????
fSensorSensibility	4	32-bit floating point	See the description above	0x????????	0x????????
fSensorOffset	4	32-bit floating point	See the description above	0x????????	0x????????
ulOption1	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	21200	0xD052	0x52D0
word count	2	16-bit integer	8	0x0800	0x0008

## Exception frame layout

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

## MODBUS interface description

identifier		integer	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	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE3317\_\_PressureSetSamplingRate

### Description

Pressure acquisition sampling rate selection

#### Parameters:

[Query frame layout] ***ulChannelGroup*** Channel group selection.

- ◇ 0 for channels 0 and 1
- ◇ 1 for channels 2 and 3
- ◇ 2 for channels 4 and 5
- ◇ 3 for channels 6 and 7
- ◇ 4 for channels 8 and 9
- ◇ 5 for channels 10 and 11
- ◇ 6 for channels 12 and 13
- ◇ 7 for channels 14 and 15
- ◇ ...
- ◇ 255 for all channels

[Query frame layout] ***ulBaseSamplingRate*** Sampling rate selection

- ◇ 5 for 5Hz
- ◇ 10 for 10Hz
- ◇ 20 for 20Hz
- ◇ 40 for 40Hz
- ◇ 80 for 80Hz
- ◇ 160 for 160Hz
- ◇ 320 for 320Hz
- ◇ 640 for 640Hz



## MODBUS interface description

◇ 1000 for 1000Hz

◇ 2000 for 2000Hz

[Query frame layout] **ulOption1** Reserved. Set to 0

### Returns:

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

◇ **syserrno** : system-error code (the value of the libc "errno" code) EPERM means a autorefresh acquisition was not started.

### 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	21250	0x0253	0x5302
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannelGroup	4	32-bit integer	See the description above	0x????????	0x????????
ulBaseSamplingRate	4	32-bit integer	See the description above	0x????????	0x????????
ulOption1	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	21250	0x0253	0x5302
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

# MSXE3317\_\_PressureCalibrationStart

## Description

This function start the calibration thread

### Parameters:

[Query frame layout] ***ulChannelGroup*** Channel group selection.

- ◇ 0 for channels 0 and 1
- ◇ 1 for channels 2 and 3
- ◇ 2 for channels 4 and 5
- ◇ 3 for channels 6 and 7
- ◇ 4 for channels 8 and 9
- ◇ 5 for channels 10 and 11
- ◇ 6 for channels 12 and 13
- ◇ 7 for channels 14 and 15
- ◇ ...

[Query frame layout] ***ulGainSelection*** Selected gain (1, 2, 4, 8, 16, 32, 64 or 128)

[Query frame layout] ***ulNbrOfRefValues*** Nbr of channel reference values used for the calibration (Max 16)

[Query frame layout] ***fRefValues*** Reference voltage values for the calibration (Max 16)

### Returns:

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

- ◇ ***syserrno*** : system-error code (the value of the libc "errno" code) EPERM means a autorefresh acquisition was not started.

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endia (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	83	0x5300	0x0053
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	21300	0x3453	0x5334
word count	2	16-bit integer	38	0x2600	0x0026
byte count	1	8-bit integer	76	0x4C	0x4C
ulChannelGroup	4	32-bit integer	See the description above	0x???????	0x???????
ulGainSelection	4	32-bit integer	See the description above	0x???????	0x???????
ulNbrOfRefValues	4	32-bit integer	See the description above	0x???????	0x???????
fRefValues	64	32-bit floating point 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	21300	0x3453	0x5334

## MODBUS interface description

word count	2	16-bit integer	38	0x2600	0x0026
------------	---	----------------	----	--------	--------

### 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

### MSXE3317\_\_PressureCalibrationNextStep

### Description

This function start the next calibration step

#### Parameters:

[Query frame layout] ***ulOption1*** Reserved. Set to 0

#### Returns:

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

◇ ***syserrno*** : system-error code (the value of the libc "errno" code) EPERM means a autorefresh acquisition was not started.

### 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	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	21350	0x6653	0x5366
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	21350	0x6653	0x5366
word count	2		2	0x0200	0x0002

## MODBUS interface description

		16-bit 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

### MSXE3317\_\_PressureCalibrationBreak

### Description

This function break the current calibration

#### Parameters:

[Query frame layout] ***ulOption1*** Reserved. Set to 0

#### Returns:

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

◇ ***syserrno*** : system-error code (the value of the libc "errno" code) EPERM means a autorefresh acquisition was not started.

