

MODBUS interface description

Table of Contents

General description	1
Introduction	1
Why a MODBUS Server on the MSX-E modules?	1
Technical details	1
FC3 (read multiple register) Functions	3
Function <u>GetLastCommandStatus</u>	3
For new application(s) or automate communication it is recommended to use the function	
<u>GetLastCommandStatusEx</u>	4
Description	4
Query frame layout	4
Response frame layout	4
Exception frame layout	5
Function <u>GetLastCommandStatusEx</u>	6
Description	6
Query frame layout	6
Response frame layout	6
Exception frame layout	7
Function <u>MXCommon GetModuleType</u>	8
For new application(s) or automate communication it is recommended to use the function	
<u>MXCommon GetModuleTypeEx</u>	8
Description	8
Query frame layout	8
Response frame layout	8
Exception frame layout	9
Function <u>MXCommon GetModuleTypeEx</u>	9
Description	9
Query frame layout	10
Response frame layout	10
Exception frame layout	11
Function <u>MXCommon GetTime</u>	11
For new application(s) or automate communication it is recommended to use the function	
<u>MXCommon GetTimeEx</u>	11
Description	11
Query frame layout	11
Response frame layout	12
Exception frame layout	12
Function <u>MXCommon GetTimeEx</u>	13
Description	13
Query frame layout	13
Response frame layout	14
Exception frame layout	14
Function <u>MXCommon TestCustomerID</u>	15
For new application(s) or automate communication it is recommended to use the function	
<u>MXCommon TestCustomerIDEx</u>	15
Description	15
Query frame layout	15
Response frame layout	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>	17
<u>Exception frame layout</u>	18
<u>Function MSXE17xx MFIncCounterRead32BitsValueAll</u>	18
<u>Description</u>	19
<u>Query frame layout</u>	19
<u>Response frame layout</u>	20
<u>Exception frame layout</u>	20
<u>Function MSXE17xx MFIncCounterRead32BitsValue0</u>	21
<u>Description</u>	21
<u>Query frame layout</u>	22
<u>Response frame layout</u>	22
<u>Exception frame layout</u>	23
<u>Function MSXE17xx MFIncCounterRead32BitsValue1</u>	23
<u>Description</u>	23
<u>Query frame layout</u>	24
<u>Response frame layout</u>	24
<u>Exception frame layout</u>	25
<u>Function MSXE17xx MFIncCounterRead32BitsValue2</u>	25
<u>Description</u>	25
<u>Query frame layout</u>	26
<u>Response frame layout</u>	27
<u>Exception frame layout</u>	27
<u>Function MSXE17xx MFIncCounterRead32BitsValue3</u>	28
<u>Description</u>	28
<u>Query frame layout</u>	28
<u>Response frame layout</u>	29
<u>Exception frame layout</u>	29
<u>Function MSXE17xx DigitalIOReadAllChannelsValue</u>	30
<u>Description</u>	30
<u>Query frame layout</u>	30
<u>Response frame layout</u>	31
<u>Exception frame layout</u>	31
<u>Function MSXE17xx DigitalIOTestShortCircuit</u>	32
<u>Description</u>	32
<u>Query frame layout</u>	32
<u>Response frame layout</u>	33
<u>Exception frame layout</u>	33
<u>Function MSXE17xx IOWatchdogGetStatusAndValue</u>	34
<u>Description</u>	34
<u>Query frame layout</u>	34
<u>Response frame layout</u>	35
<u>Exception frame layout</u>	35
<u>Function MSXE170x DigitalIOReadAllChannelsValue</u>	36

Table of Contents

FC3 (read multiple register) Functions

<u>Description</u>	36
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u> MSXE17xx DigitalIOReadAllChannelsValue</u>	36
<u>Query frame layout</u>	36
<u>Response frame layout</u>	37
<u>Exception frame layout</u>	37

FC16 (write multiple register) Functions.....39

<u>Function MXCommon SetHardwareTriggerFilterTime</u>	41
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u> MXCommon SetHardwareTriggerFilterTimeEx</u>	41
<u>Description</u>	41
<u>Query frame layout</u>	41
<u>Response frame layout</u>	42
<u>Exception frame layout</u>	43
<u>Function MXCommon SetHardwareTriggerFilterTimeEx</u>	43
<u>Description</u>	43
<u>Query frame layout</u>	44
<u>Response frame layout</u>	44
<u>Exception frame layout</u>	45
<u>Function MXCommon InitAndStartSynchroTimer</u>	45
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u> MXCommon InitAndStartSynchroTimerEx</u>	45
<u>Description</u>	46
<u>Query frame layout</u>	47
<u>Response frame layout</u>	48
<u>Exception frame layout</u>	48
<u>Function MXCommon InitAndStartSynchroTimerEx</u>	49
<u>Description</u>	49
<u>Query frame layout</u>	50
<u>Response frame layout</u>	51
<u>Exception frame layout</u>	52
<u>Function MXCommon StopAndReleaseSynchroTimer</u>	52
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u> MXCommon StopAndReleaseSynchroTimerEx</u>	52
<u>Description</u>	52
<u>Query frame layout</u>	53
<u>Response frame layout</u>	53
<u>Exception frame layout</u>	54
<u>Function MXCommon StopAndReleaseSynchroTimerEx</u>	54
<u>Description</u>	54
<u>Query frame layout</u>	55
<u>Response frame layout</u>	55
<u>Exception frame layout</u>	56
<u>Function MXCommon Reboot</u>	56
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u> MXCommon RebootEx</u>	56