### 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	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	21400	0x9853	0x5398
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	21400	0x9853	0x5398
word count	2		2	0x0200	0x0002



## MODBUS interface description

		16-bit 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

### MSXE3317\_\_AcquisitionAutoRefreshInitAr

### Description

Initialise and start an auto refresh acquisition using provided configuration

#### Parameters

- ◆ [Query frame layout] ***ulChannelMask*** Mask of the channel to acquire by the auto refresh (1 bit = 1 Channel). 0 for all available acquisition channels
- ◆ [Query frame layout] ***ulAverageValue*** Set the average value
  - ◇ 1 : not used
  - ◇ max value : 255
- ◆ [Query frame layout] ***ulRefreshTimeUnit*** Refresh Time Unit
  - ◇ 0 : microsecond
  - ◇ 1 : millisecond
  - ◇ 2 : second
- ◆ [Query frame layout] ***ulRefreshTime*** Refresh Time

## MODBUS interface description

- ◇ range from min 1000 to 65535 when the unit is the microsecond
- ◇ range from min 1 to 65535 when the unit is the millisecond
- ◇ range from min 1 to 65535 when the unit is the second
- ◆ [Query frame layout] ***ulTriggerMask*** Define the source of the trigger
  - ◇ 0 : trigger disabled
  - ◇ 1 : Enable Hardware Digital Input Trigger
  - ◇ 2 : Enable Synchro Trigger
  - ◇ 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/Cosine input)
  - ◇ 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- ◆ [Query frame layout] ***ulTriggerMode*** Define the trigger mode
  - ◇ 1 : One shot trigger
  - ◇ 2 : Sequence trigger
- ◆ [Query frame layout] ***ulHardwareTriggerEdge*** 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] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
  - ◇ 1 : all trigger event start the action
  - ◇ max value : 65535
- ◆ [Query frame layout] ***ulByTriggerNbrOfSeqToAcquire*** Define the number of sequence to acquire by each trigger event
  - ◇ 0 : continuous mode
  - ◇ 0 : number of sequence : (1..0xFFFFFFFF)
- ◆ [Query frame layout] ***ulDataFormat***
  - ◇ D0 : Absolute time stamp information
    - 0: no time stamp information
    - 1: time stamp information
  - ◇ D1 : Relative time stamp information
    - 0: no time stamp information
    - 1: time stamp information
  - ◇ D2 : Auto refresh counter information
    - 0 : No auto refresh counter information
    - 1 : Auto refresh counter information
  - ◇ D3 : System status information
    - 0 : No system status information required
    - 1 : System status information required
  - ◇ D4 : Data format
    - 0: Digital value
    - 1: Analog value
- ◆ [Query frame layout] ***ulForceStart***

## MODBUS interface description

◇ 0 : Function return a error if any acquisition already in progress

◇ 1 : If any acquisition in progress then stop this

◆ [Query frame layout] **ulOption1** Reserved. Set to 0

◆ [Query frame layout] **ulOption2** Reserved. Set to 0

◆ [Query frame layout] **ulOption3** Reserved. Set to 0

### Returns:

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

◆ **0** The remote function performed OK

◆ **-2** Any acquisition already in progress

◆ **-3** Any selected channel not OK, call the diagnostic function for more information

◆ **-4** Channel Mask error

◆ **-5** Not available average value

◆ **-6** Not available refresh time unit

◆ **-7** The minimal refresh time is 1000 us

◆ **-8** The maximal refresh time is 65535

◆ **-9** Trigger mask not available

◆ **-10** Trigger mask : 2 different trigger source cannot be simultaneously be activated

◆ **-11** Trigger mode not available

◆ **-12** Trigger mask : 2 trigger mode cannot be simultaneously activated

◆ **-13** Hardware trigger : front definition error

◆ **-14** Hardware trigger count value not available

◆ **-15** Nbr of sequence to acquire by trigger mode not available

◆ **-16** Data format not available

◆ **-17** Selected channels combination not available

◆ **-100** 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	63	0x3F00	0x003F
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

## MODBUS interface description

Reference number (=register)	2	16-bit integer	15750	0x863D	0x
word count	2	16-bit integer	28	0x1C00	0x
byte count	1	8-bit integer	56	0x38	0x
ulChannelMask	4	32-bit integer	See the description above	0x????????	0x
ulAverageValue	4	32-bit integer	See the description above	0x????????	0x
ulRefreshTime	4	32-bit integer	See the description above	0x????????	0x
ulRefreshTimeUnit	4	32-bit integer	See the description above	0x????????	0x
ulTriggerMask	4	32-bit integer	See the description above	0x????????	0x
ulTriggerMode	4	32-bit integer	See the description above	0x????????	0x
ulHardwareTriggerEdge	4	32-bit integer	See the description above	0x????????	0x
ulHardwareTriggerCount	4	32-bit integer	See the description above	0x????????	0x
ulByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????	0x
ulDataFormat	4	32-bit integer	See the description above	0x????????	0x
ulForceStart	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
ulOption3	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	15750	0x863D	0x3D86
word count	2	16-bit integer	28	0x1C00	0x001C