Table of Contents

FC16 (write multiple register) Functions

<u>Description</u>	56
<u>Query frame layout</u>	57
<u>Response frame layout</u>	57
<u>Exception frame layout</u>	58
<u>Function MXCommon RebootEx</u>	58
<u>Description</u>	58
<u>Query frame layout</u>	59
<u>Response frame layout</u>	59
<u>Exception frame layout</u>	60
<u>Function MXCommon SetCustomerKey</u>	60
For new application(s) or automate communication it is recommended to use the function	
<u>MXCommon SetCustomerKeyEx</u>	60
<u>Description</u>	61
<u>Query frame layout</u>	61
<u>Response frame layout</u>	62
<u>Exception frame layout</u>	62
<u>Function MXCommon SetCustomerKeyEx</u>	63
<u>Description</u>	63
<u>Query frame layout</u>	63
<u>Response frame layout</u>	64
<u>Exception frame layout</u>	64
<u>Function MXCommon SetFilterChannels</u>	65
For new application(s) or automate communication it is recommended to use the function	
<u>MXCommon SetFilterChannelsEx</u>	65
<u>Description</u>	65
<u>Query frame layout</u>	65
<u>Response frame layout</u>	66
<u>Exception frame layout</u>	67
<u>Function MXCommon SetFilterChannelsEx</u>	67
<u>Description</u>	67
<u>Query frame layout</u>	68
<u>Response frame layout</u>	68
<u>Exception frame layout</u>	69
<u>Function MSXE17xx MFCommonSetInputsFilter</u>	69
<u>Description</u>	69
<u>Query frame layout</u>	70
<u>Response frame layout</u>	71
<u>Exception frame layout</u>	72
<u>Function MSXE17xx MFCommonReferenceVoltageActivation</u>	72
<u>Description</u>	72
<u>Query frame layout</u>	73
<u>Response frame layout</u>	74
<u>Exception frame layout</u>	74
<u>Function MSXE17xx MFCommonSetFIFO0Level</u>	75
<u>Description</u>	75
<u>Query frame layout</u>	75
<u>Response frame layout</u>	76

Table of Contents

FC16 (write multiple register) Functions

Exception frame layout.....	77
Function MSXE17xx MFCommonEnableDisableTriggerGate.....	77
Description.....	77
Query frame layout.....	78
Response frame layout.....	79
Exception frame layout.....	79
Function MSXE17xx MFIIncCounterInit.....	80
Description.....	80
Query frame layout.....	81
Response frame layout.....	82
Exception frame layout.....	82
Function MSXE17xx MFIIncCounterRelease.....	83
Description.....	83
Query frame layout.....	83
Response frame layout.....	83
Exception frame layout.....	84
Function MSXE17xx MFIIncCounterClear.....	84
Description.....	85
Query frame layout.....	85
Response frame layout.....	85
Exception frame layout.....	86
Function MSXE17xx MFIIncCounterWrite32BitValue.....	86
Description.....	87
Query frame layout.....	87
Response frame layout.....	88
Exception frame layout.....	88
Function MSXE17xx MFIIncCounterInitAndEnableCompareLogic.....	89
Description.....	89
Query frame layout.....	90
Response frame layout.....	91
Exception frame layout.....	91
Function MSXE17xx MFIIncCounterDisableAndReleaseCompareLogic.....	92
Description.....	92
Query frame layout.....	92
Response frame layout.....	93
Exception frame layout.....	94
Function MSXE17xx MFIIncCounterInitHardwareTrigger.....	94
Description.....	94
Query frame layout.....	95
Response frame layout.....	96
Exception frame layout.....	96
Function MSXE17xx MFIIncCounterReleaseHardwareTrigger.....	97
Description.....	97
Query frame layout.....	97
Response frame layout.....	98
Exception frame layout.....	98
Function MSXE17xx MFIIncCounterInitIndex.....	99

Table of Contents

FC16 (write multiple register) Functions

<u>Description</u>	99
<u>Query frame layout</u>	100
<u>Response frame layout</u>	100
<u>Exception frame layout</u>	101
<u>Function MSXE17xx MFIncCounterReleaseIndex</u>	101
<u>Description</u>	101
<u>Query frame layout</u>	102
<u>Response frame layout</u>	103
<u>Exception frame layout</u>	103
<u>Function MSXE17xx MFIncCounterInitAndEnableLatch</u>	104
<u>Description</u>	104
<u>Query frame layout</u>	105
<u>Response frame layout</u>	106
<u>Exception frame layout</u>	107
<u>Function MSXE17xx MFIncCounterDisableAndReleaseLatch</u>	107
<u>Description</u>	107
<u>Query frame layout</u>	108
<u>Response frame layout</u>	109
<u>Exception frame layout</u>	109
<u>Function MSXE17xx MFIncCounterInitAndEnableClear</u>	110
<u>Description</u>	110
<u>Query frame layout</u>	111
<u>Response frame layout</u>	112
<u>Exception frame layout</u>	112
<u>Function MSXE17xx MFIncCounterDisableAndReleaseClear</u>	113
<u>Description</u>	113
<u>Query frame layout</u>	113
<u>Response frame layout</u>	114
<u>Exception frame layout</u>	115
<u>Function MSXE17xx MFIncCounterInitAndEnableFrequencyMeasurement</u>	115
<u>Description</u>	115
<u>Query frame layout</u>	117
<u>Response frame layout</u>	117
<u>Exception frame layout</u>	118
<u>Function MSXE17xx MFIncCounterDisableAndReleaseFrequencyMeasurement</u>	118
<u>Description</u>	118
<u>Query frame layout</u>	119
<u>Response frame layout</u>	120
<u>Exception frame layout</u>	120
<u>Function MSXE17xx DigitalIOWriteAllChannelsValue</u>	121
<u>Description</u>	121
<u>Query frame layout</u>	121
<u>Response frame layout</u>	122
<u>Exception frame layout</u>	122
<u>Function MSXE17xx DigitalIORearmShortCircuit</u>	123
<u>Description</u>	123
<u>Query frame layout</u>	123

Table of Contents

FC16 (write multiple register) Functions

<u>Response frame layout</u>	124
<u>Exception frame layout</u>	124
<u>Function MSXE17xx DigitalIOInitPort</u>	125
<u>Description</u>	125
<u>Query frame layout</u>	125
<u>Response frame layout</u>	126
<u>Exception frame layout</u>	126
<u>Function MSXE17xx IOWatchdogInitAndStart</u>	127
<u>Description</u>	127
<u>Query frame layout</u>	127
<u>Response frame layout</u>	128
<u>Exception frame layout</u>	129
<u>Function MSXE17xx IOWatchdogStopAndRelease</u>	129
<u>Description</u>	129
<u>Query frame layout</u>	130
<u>Response frame layout</u>	130
<u>Exception frame layout</u>	131
<u>Function MSXE170x MFIncCounterInit</u>	131
<u>Description</u>	131
<u>For new application(s) or automat communication is it recommended tu use the function</u> <u>MSXE17xx MFIncCounterInit</u>	131
<u>Query frame layout</u>	132
<u>Response frame layout</u>	133
<u>Exception frame layout</u>	134
<u>Function MSXE170x MFIncCounterRelease</u>	134
<u>Description</u>	134
<u>For new application(s) or automat communication is it recommended tu use the function</u> <u>MSXE17xx MFIncCounterRelease</u>	134
<u>Query frame layout</u>	135
<u>Response frame layout</u>	135
<u>Exception frame layout</u>	136
<u>Function MSXE170x MFIncCounterClear</u>	136
<u>Description</u>	136
<u>For new application(s) or automat communication is it recommended tu use the function</u> <u>MSXE17xx MFIncCounterClear</u>	136
<u>Query frame layout</u>	137
<u>Response frame layout</u>	137
<u>Exception frame layout</u>	138
<u>Function MSXE170x MFIncCounterInitAndEnableLatchRegister</u>	138
<u>Description</u>	138
<u>For new application(s) or automat communication is it recommended tu use the function</u> <u>MSXE17xx MFIncCounterInitAndEnableLatch</u>	138
<u>Query frame layout</u>	139
<u>Response frame layout</u>	140
<u>Exception frame layout</u>	141
<u>Function MSXE170x MFIncCounterDisableAndReleaseLatchRegister</u>	141
<u>Description</u>	141