**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

## MSXE3317\_\_AcquisitionAutoRefreshStopAndRelease

## Description

Stops the current auto refresh acquisition

## Parameters

- ◆ [Query frame layout] **ulOption1** Reserved. Set to 0

## Returns:

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

- ◆ **0** The remote function performed OK
- ◆ **-100** 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	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	15850	0xEA3D	0x3DEA
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption1	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	15850	0xEA3D	0x3DEA
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

# MSXE3317\_\_AcquisitionSequenceInitAndS

## Description

Initialise and start the sequence acquisition mode

### Parameters

- ◆ [Query frame layout] ***ulNbrOfChannel*** Number of channel in the sequence
- ◆ [Query frame layout] ***psChannelList*** List of the channel who compose the sequence.
- ◆ [Query frame layout] ***ulAcquisitionTime*** Acquisition Time
  - ◇ range from min 1000 to 65535 when the unit is the microsecond
  - ◇ range from min 1 to 65535 when the unit is the millisecond
  - ◇ range from min 1 to 65535 when the unit is the second
- ◆ [Query frame layout] ***ulAcquisitionTimeUnit*** Acquisition Time Unit
  - ◇ 0 : us
  - ◇ 1 : ms
  - ◇ 2 : s
- ◆ [Query frame layout] ***ulNbrOfSequence*** Number of sequence to acquire
  - ◇ 0 : continuous mode
  - ◇ Superior to 0 : number of sequence
- ◆ [Query frame layout] ***ulNbrMaxSequenceToTransfer*** Max number of sequence to acquire before a data transfer. (from 1 to 65535)
- ◆ [Query frame layout] ***ulTriggerMask*** Define the source of the trigger
  - ◇ 0 : trigger disabled
  - ◇ 1 : Enable Hardware Digital Input Trigger
  - ◇ 2 : Enable Synchro Trigger
  - ◇ 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/Cosine input)
  - ◇ 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- ◆ [Query frame layout] ***ulTriggerMode*** Define the trigger mode
  - ◇ 1 : One shot trigger
  - ◇ 2 : Sequence trigger
- ◆ [Query frame layout] ***ulHardwareTriggerEdge*** Define the edge of the hardware trigger who generate a trigger action



## MODBUS interface description

- ◇ 1 : rising front (Only if hardware trigger selected)
- ◇ 2 : falling front (Only if hardware trigger selected)
- ◇ 3 : Both front (Only if hardware trigger selected)
- ◆ [Query frame layout] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
  - ◇ 1 : all trigger event start the action
  - ◇ max value : 65535
- ◆ [Query frame layout] ***ulByTriggerNbrOfSeqToAcquire*** define the number of sequence to acquire by each trigger event
  - ◇ 0 : continuous mode
  - ◇ Superior to 0 : number of sequence : (1..0xFFFFFFFF)
- ◆ [Query frame layout] ***ulDataFormat*** Data format option
  - ◇ D0 : Absolute time stamp information
    - 0: no time stamp information
    - 1: time stamp information
  - ◇ D1 : Relative time stamp information
    - 0: no time stamp information
    - 1: time stamp information
  - ◇ D2 : Sequence counter information
    - 0 : No sequence counter information
    - 1 : Sequence counter information
  - ◇ D3 : System status information
    - 0 : No system status information required
    - 1 : System status information required
  - ◇ D4 : Data format
    - 0: Digital value
    - 1: Analog value
- ◆ [Query frame layout] ***ulForceStart***
  - ◇ 0 : Function return a error if any acquisition already in progress
  - ◇ 1 : If any acquisition in progress then stop this
- ◆ [Query frame layout] ***ulOption1*** Reserved. Set to 0
- ◆ [Query frame layout] ***ulOption2*** Reserved. Set to 0
- ◆ [Query frame layout] ***ulOption3*** Reserved. Set to 0

### Returns:

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

- ◆ 0 The remote function performed OK
- ◆ -2 Any acquisition already in progress
- ◆ -3 The number of channel in the sequence is null or too high
- ◆ -4 Channel index selection error
- ◆ -5 Channel already selected
- ◆ -6 Any selected channel not OK, call the diagnostic function for more information
- ◆ -7 Not available acquisition time unit
- ◆ -8 The minimal acquisition time is 1000 us !
- ◆ -9 The maximal acquisition time is 65535 !