Table of Contents

FC16 (write multiple register) Functions

For new application(s) or automat communication is it recommended to use the function	
<u>MSXE17xx MFIncCounterDisableAndReleaseLatch</u>	141
<u>Query frame layout</u>	142
<u>Response frame layout</u>	142
<u>Exception frame layout</u>	143
Function <u>MSXE170x MFIncCounterWrite32BitValue</u>	143
<u>Description</u>	143
For new application(s) or automat communication is it recommended to use the function	
<u>MSXE17xx MFIncCounterWrite32BitValue</u>	143
<u>Query frame layout</u>	144
<u>Response frame layout</u>	144
<u>Exception frame layout</u>	145
Function <u>MSXE170x MFIncCounterSetFIFO0Level</u>	145
<u>Description</u>	145
For new application(s) or automat communication is it recommended to use the function	
<u>MSXE17xx MFCommonSetFIFO0Level</u>	146
<u>Query frame layout</u>	146
<u>Response frame layout</u>	147
<u>Exception frame layout</u>	147
Function <u>MSXE170x DigitalIOWriteAllChannelsValue</u>	148
<u>Description</u>	148
For new application(s) or automat communication is it recommended to use the function	
<u>MSXE17xx DigitalIOWriteAllChannelsValue</u>	148
<u>Query frame layout</u>	148
<u>Response frame layout</u>	149
<u>Exception frame layout</u>	150

FC23 (read/write registers) Functions.....151

<u>Query frame layout</u>	151
<u>Response frame layout</u>	152
<u>Exception frame layout</u>	152

Exception code description.....153

Siemens Step 7 compatibility information (AWL/SDF code).....154

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.

• <u>GetLastCommandStatus</u>	Register: 0
• <u>GetLastCommandStatusEx</u>	Register: 10000
• <u>MXCommon_GetModuleType</u>	Register: 1
• <u>MXCommon_GetModuleTypeEx</u>	Register: 10200
• <u>MXCommon_GetTime</u>	Register: 2
• <u>MXCommon_GetTimeEx</u>	Register: 10500
• <u>MXCommon_TestCustomerID</u>	Register: 3
• <u>MXCommon_TestCustomerIDEx</u>	Register: 10550
• <u>MSXE17xx_MFIncCounterRead32BitsValueAll</u>	Register: 4000
• <u>MSXE17xx_MFIncCounterRead32BitsValue0</u>	Register: 4050
• <u>MSXE17xx_MFIncCounterRead32BitsValue1</u>	Register: 4100
• <u>MSXE17xx_MFIncCounterRead32BitsValue2</u>	Register: 4150
• <u>MSXE17xx_MFIncCounterRead32BitsValue3</u>	Register: 4200
• <u>MSXE17xx_DigitalIOReadAllChannelsValue</u>	Register: 7000
• <u>MSXE17xx_DigitalIOTestShortCircuit</u>	Register: 7050
• <u>MSXE17xx_IOWatchdogGetStatusAndValue</u>	Register: 8000
• <u>MSXE170x_DigitalIOReadAllChannelsValue</u>	Register: 100

Function GetLastCommandStatus

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

Description

Return the result of the last remote function call

Parameters:

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

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

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

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

Query frame layout

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

Response frame layout

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

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

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	112	0x7000	0x0070
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	108	0x6C00	0x006C
ReturnValue	4	32-bit integer	See the description above	0x????????	0x????????
Syserrno	4	32-bit integer	See the description above	0x????????	0x????????
Errstr	100	8-bit integer array	See the description above	0x??[100]	0x??[100]

Exception frame layout

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

Function GetLastCommandStatusEx

Description

Return the result of the last remote function call

Parameters:

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

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

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

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

Query frame layout

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

Response frame layout

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

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

Response frame layout

Function MXCommon__GetModuleType

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

Description

Returns the type of the MSX-E Module

Parameters:

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

Query frame layout

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

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
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	204	0xCC00	0x00CC
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	200	0xC800	0x00C8
str	200	8-bit integer array	See the description above	0x??[200]	0x??[200]

Exception frame layout

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

Function MXCommon__GetModuleTypeEx

Description

Returns the type of the MSX-E Module

Parameters:

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

Response frame layout

Query frame layout

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

Response frame layout

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

MODBUS interface description

		integer array	description above		
--	--	------------------	----------------------	--	--

Exception frame layout

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

Function MXCommon__GetTime

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

Description

Get the time on the module

Parameters:

[Response frame layout] **tv_sec**: Number of seconds since the Epoch

[Response frame layout] **tv_usec**: Number of microseconds since the begin of the second

Query frame layout

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

Response frame layout

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	2	0x0200	0x0002
word count	2	16-bit integer	4	0x0400	0x0004

Response frame layout

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

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MXCommon__GetTimeEx

Description

Get the time on the module

Parameters:

[Response frame layout] **tv_sec**: Number of seconds since the Epoch

[Response frame layout] **tv_usec**: Number of microseconds since the begin of the second

Query frame layout

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

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 MXCommon__TestCustomerID

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

Description

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

Parameters:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number	2	16-bit integer	3	0x0300	0x0003

Exception frame layout

MODBUS interface description

(=register)					
word count	2	16-bit integer	16	0x1000	0x0010

Response frame layout

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

Exception frame layout

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

Description

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

Parameters:

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

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

Query frame layout

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

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	35	0x2300	0x0023
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	32	0x20	0x20
bValueArray	16	8-bit integer array	See the description above	0x??[16]	0x??[16]
bCryptedValueArray	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

Exception frame layout

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

Function MSXE17xx__MFIncCounterRead32BitsValueAll

Description

Read measured value on all modules.