## MODBUS interface description

- ◆ **-10** Transfer sequence size error (1 to 4096) !
- ◆ **-11** The total number of sequences is not a multiple from number of sequences to transfer
- ◆ **-12** Trigger mask not available
- ◆ **-13** Trigger mask : 2 different trigger source cannot be simultaneously be activated
- ◆ **-14** Trigger mode not available
- ◆ **-15** Trigger mask : 2 trigger mode cannot be simultaneously be activated
- ◆ **-16** Hardware trigger : front definition error
- ◆ **-17** Hardware trigger count value not available
- ◆ **-18** Number of sequence to acquire by trigger mode not available
- ◆ **-19** Data format not available
- ◆ **-20** Selected channels combination not available
- ◆ **-100** Internal system error occurred. See value of syserrno

## Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000
protocol identifier	2	16-bit integer	0	0x0000
length	2	16-bit integer	131	0x8300
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10
Reference number (=register)	2	16-bit integer	15900	0x1C3E
word count	2	16-bit integer	62	0x3E00
byte count	1	8-bit integer	124	0x7C
ulNbrOfChannel	4	32-bit integer	See the description above	0x????????
ulChannelList	64	32-bit integer array	See the description above	0x????????[16
ulAcquisitionTime	4	32-bit integer	See the description above	0x????????
ulAcquisitionTimeUnit	4			0x????????

## MODBUS interface description

		32-bit integer	See the description above	
ulNbrOfSequence	4	32-bit integer	See the description above	0x????????
ulNbrMaxSequenceToTransfer	4	32-bit integer	See the description above	0x????????
ulTriggerMask	4	32-bit integer	See the description above	0x????????
ulTriggerMode	4	32-bit integer	See the description above	0x????????
ulHardwareTriggerEdge	4	32-bit integer	See the description above	0x????????
ulHardwareTriggerCount	4	32-bit integer	See the description above	0x????????
ulByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????
ulDataFormat	4	32-bit integer	See the description above	0x????????
ulForceStart	4	32-bit integer	See the description above	0x????????
ulOption1	4	32-bit integer	See the description above	0x????????
ulOption2	4	32-bit integer	See the description above	0x????????
ulOption3	4	32-bit integer	See the description above	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 -	0x0000	0x0000

## MODBUS interface description

			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	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	15900	0x1C3E	0x3E1C
word count	2	16-bit integer	62	0x3E00	0x003E

## 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

## MSXE3317\_\_AcquisitionSequenceStopAnd

## Description

Stop and release the sequence acquisition mode

## Parameters

- ♦ [Query frame layout] **ulOption1** Reserved. Set to 0

## MODBUS interface description

### Returns:

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

- ◆ **0** The remote function performed OK
- ◆ **-2** No sequence acquisition in progress
- ◆ **-100** 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	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	15950	0x4E3E	0x3E4E
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

## MODBUS interface description

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	15950	0x4E3E	0x3E4E
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

### MSXE3317\_\_DigitalIOSetPortDirections

## Description

Write the current digital I/O direction for the selected port.

## Parameters

- ◆ [Query frame layout] **ulPort** Selected digital I/O port (0 to MSXE3317\_\_DigitalIOGetNumberOfPorts).

## MODBUS interface description

- ◆ [Query frame layout] **ulDirection** Digital I/O direction. Each bit indicates if the channel is used as an input or an output. (0: input, 1: output)
- ◆ [Query frame layout] **ulOption1** Reserved. Set to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The ulPort parameter is wrong
- ◆ **-3** The ulDirection parameter is wrong
- ◆ **-100** 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	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	26500	0x8467	0x6784
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulPort	4	32-bit integer	See the description above	0x????????	0x????????
ulDirection	4	32-bit integer	See the description above	0x????????	0x????????
ulOption1	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	26500	0x8467	0x6784
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

# MSXE3317\_\_DigitalIOSetInputsFilterTime

## Description

Sets the filter time for the digital inputs in steps of 250 ns (max value: 16777215)

## Parameters

- ◆ [Query frame layout] **ulFilterTime** Filter time for the digital inputs in steps of 250 ns (max value: 16777215)
  - ◇ **0**: Disable the filter
  - ◇ **1**: Sets the filter time to 250 ns
  - ◇ **2**: Sets the filter time to 500 ns
  - ◇ ...
  - ◇ **16777215**: Sets the filter time to 4 s
- ◆ [Query frame layout] **ulOption1** Reserved. Set to 0

## Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The ulFilterTime parameter is wrong
- ◆ **-100** 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	26550	0xB667	0x67B6

## MODBUS interface description

word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulFilterTime	4	32-bit integer	See the description above	0x????????	0x????????
ulOption1	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	26550	0xB667	0x67B6
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

## 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

### MSXE3317\_\_DigitalIOEnableDisableInputs

## Description

Enables/disables the digital input filter for the selected port.