Parameters:

[Response frame layout] **ulValues** : Array that contain the measured values.

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

[Response frame layout] **ulTimeStampLows** : 32 bit low part of time stamp (us) array

- ulTimeStampLows [0] : Module 0 time stamp (us)
- ulTimeStampLows [1] : Module 1 time stamp (us)
- ulTimeStampLows [2] : Module 2 time stamp (us)
- ulTimeStampLows [3] : Module 3 time stamp (us)

[Response frame layout] **ulTimeStampHighs** : 32 bit high part of time stamp (s) array

- ◆ ulTimeStampHighs [0] : Module 0 time stamp (s)
- ◆ ulTimeStampHighs [1] : Module 1 time stamp (s)
- ◆ ulTimeStampHighs [2] : Module 2 time stamp (s)
- ◆ ulTimeStampHighs [3] : Module 3 time stamp (s)

Returns:

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

- 0 : No error.
- -1 : means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter.
- -4: Incremental counter not initialised
- -100 : Kernel function error (see syserrno).

Query frame layout

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

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	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	4000	0xA00F	0x0FA0
word count	2	16-bit integer	24	0x1800	0x0018

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big (Mo
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	51	0x3300	0x0033
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	48	0x30	0x30
ulValues	16	32-bit integer array	See the description above	0x????????[4]	0x???
ulTimeStampLows	16	32-bit integer array	See the description above	0x????????[4]	0x???
ulTimeStampHighs	16	32-bit integer array	See the description above	0x????????[4]	0x???

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

Function MSXE17xx__MFIncCounterRead32Bits

Description

Read measured value on the module 0.

Parameters:

[Response frame layout] **ulValue** : Measured value.

[Response frame layout] **ulTimeStampLow** : 32 bit low part of time stamp (us)

[Response frame layout] **ulTimeStampHigh** : 32 bit high part of time stamp (s)

Returns:

- ◇ Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
 - 0 : No error.
 - -1 : means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter.
 - -4: Incremental counter not initialised
 - -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	4050	0xD20F	0x0FD2
word count	2	16-bit integer	6	0x0600	0x0006

Response frame layout

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

MODBUS interface description

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

Exception frame layout

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

Function

MSXE17xx__MFIncCounterRead32Bits

Description

Read measured value on the module 1.

Parameters:

[Response frame layout] **ulValue** : Measured value.

[Response frame layout] **ulTimeStampLow** : 32 bit low part of time stamp (us)

[Response frame layout] **ulTimeStampHigh** : 32 bit high part of time stamp (s)

Returns:

MODBUS interface description

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

- 0 : No error.
- -1 : means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter.
- -4: Incremental counter not initialised
- -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	4100	0x0410	0x1004
word count	2	16-bit integer	6	0x0600	0x0006

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2		15	0x0F00	0x000F

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	12	0x0C	0x0C
ulValue	4	32-bit integer	See the description above	0x???????	0x???????
ulTimeStampLow	4	32-bit integer	See the description above	0x???????	0x???????
ulTimeStampHigh	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MSXE17xx_MFIncCounterRead32Bits

Description

Read measured value on the module 2.

Parameters:

MODBUS interface description

[Response frame layout] **ulValue** : Measured value.

[Response frame layout] **ulTimeStampLow** : 32 bit low part of time stamp (us)

[Response frame layout] **ulTimeStampHigh** : 32 bit high part of time stamp (s)

Returns:

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

- 0 : No error.
- -1 : means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter.
- -4: Incremental counter not initialised
- -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	4150	0x3610	0x1036
word count	2	16-bit integer	6	0x0600	0x0006

Response frame layout

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

MODBUS interface description

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

Function

MSXE17xx__MFIIncCounterRead32Bits

Description

Read measured value on the module 3.

Parameters:

[Response frame layout] **ulValue** : Measured value.

[Response frame layout] **ulTimeStampLow** : 32 bit low part of time stamp (us)

[Response frame layout] **ulTimeStampHigh** : 32 bit high part of time stamp (s)

Returns:

- ◇ Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
- 0 : No error.
 - -1 : means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter.
 - -4: Incremental counter not initialised
 - -100 : Kernel function error (see syserrno).

Query frame layout

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

MODBUS interface description

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	4200	0x6810	0x1068
word count	2	16-bit integer	6	0x0600	0x0006

Response frame layout

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

Exception frame layout

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

MODBUS interface description

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

Function

MSXE17xx__DigitalIOWReadAllChannels

Description

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

Parameters:

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

Returns:

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

- 0 : No error.
- -1 : Means an system error occurred
- -100 : Kernel function error (see syserrno).

Query frame layout

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

MODBUS interface description

length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	7000	0x581B	0x1B58
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit 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

MSXE17xx__DigitalIOTestShortCircuit

Description

Parameters:

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

- 0 : no short circuit
- 1 : short circuit

Returns:

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

- 0 : No error.
- -1 : Means an system error occurred
- -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	7050	0x8A1B	0x1B8A
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

MODBUS interface description

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

Function MSXE17xx__IOWatchdogGetStatusAn

Description

Get watchdog current status and value information

Parameters:

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

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

Returns:

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

- 0 : No error.
- -1 : Means an system error occurred
- -100 : Kernel function error (see syserrno).

Query frame layout

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

MODBUS interface description

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

Response frame layout

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

Exception frame layout

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

MODBUS interface description

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

Function

MSXE170x__DigitalIOReadAllChannels

Description

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

For new application(s) or automat communication is it recommended to use the function

MSXE17xx__DigitalIOReadAllChannelsValue.