## Parameters

- ◆ [Query frame layout] **ulPort** Selected digital I/O port (0 to MSXE3317\_\_DigitalIOGetNumberOfPorts).
- ◆ [Query frame layout] **ulFilter** Digital input filter selection. Each bit indicates if the filter is enabled on the input. (0: disabled, 1: enabled)
- ◆ [Query frame layout] **ulOption1** Reserved. Set to 0

## Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The ulPort parameter is wrong
- ◆ **-3** The ulFilterTime parameter is wrong
- ◆ **-4** Any selected input is not an input or a bidirectional channel
- ◆ **-100** 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

## MODBUS interface description

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	26600	0xE867	0x67E8
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulPort	4	32-bit integer	See the description above	0x???????	0x???????
ulFilter	4	32-bit integer	See the description above	0x???????	0x???????
ulOption1	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	26600	0xE867	0x67E8
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

### MSXE3317\_\_DigitalIORearmOutputsShortCircuit

## Description

Rearm the digital outputs short circuit.

Please use only this function if you detected a short circuit using the function MSXE3317\_\_DigitalIOTestOutputsShortCircuit.

## Parameters

- ◆ [Query frame layout] **ulOption1** Reserved. Set to 0

## Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-100** Internal system error occurred. See value of syserrno

## Query 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	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	26650	0x1A68	0x681A
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption1	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	26650	0x1A68	0x681A
word count	2		2	0x0200	0x0002

## MODBUS interface description

		16-bit 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

### MSXE3317\_\_DigitalIOWriteChannel

### Description

Set the selected digital output channel to on or off.

#### Parameters

- ◆ [Query frame layout] ***ulChannel*** Selected digital I/O channel (0 to MSXE3317\_\_DigitalIOGetNumberOfChannels).
- ◆ [Query frame layout] ***ulState*** Digital I/O channel state (0: off, 1: on)
- ◆ [Query frame layout] ***ulOption1*** Reserved. Set to 0

#### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The ulChannel parameter is wrong
- ◆ **-3** The ulState parameter is wrong
- ◆ **-4** The selected digital I/O is not an output

## MODBUS interface description

♦ -100 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	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	26700	0x4C68	0x684C
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannel	4	32-bit integer	See the description above	0x???????	0x???????
ulState	4	32-bit integer	See the description above	0x???????	0x???????
ulOption1	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
	2		0	0x0000	0x0000



## 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	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	26700	0x4C68	0x684C
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 MSXE3317\_\_DigitalIOWritePort

### Description

Write a value to the selected digital I/O port.

### Parameters

- ◆ [Query frame layout] **ulPort** Selected digital I/O port (0 to MSXE3317\_\_DigitalIOGetNumberOfPorts).

## MODBUS interface description

- ◆ [Query frame layout] **ulState** Digital I/O state. Each bit set the state for one digital I/O channel (0: off, 1: on)
- ◆ [Query frame layout] **ulOption1** Reserved. Set to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The ulPort parameter is wrong
- ◆ **-3** The ulState parameter is wrong
- ◆ **-4** Any digital I/O set to 1 is not an output channel
- ◆ **-100** 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	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	26750	0x7E68	0x687E
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulPort	4	32-bit integer	See the description above	0x????????	0x????????
ulState	4	32-bit integer	See the description above	0x????????	0x????????
ulOption1	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	26750	0x7E68	0x687E
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 MSXE3317\_\_SinCosInit

### Description

Initialise the selected Sine/Cosine channel.

### Parameters

- ◆ [Query frame layout] **ulChannel** Channel selection (0 to SinCosGetNumberOfChannels)
- ◆ [Query frame layout] **fSignalPeriod** Signal period (Unit/period)
- ◆ [Query frame layout] **ulResolution** Resolution to use for the measure (steps/period)
- ◆ [Query frame layout] **ulOption1** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption2** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption3** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption4** Reserved. Set it to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The PLD is not working
- ◆ **-3** The ulChannel parameter is wrong
- ◆ **-4** The current status is not correct. The selected Sine/Cosine channel has already been initialised. Please call the SinCosRelease and retry
- ◆ **-5** The fSignalPeriod parameter is wrong
- ◆ **-6** The ulResolution parameter is wrong
- ◆ **-7** Gain calibration error occurred. Please check the connection with the sensor
- ◆ **-21** Timeout while initialising the Sine/Cosine sensor. Please check the connection with the sensor
- ◆ **-100** 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		35	0x2300	0x0023

## 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	37200	0x5091	0x9150
word count	2	16-bit integer	14	0x0E00	0x000E
byte count	1	8-bit integer	28	0x1C	0x1C
ulChannel	4	32-bit integer	See the description above	0x????????	0x????????
fSignalPeriod	4	32-bit floating point	See the description above	0x????????	0x????????
ulResolution	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

## 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	37200	0x5091	0x9150
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 MSXE3317\_\_SinCosRelease

### Description

Release the selected Sine / Cosine channel.

### Parameters

- ◆ [Query frame layout] ***ulChannel*** Channel selection (0 to SinCosGetNumberOfChannels)
- ◆ [Query frame layout] ***ulOption1*** Reserved. Set it to 0
- ◆ [Query frame layout] ***ulOption2*** Reserved. Set it to 0

### Returns

## MODBUS interface description

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The PLD is not working
- ◆ **-3** The ulChannel parameter is wrong
- ◆ **-4** The current status is not correct. The selected Sine/Cosine channel is not initialised
- ◆ **-100** 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	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	37300	0xB491	0x91B4
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannel	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)
-------	--------------	------	-------	-----------------------	-----------------------

## 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	37300	0xB491	0x91B4
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 MSXE3317\_\_SinCosClear



## Description

Clear the measured value of the selected Sine/Cosine channel.

### Parameters

- ◆ [Query frame layout] **ulChannel** Channel selection (0 to SinCosGetNumberOfChannels)
- ◆ [Query frame layout] **ulOption1** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption2** Reserved. Set it to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The PLD is not working
- ◆ **-3** The ulChannel parameter is wrong
- ◆ **-4** The current status is not correct. The selected Sine/Cosine channel is not initialised
- ◆ **-100** 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	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	37400	0x1892	0x9218
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannel	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????????

## 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	37400	0x1892	0x9218
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 interface description

MODBUS Function code	1	8-bit integer	0x90	0x90	0x90
Exception code	1	8-bit integer	See corresponding chapter	0x??	0x??

## Function MSXE3317\_\_SinCosClearError

### Description

Clear the error flag of the selected Sine/Cosine channel.

### Parameters

- ◆ [Query frame layout] **ulChannel** Channel selection (0 to SinCosGetNumberOfChannels)
- ◆ [Query frame layout] **ulOption1** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption2** Reserved. Set it to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The PLD is not working
- ◆ **-3** The ulChannel parameter is wrong
- ◆ **-4** The current status is not correct. The selected Sine/Cosine channel is not initialised
- ◆ **-100** 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	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
	1		0x10	0x10	0x10

## MODBUS interface description

MODBUS Function code		8-bit integer			
Reference number (=register)	2	16-bit integer	37500	0x7C92	0x927C
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannel	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	37500	0x7C92	0x927C
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 MSXE3317\_\_SinCosInitIndex

### Description

Initialise the Index configuration. The index function enables to generate an event, or do an action, when the index signal occurs. This signal occurs one time per rotation for a rotary encoder, and one time per ride for a linear sensor (for more information see documentation of your sensor).

### Parameters

- ◆ [Query frame layout] ***ulChannel*** Channel selection (0 to SinCosGetNumberOfChannels)
- ◆ [Query frame layout] ***ulEdge*** Edge selection
  - ◇ **0b01 = 1**: Rising edge
  - ◇ **0b10 = 2**: Falling edge
  - ◇ **0b11 = 3**: Both edges
- ◆ [Query frame layout] ***ulAction*** Action to do when the index trigger occurs
  - ◇ **0b00 = 0**: Do nothing (but index trigger can also be used to trigger the acquisition)
  - ◇ **0b01 = 1**: Clear the value of the sensor
  - ◇ **0b10 = 2**: Generate a synchro trigger

## MODBUS interface description

◇ **0b11 = 3**: Clear the value of the sensor AND  
Generate a synchro trigger

- ◆ [Query frame layout] **ulOption1** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption2** Reserved. Set it to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The PLD is not working
- ◆ **-3** The ulChannel parameter is wrong
- ◆ **-4** The ulEdge parameter is wrong
- ◆ **-5** The ulAction parameter is wrong
- ◆ **-6** The current status is not correct. The selected Sine/Cosine channel is not initialised
- ◆ **-7** The current status is not correct. The selected Sine/Cosine channel Index Logic has already been initialised. Please call the function SinCosReleaseIndex
- ◆ **-100** 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	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	37600	0xE092	0x92E0
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulChannel	4	32-bit integer	See the description above	0x???????	0x???????