Parameters:

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

Query frame layout

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

MODBUS interface description

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	100	0x6400	0x0064
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	8	0x0800	0x0008
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	4	0x0400	0x0004
ChannelsValue	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 interface description

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

FC16 (write multiple register) Functions

[Top](#)

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

- ◇ [MXCommon_SetHardwareTriggerFilterTime](#)
- ◇ [MXCommon_SetHardwareTriggerFilterTimeEx](#)
- ◇ [MXCommon_InitAndStartSynchroTimer](#)
- ◇ [MXCommon_InitAndStartSynchroTimerEx](#)
- ◇ [MXCommon_StopAndReleaseSynchroTimer](#)
- ◇ [MXCommon_StopAndReleaseSynchroTimerEx](#)
- ◇ [MXCommon_Reboot](#)
- ◇ [MXCommon_RebootEx](#)
- ◇ [MXCommon_SetCustomerKey](#)
- ◇ [MXCommon_SetCustomerKeyEx](#)
- ◇ [MXCommon_SetFilterChannels](#)
- ◇ [MXCommon_SetFilterChannelsEx](#)
- ◇ [MSXE17xx_MFCommonSetInputsFilter](#)
- ◇ [MSXE17xx_MFCommonReferenceVoltageActivation](#)
- ◇ [MSXE17xx_MFCommonSetFIFO0Level](#)
- ◇ [MSXE17xx_MFCommonEnableDisableTriggerGate](#)
- ◇ [MSXE17xx_MFIncCounterInit](#)
- ◇ [MSXE17xx_MFIncCounterRelease](#)
- ◇ [MSXE17xx_MFIncCounterClear](#)

MODBUS interface description

- ◇ MSXE17xx MFIncCounterWrite32BitValue
- ◇ MSXE17xx MFIncCounterInitAndEnableCompareLogic
- ◇ MSXE17xx MFIncCounterDisableAndReleaseCompareLogic
- ◇ MSXE17xx MFIncCounterInitHardwareTrigger
- ◇ MSXE17xx MFIncCounterReleaseHardwareTrigger
- ◇ MSXE17xx MFIncCounterInitIndex
- ◇ MSXE17xx MFIncCounterReleaseIndex
- ◇ MSXE17xx MFIncCounterInitAndEnableLatch
- ◇ MSXE17xx MFIncCounterDisableAndReleaseLatch
- ◇ MSXE17xx MFIncCounterInitAndEnableClear
- ◇ MSXE17xx MFIncCounterDisableAndReleaseClear
- ◇ MSXE17xx MFIncCounterInitAndEnableFrequencyMeasure
- ◇ MSXE17xx MFIncCounterDisableAndReleaseFrequencyMeasure
- ◇ MSXE17xx DigitalIOWriteAllChannelsValue
- ◇ MSXE17xx DigitalIORearmShortCircuit
- ◇ MSXE17xx DigitalIOInitPort
- ◇ MSXE17xx IOWatchdogInitAndStart
- ◇ MSXE17xx IOWatchdogStopAndRelease
- ◇ MSXE170x MFIncCounterInit
- ◇ MSXE170x MFIncCounterRelease
- ◇ MSXE170x MFIncCounterClear
- ◇ MSXE170x MFIncCounterInitAndEnableLatchRegister
- ◇ MSXE170x MFIncCounterDisableAndReleaseLatchRegister

◇ MSXE170x MFIncCounterWrite32BitValue

◇ MSXE170x MFIncCounterSetFIFO0Level

◇ MSXE170x DigitalIOWriteAllChannelsValue

Function

MXCommon__SetHardwareTriggerFilterTime

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

MXCommon__SetHardwareTriggerFilterTime

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)
-------	--------------	------	-------	-----------------------	-----------------------

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	16	0x1000	0x0010
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	100	0x6400	0x0064
word count	2	16-bit integer	4	0x0400	0x0004
byte count	2	16-bit integer	8	0x0800	0x0008
ulFilterTime	4	32-bit integer	See the description above	0x????????	0x????????
Reserved	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

MODBUS interface description

Reference number (=register)		16-bit integer			
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__SetHardwareTriggerFilter

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

MODBUS interface description

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

Returns

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

◇ **0** The remote function performed OK

◇ **-1** Internal system error occurred. See value of syserrno

Query frame layout

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

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

MXCommon__InitAndStartSynchroTimerEx.

Description

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

Parameters:

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

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

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

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

[Query frame layout] **ulGenerateTriggerMode:**

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

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

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

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

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

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

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

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

Returns:

Possible return value on the remote system (read

MODBUS interface description

them with GetLastCommandStatus)

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

Query frame layout

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

MODBUS interface description

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

Response frame layout

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

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

MXCommon__InitAndStartSynchroTim

Description

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

Parameters:

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

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

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

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

[Query frame layout] **ulGenerateTriggerMode:**

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

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

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

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

MODBUS interface description

- 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 occurred
- -2: not available time base
- -3: timer reload value can not be greater than 65535
- -4: minimum time reload is 5 us
- -5: Number of cycle can not be greater than 65535
- -6: Generate trigger mode error
- -100: Init timer error
- -101: Start timer error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	39	0x2700	0x0027
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11050	0x2A2B	0x2B2A
word count	2		16	0x1000	0x0010

MODBUS interface description

		16-bit integer			
byte count	1	8-bit integer	32	0x20	0x2
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
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	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__StopAndReleaseSynchr

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

MXCommon__StopAndReleaseSynchroTimerE

Description

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

Parameters:

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

Returns:

MODBUS interface description

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 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	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__StopAndReleaseSynchr

Description

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

Parameters:

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

Returns:

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
- -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 - 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	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:

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 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	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 -	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	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

MODBUS interface description

Returns:

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

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

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	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]

[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			0x??[32]	0x??[32]

MODBUS interface description

		8-bit integer array	See the description above		
bPublicKey	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

Response frame layout

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

Exception frame layout

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

MODBUS interface description

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

Function MXCommon__SetCustomerKeyEx

Description

Permit to set the Customer key

Parameters:

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

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

Returns:

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

- 0 : means the remote function performed OK
- -1: means an system error 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
	1		0x10	0x10	0x10

MODBUS interface description

MODBUS Function code		8-bit integer			
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	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

MXCommon__SetFilterChannels

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

MXCommon__SetFilterChannelsEx.

Description

Permit to set a filter per channel

Parameters:

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

Returns:

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

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

Query frame layout

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

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	105	0x6900	0x0069

MODBUS interface description

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

Exception frame layout

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

Function

MXCommon__SetFilterChannelsEx

Description

Permit to set a filter per channel

Parameters:

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

Returns:

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

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

Query frame layout

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

Response frame layout

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

MODBUS interface description

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

Exception frame layout

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

Function

MSXE17xx__MFCommonSetInputsFilter

Description

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

Parameters:

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

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

- 0: Filter nicht benutzt
- 1: 100 ns
- 2: 200 ns
- 3: 300 ns ...

MODBUS interface description

- 262143 : 26,2143 ms

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

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

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

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

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

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

Returns:

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

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

Query frame layout

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

MODBUS interface description

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	6000	0x7017	0x1770
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x??????
ulInputAFilterValue	4	32-bit integer	See the description above	0x????????	0x??????
ulInputBFilterValue	4	32-bit integer	See the description above	0x????????	0x??????
ulInputCFilterValue	4	32-bit integer	See the description above	0x????????	0x??????
ulInputDFilterValue	4	32-bit integer	See the description above	0x????????	0x??????