## MODBUS interface description

ulEdge	4	32-bit integer	See the description above	0x????????	0x????????
ulAction	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	37600	0xE092	0x92E0
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

## 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 MSXE3317\_\_SinCosReleaseIndex

### Description

Release the Index configuration.

### Parameters

- ◆ [Query frame layout] **ulChannel** Channel selection (0 to SinCosGetNumberOfChannels)
- ◆ [Query frame layout] **ulOption1** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption2** Reserved. Set it to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The PLD is not working
- ◆ **-3** The ulChannel parameter is wrong
- ◆ **-4** The current status is not correct. The selected Sine/Cosine channel Index Logic is not initialised
- ◆ **-100** 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		19	0x1300	0x0013



## 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	37700	0x4493	0x9344
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannel	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	37700	0x4493	0x9344
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

### MSXE3317\_\_SinCosInitAndEnableCompare

## Description

Initialise and enable a Sine/Cosine Compare Logic.

### Parameters

- ◆ [Query frame layout] **ulChannel** Channel selection (0 to SinCosGetNumberOfChannels)
- ◆ [Query frame layout] **fValue** Compare value. Possible value depends on the variable ulValueFormat
  - ◇ If **ulValueFormat** is set to 0: 0 to 0xFFFFFFFF
  - ◇ If **ulValueFormat** is set to 1: Standardised value (Unit). The value cannot be negative
- ◆ [Query frame layout] **ulValueFormat** Format of the value fValue
  - ◇ 0: Raw value
  - ◇ 1: Standardised value. The result of the operation dValue (ulResolution / fSignalPeriod) must be an integer
- ◆ [Query frame layout] **ulMode** Compare mode
  - ◇ 0: Simple mode. As soon as the counter value corresponds to the compare value (dValue), a trigger or synchro trigger is released.

## MODBUS interface description

- ◇ **1**: Modulo mode. When the counter value corresponds to the compare value (dValue) or a multiple of it, a trigger or synchro trigger is released. In that mode, the value dValue cannot be 0.
- ◆ [Query frame layout] **ulSynchroTrigger** Generate synchro trigger, which can be used to trigger another MSX-E system, when the compare trigger is released
  - ◇ **0**: No
  - ◇ **1**: Yes
- ◆ [Query frame layout] **ulOption1** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption2** Reserved. Set it to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The PLD is not working
- ◆ **-3** The ulChannel parameter is wrong
- ◆ **-4** The ulValueFormat parameter is wrong
- ◆ **-5** The ulMode parameter is wrong
- ◆ **-6** The ulMode parameter is wrong
- ◆ **-7** In Modulo mode, the parameter fValue cannot be 0
- ◆ **-8** The fValue parameter is wrong
- ◆ **-9** The current status is not correct. The selected Sine/Cosine channel is not initialised
- ◆ **-10** The current status is not correct. The selected Sine/Cosine channel Compare Logic has already been initialised. Please call the function SinCosDisableAndReleaseCompareLogic
- ◆ **-100** 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	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
	2		37800	0xA893	0x93A8

## MODBUS interface description

Reference number (=register)		16-bit integer			
word count	2	16-bit integer	14	0x0E00	0x000E
byte count	1	8-bit integer	28	0x1C	0x1C
ulChannel	4	32-bit integer	See the description above	0x????????	0x????????
fValue	4	32-bit floating point	See the description above	0x????????	0x????????
ulValueFormat	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

## MODBUS interface description

Reference number (=register)	2	16-bit integer	37800	0xA893	0x93A8
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

## MSXE3317\_\_SinCosDisableAndReleaseCo

## Description

Disable and Release the Sine/Cosine Compare Logic.

### Parameters

- ◆ [Query frame layout] **ulChannel** Channel selection (0 to SinCosGetNumberOfChannels)
- ◆ [Query frame layout] **ulOption1** Reserved. Set it to 0
- ◆ [Query frame layout] **ulOption2** Reserved. Set it to 0

### Returns

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

- ◆ **0** The remote function performed OK
- ◆ **-2** The PLD is not working
- ◆ **-3** The ulChannel parameter is wrong

## MODBUS interface description

- ◆ **-4** The current status is not correct. The selected Sine/Cosine channel Compare Logic is not initialised
- ◆ **-100** 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	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	37900	0x0C94	0x940C
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannel	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

## 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	37900	0x0C94	0x940C
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??