Response frame layout

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

MODBUS interface description

Reference number (=register)	2	16-bit integer	6000	0x7017	0x1770
word count	2	16-bit integer	10	0x0A00	0x000A

Exception frame layout

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

Function

MSXE17xx__MFCommonReferenceVo

Description

Permit to activate the reference voltage to pin D-

Parameters:

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

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

- 0: normal mode from D- (Default mode)
- 1: activate the reference voltage to pin D-

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

MODBUS interface description

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

Returns:

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

- 0 : No error.
- -1 : Means an system error occurred
- -2 : Multifunction sub module index selection error
- -3 : Activation flag selection error
- -100 : Kernel function error (see syserrno).

Query frame layout

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

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 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	6050	0xA217	0x17A2
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function

MSXE17xx__MFCommonSetFIFO0Lev

Description

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

Parameters:

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

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

[Query frame layout] **ulTimeOutTimeBase** :
Define a Time out : permit to receive the data from the FIFO before the FIFO level is reached.
Time base of the timer (0: disabled, 1 for us, 2 for ms, 3 for s)

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

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

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

Returns:

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

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

Query frame layout

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

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	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6100	0xD417	0x17D4
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x????
ulFIFOLevel	4	32-bit integer	See the description above	0x???????	0x????
ulTimeOutTimeBase	4	32-bit integer	See the description above	0x???????	0x????
ulReloadValue	4	32-bit integer	See the description above	0x???????	0x????
ulOption01	4	32-bit integer	See the description above	0x???????	0x????
ulOption02	4	32-bit integer	See the description above	0x???????	0x????

Response frame layout

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

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	6100	0xD417	0x17D4
word count	2	16-bit integer	12	0x0C00	0x000C

Exception frame layout

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

Function

MSXE17xx__MFCommonEnableDisab

Description

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

Parameters:

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

MODBUS interface description

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

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

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

Returns:

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

- 0 : No error.
- -1 : Means an system error occured
- -2 : ulTriggerConfiguration parameter is wrong
- -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big (Motorola)
transaction identifier	2	16-bit 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	6150	0x0618	0x1806
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulTriggerConfiguration	4	32-bit integer	See the description above	0x????????	0x????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????
ulOption02	4			0x????????	0x????

MODBUS interface description

		32-bit integer	See the description above	
--	--	----------------	---------------------------	--

Response frame layout

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

code		integer	corresponding chapter		
------	--	---------	--------------------------	--	--

Function

MSXE17xx__MFIncCounterInit

Description

Initialise the counter

[Query frame layout]

- ◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout]

- ◇ **ulCounterMode** : Set the counter mode : either
 - MSXE170X_COUNTER_QUADRUPLE_MODE (0x4)
 - MSXE170X_COUNTER_DOUBLE_MODE (0x2)
 - MSXE170X_COUNTER_SIMPLE_MODE (0x1)
 - MSXE170X_COUNTER_DIRECT_MODE (0x0)

[Query frame layout]

- ◇ **ulCounterOption** : Set the counter option if in QUADRUPLE/DOUBLE/SIMPLE mode : either
 - MSXE170X_COUNTER_HYSTERESIS_ON (0x1)
 - MSXE170X_COUNTER_HYSTERESIS_OFF (0x0)
 if in DIRECT mode :
 - MSXE170X_COUNTER_INCREMENT (0x0)
 - MSXE170X_COUNTER_DECREMENT (0x1)

[Query frame layout]

- ◇ **ulOption01** : Set it to 0 [Query frame layout]
- ◇ **ulOption02** : Set it to 0 [Query frame layout]
- ◇ **ulOption03** : Set it to 0 [Query frame layout]
- ◇ **ulOption04** : Set it to 0

[Response frame layout]

- ◆ **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
 - ◇ 0: means the remote function performed OK
 - ◇ -1: means an system error occurred
 - ◇ -2: Multifunction sub module index selection error
 - ◇ -3: Counter mode selection error
 - ◇ -4: Counter option selection error

MODBUS interface description

◇ -5: Multifunction sub module is not a incremental counter

◇ -100: Init counter kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	35	0x2300	0x0023
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4300	0xCC10	0x10CC
word count	2	16-bit integer	14	0x0E00	0x000E
byte count	1	8-bit integer	28	0x1C	0x1C
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulCounterMode	4	32-bit integer	See the description above	0x???????	0x???????
ulCounterOption	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????
ulOption02	4	32-bit integer	See the description above	0x???????	0x???????
ulOption03	4	32-bit integer	See the description above	0x???????	0x???????
ulOption04	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4300	0xCC10	0x10CC
word count	2	16-bit integer	14	0x0E00	0x000E

Exception frame layout

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

Function

MSXE17xx__MFIncCounterRelease

Description

Release the counter

[Query frame layout]

◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Response frame layout]

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

◇ 0: means the remote function performed OK

◇ -1: means an system error occurred

◇ -2: Multifunction sub module index selection error

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	4350	0xFE10	0x10FE
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulMFModuleIndex	4	32-bit integer	See the description above	0x??????	0x??????

Response frame layout

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

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	4350	0xFE10	0x10FE
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function MSXE17xx__MFIncCounterClear

Description

Clear the 32 bits counter

[Query frame layout]

- ◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Response frame layout]

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

- ◇ 0: means the remote function performed OK
- ◇ -1: means an system error occurred
- ◇ -2: Multifunction sub module index selection error
- ◇ -3: Multifunction sub module is not a incremental counter
- ◇ -4: Incremental counter not initialised
- ◇ -100: Kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	4400	0x3011	0x1130
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x??????