# 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
MODBUS_ILLEGAL_FUNCTION	0x1
MODBUS_ILLEGAL_DATA_ADDRESS	0x2
MODBUS_ILLEGAL_DATA_VALUE	0x3
MODBUS_ILLEGAL_DATA_RESPONSE_LENGTH	0x4
MODBUS_ACKNOWLEDGE	0x5
MODBUS_DSLAVE_DEVICE_BUSY	0x6
MODBUS_NEGATIVE_ACKNOWLEDGE	0x7
MODBUS_MEMORY_PARITY_ERROR	0x8
MODBUS_REMOTE_EXECUTION_ERROR	0x9
MODBUS_GATEWAY_PATH_UNAVAILABLE	0xA
MODBUS_GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND	0xB

# 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	Rename
MXCommon__GetModuleType	GetModuleType
MXCommon__GetTime	GetTime
MXCommon__TestCustomerID	TestCustomerID
MSXE3317__PressureGetNumberOfChannels	PressGetNbChan
MSXE3317__PressureCalibrationGetCurrentStatus	PressCalibStatus
MSXE3317__AcquisitionGetNumberOfChannels	AcqGetNbChan
MSXE3317__AcquisitionGetChannelsInfo	AcqGetChanInfo
MSXE3317__AcquisitionAutoRefreshGetValues	AcqGetAutoRefV
MSXE3317__AcquisitionAutoRefreshGetBlockingValues	AcqGetAutoRefBL
MSXE3317__AcquisitionAutoRefreshGetConfiguration	AcqAutoRefGetCo
ulByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSe
MSXE3317__AcquisitionSequenceGetConfiguration	AcqSequenceGetC
ulNbrMaxSequenceToTransfer	NbrMaxSequenc
ulByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSe
MSXE3317__DigitalIOGetNumberOfChannels	DigIOGetNbChan
MSXE3317__DigitalIOGetNumberOfPorts	DigIOGetNbPorts
MSXE3317__DigitalIOGetPortAvailableDirections	DigIOGetAvDir
MSXE3317__DigitalIOGetPortDirections	DigIOGetDir
MSXE3317__DigitalIOGetInputsFilterConfiguration	DigIOGetFiltCfg
MSXE3317__DigitalIOTestOutputsShortCircuit	DigIOTestSC
MSXE3317__DigitalIOReadChannel	DigIORdChl
MSXE3317__DigitalIOReadPort	DigIORdPort
MSXE3317__SinCosGetNumberOfChannels	SinCosGetNbChar
MSXE3317__SinCosRead	SinCosRead
MXCommon__SetHardwareTriggerFilterTime	SetHwTrigFiltTim
MXCommon__InitAndStartSynchroTimer	InitStartSyncTime
MXCommon__StopAndReleaseSynchroTimer	StopRelSyncTime
MXCommon__Reboot	Reboot
MXCommon__SetCustomerKey	SetCustomerKey
MSXE3317__SetDataCursor	SetCursor
MSXE3317__PressureSetChannelConfiguration	PressSetChannelC
MSXE3317__PressureSetSamplingRate	PressSetSampling
MSXE3317__PressureCalibrationStart	PressCalibStart
MSXE3317__PressureCalibrationNextStep	PressCalibNext

## MODBUS interface description

MSXE3317__PressureCalibrationBreak	PressCalibBreak
MSXE3317__AcquisitionAutoRefreshInitAndStart	AcqAutoRefStart
ulByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSe
MSXE3317__AcquisitionAutoRefreshStopAndRelease	AcqAutoRefStop
MSXE3317__AcquisitionSequenceInitAndStart	AcqSequenceStart
ulNbrMaxSequenceToTransfer	NbrMaxSequenc
ulByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSe
MSXE3317__AcquisitionSequenceStopAndRelease	AcqSequenceStop
MSXE3317__DigitalIOSetPortDirections	DigIOSetDir
MSXE3317__DigitalIOSetInputsFilterTime	DigIOSetFiltTime
MSXE3317__DigitalIOEnableDisableInputsFilter	DigIOEnaDisFilt
MSXE3317__DigitalIORearmOutputsShortCircuit	DigIORearmSC
MSXE3317__DigitalIOWriteChannel	DigIOWrChl
MSXE3317__DigitalIOWritePort	DigIOWrPort
MSXE3317__SinCosInit	SinCosInit
MSXE3317__SinCosRelease	SinCosRelease
MSXE3317__SinCosClear	SinCosClear
MSXE3317__SinCosClearError	SinCosClearError
MSXE3317__SinCosInitIndex	SinCosInitIndex
MSXE3317__SinCosReleaseIndex	SinCosReleaseInde
MSXE3317__SinCosInitAndEnableCompareLogic	SinCosInitCompL
MSXE3317__SinCosDisableAndReleaseCompareLogic	SinCosReleaseCor