Response frame layout

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

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	4400	0x3011	0x1130
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function

MSXE17xx__MFIncCounterWrite32BitV

Description

write a 32 bits counter value

[Query frame layout]

◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout]

◇ **ulCounterValue** : Counter value

[Response frame layout]

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

◇ 0: means the remote function performed OK

◇ -1: means an system error occurred

◇ -2: Multifunction sub module index selection error

◇ -3: Counter value error

◇ -4: Multifunction sub module is not a incremental counter

◇ -5: Incremental counter not initialised

◇ -100: Kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4450	0x6211	0x1162
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulCounterValue	4	32-bit integer	See the description	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	4450	0x6211	0x1162
word count	2	16-bit integer	4	0x0400	0x0004

Exception frame layout

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

Function

MSXE17xx__MFIncCounterInitAndEna

Description

Init and enable a counter compare value

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

- ◇ DWORD 0 : Time stamp micro s
- ◇ DWORD 1 : Time stamp s
- ◇ DWORD 2 :
 - D1-D0 : Sub module index (0 to 3)
 - D31-D16 : Sub module functionality (0)
 - or value of the digital inputs when ulLatchDigValue is enabled
- ◇ DWORD 3 : Event mask
 - D31-D0 :
 - 0: Compare
 - 1: Frequency measurement
 - 2: Hardware trigger latch occur
 - 3: Synchro input latch occur
 - 4: Index input latch occur
- ◇ DWORD 4 :
 - D31-D0: Counter value (DWORD)

Parameters:

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

[Query frame layout] **ulValue / ulValueLow** : compare value (double). ulValueLow

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

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

[Query frame layout] **ulSynchroTrigger**

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

[Query frame layout] **ulLatchDigValue** : Parallel latch of the digital inputs

- 0: The value of the digital inputs are not

MODBUS interface description

latched.

- 1: The Value of the digital inputs are latched and stored in the upper 16 Bits of the data package for source counter.

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

Returns:

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

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Compare value error
- -4: Compare mode error
- -5: Synchro trigger error
- -6: Multifunction sub module is not a incremental counter
- -7: Incremental counter not initialised
- -8: Compare logic already initialised
- -100: Init and enable counter compare kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endi (Motoro
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4500	0x9411	0x1194
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulMFModuleIndex	4	32-bit integer	See the description	0x???????	0x???????

MODBUS interface description

			above		
ulValue	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??????
ulLatchDigValue	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	4500	0x9411	0x1194
word count	2	16-bit integer	12	0x0C00	0x000C

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

MSXE17xx__MFIncCounterDisableAnd

Description

Disable and Release a counter compare value

Parameters:

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

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (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	4550	0xC611	0x11C6
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4550	0xC611	0x11C6
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function

MSXE17xx__MFIncCounterInitHardwa

Description

Init the hardware trigger configuration

Parameters:

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

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

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

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

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

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

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

MODBUS interface description

Returns:

- ◇ Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter.
 - -4: Trigger edge selection error
 - -5: Trigger count selection error
 - -6: Incremental counter not initialised
 - -7: Hardware trigger already initialised
 - -100: Init hardware trigger kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	27	0x1B00	0x001B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4600	0xF811	0x11F8
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulEdgeSelection	4	32-bit integer	See the description above	0x???????	0x???????
ulCount	4	32-bit integer	See the description above	0x???????	0x???????

MODBUS interface description

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

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	4600	0xF811	0x11F8
word count	2	16-bit integer	10	0x0A00	0x000A

Exception frame layout

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

MODBUS interface description

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

Function

MSXE17xx__MFIncCounterReleaseHa

Description

Release the hardware trigger

Parameters:

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

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

Returns:

- ◇ Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter.
 - -4: Incremental counter not initialised
 - -5: Hardware trigger not initialised
 - -6: Hardware trigger used and can not released
 - -100: Release hardware trigger kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endi (Motoro
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	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	4650	0x2A12	0x122A
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4650	0x2A12	0x122A
word count	2	16-bit integer	4	0x0400	0x0004

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

Function MSXE17xx__MFIncCounterInitIndex

Description

Init the index configuration

Parameters:

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

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

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

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

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

Returns:

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

MODBUS interface description

- -3: Multifunction sub module is not a incremental counter.
- -4: Index edge selection error
- -5: Incremental counter not initialised
- -6: Index already initialised

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	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	4700	0x5C12	0x125C
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x???????
ulEdgeSelection	4	32-bit integer	See the description above	0x????????	0x???????
ulOption01	4	32-bit integer	See the description above	0x????????	0x???????
ulOption02	4	32-bit integer	See the description above	0x????????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied	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	4700	0x5C12	0x125C
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function

MSXE17xx__MFIIncCounterReleaseInd

Description

Release the index

Parameters:

MODBUS interface description

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

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

Returns:

- ◇ Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter.
 - -4: Incremental counter not initialised
 - -5: Index not initialised
 - -6: Index used and can not released
 - -100: Release Index kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4750	0x8E12	0x128E
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4750	0x8E12	0x128E
word count	2	16-bit integer	4	0x0400	0x0004

Exception frame layout

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

Function

MSXE17xx__MFIncCounterInitAndEna

Description

Init and enable a counter latch logic

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

- ◇ DWORD 0 : Time stamp micro s
- ◇ DWORD 1 : Time stamp s
- ◇ DWORD 2 :
 - D1-D0 : Sub module index (0 to 3)
 - D31-D16 : Sub module functionality (0)
 - or value of the digital inputs when ulLatchDigValue is enabled
- ◇ DWORD 3 : Event mask
 - D31-D0 :
 - 0: Compare
 - 1: Frequency measurement
 - 2: Hardware trigger latch occur
 - 3: Synchro input latch occur
 - 4: Index input latch occur
- ◇ DWORD 4 :
 - D31-D0: Counter value (DWORD)

Parameters:

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

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

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

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

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

MODBUS interface description

latch source)

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

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

[Query frame layout] **ulLatchDigValue** : Parallel latch of the digital inputs

- 0: The value of the digital inputs are not latched.
- 1: The Value of the digital inputs are latched and stored in the upper 16 Bits of the data package for source counter.

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

Returns:

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

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter.
- -4: Incremental counter not initialised
- -5: Latch logic already initialised
- -6: Latch source selection error
- -7: Previously condition selection error
- -8: Auto mode selection error
- -9: Hardware trigger not initialised. Refer to MSXE17xx__MFIncCounterInitHardwareTrigger
- -10: Index input not initialised. Refer to MSXE17xx__MFIncCounterInitIndex
- -100: Init and enable counter latch kernel function error

Query frame layout

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

MODBUS interface description

length	2	16-bit integer	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4800	0xC012	0x12C0
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulLatchSource	4	32-bit integer	See the description above	0x???????	0x???????
ulCondition	4	32-bit integer	See the description above	0x???????	0x???????
ulAutoMode	4	32-bit integer	See the description above	0x???????	0x???????
ulLatchDigValue	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 interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4800	0xC012	0x12C0
word count	2	16-bit integer	12	0x0C00	0x000C

Exception frame layout

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

Function

MSXE17xx__MFIncCounterDisableAnd

Description

Disable and Release a counter latch logic

Parameters:

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

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

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

MODBUS interface description

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

Returns:

- ◇ Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter.
 - -4: Incremental counter not initialised
 - -5: Latch logic not initialised
 - -6: Latch source selection error
 - -100: Disable and release counter latch register kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4850	0xF212	0x12F2
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulLatchSource	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description	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	4850	0xF212	0x12F2
word count	2	16-bit integer	6	0x0600	0x0006

Exception frame layout

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

Function

MSXE17xx__MFIncCounterInitAndEna

Description

Init and enable a counter clear logic

Parameters:

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

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

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

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

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

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

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

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

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

Returns:

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

MODBUS interface description

- -3: Multifunction sub module is not a incremental counter.
- -4: Incremental counter not initialised
- -5: Clear logic already initialised
- -6: Clear source selection error
- -7: Previously condition selection error
- -8: Auto mode selection error
- -9: Hardware trigger not initialised. Refer to MSXE17xx__MFIncCounterInitHardwareTrigger
- -10: Index input not initialised. Refer to MSXE17xx__MFIncCounterInitIndex
- -100: Init and enable counter clear kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4900	0x2413	0x1324
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulClearSource	4	32-bit integer	See the description above	0x???????	0x???????
ulCondition	4	32-bit integer	See the description above	0x???????	0x???????
ulAutoMode	4	32-bit integer	See the description above	0x???????	0x???????

MODBUS interface description

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

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	4900	0x2413	0x1324
word count	2	16-bit integer	12	0x0C00	0x000C

Exception frame layout

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

MODBUS interface description

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

Function

MSXE17xx__MFIncCounterDisableAnd

Description

Disable and Release a counter clear logic

Parameters:

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

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

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

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

Returns:

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

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

Query frame layout

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

MODBUS interface description

			server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4950	0x5613	0x1356
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x??????
ulClearSource	4	32-bit integer	See the description above	0x????????	0x??????
ulOption01	4	32-bit integer	See the description above	0x????????	0x??????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number	2	16-bit integer	4950	0x5613	0x1356

MODBUS interface description

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

Exception frame layout

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

Function

MSXE17xx_MFIncCounterInitAndEna

Description

Init and enable frequency measurement

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

- ◇ DWORD 0 : Time stamp micro s
- ◇ DWORD 1 : Time stamp s
- ◇ DWORD 2 : Counter index
 - D1-D0 : Sub module index (0 to 3)
 - D31-D16 : Sub module functionality (0)
 - or value of the digital inputs when ulLatchDigValue is enabled
- ◇ DWORD 3 : Event mask
 - D31-D0 :
 - 0: Compare
 - 1: Frequency measurement
 - 2: Hardware trigger latch occur
 - 3: Synchro input latch occur

MODBUS interface description

4: Index input latch occur

◇ DWORD 4 :

· D31-D0: Counter value (DWORD)

Parameters:

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

[Query frame layout] ***ulTimingInterval*** : Timing interval from 1 to 0xFFFF, one step correspond to 100 ns.

- 1: Time interval = 100 ns
- 2: Time interval = 200 ns
- ...

[Query frame layout] ***ulOption01*** : set the measure mode

- 0: One shot, the frequency measure is done at each call of this function
- 1: Triggered mode, this function has to be called only once (to initialize the measure). The synchro trigger is used to trigger the measure. The synchro trigger cycle time should be > frequency time interval

[Query frame layout] ***ulOption02*** : set the measure time base

- 0: 100 ns
- 1: 100 us

Returns:

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

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Timing interval selection error
- -4: Multifunction sub module is not an incremental counter
- -5: Incremental counter not initialised
- -6: Frequency measurement already initialised and started
- -7: Frequency measurement wrong mode paramter
- -8: Frequency measurement wrong time base parameter
- -100: Init and enable frequency measurement kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	5000	0x8813	0x1388
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x???????
ulTimingInterval	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

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	5000	0x8813	0x1388
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function

MSXE17xx__MFIncCounterDisableAnd

Description

Disable and release frequency measurement

Parameters:

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

MODBUS interface description

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

Returns:

- ◇ Possible return value on the remote system (read them with **GetLastCommandStatusEx**):
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not an incremental counter
 - -4: Incremental counter not initialised
 - -5: Frequency measurement not initialised
 - -100: Init and enable frequency measurement kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	5050	0xBA13	0x13BA
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	5050	0xBA13	0x13BA
word count	2	16-bit integer	4	0x0400	0x0004

Exception frame layout

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

Function

MSXE17xx__DigitalIOWriteAllChannels

Description

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

Parameters:

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

Returns:

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

- 0: means the remote function performed OK
- -1: means an system error occurred
- -100: Write digital I/O kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - 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	7100	0xBC1B	0x1BBC
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

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

Exception frame layout

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

Function

MSXE17xx__DigitalIORearmShortCircuit

Description

Rearm digital outputs after a short circuit happened.

Parameters:

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

Returns:

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

- 0 : No error.
- -1 : means an system error occurred
- -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - 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	7150	0xEE1B	0x1BEE
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Exception frame layout

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

Function

MSXE17xx__DigitalIOInitPort

Description

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

Parameters:

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

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

- 0 : input
- 1 : output

Returns:

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

- 0 : No error.
- -1 : means an system error occurred
- -2: Digital i/o port selection error
- -3: Port configuration selection error
- -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big end (Motor)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7200	0x201C	0x1C20
word count	2	16-bit integer	4	0x0400	0x0004

MODBUS interface description

byte count	1	8-bit integer	8	0x08	0x08
ulPort	4	32-bit integer	See the description above	0x????????	0x??????
ulPortConfiguration	4	32-bit integer	See the description above	0x????????	0x??????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7200	0x201C	0x1C20
word count	2	16-bit integer	4	0x0400	0x0004

Exception frame layout

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

MODBUS interface description

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

Function

MSXE17xx__IOWatchdogInitAndStart

Description

Init and start the digital output IO watchdog.

Parameters:

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

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

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

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

Returns:

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

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

Query frame layout

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

MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	8050	0x721F	0x1F72
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulTimeValue	4	32-bit integer	See the description above	0x????????	0x????????
ulOption1	4	32-bit integer	See the description above	0x????????	0x????????
ulOption2	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit 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	8050	0x721F	0x1F72
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function

MSXE17xx__IOWatchdogStopAndRelease

Description

Stop and release the digital output watchdog.

Parameters:

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

Returns:

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

- 0 : No error.
- -1 : Means an system error occurred
- -100 : Kernel function error (see syserrno).

Query frame layout

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

Response frame layout

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

MODBUS interface description

Function code		integer			
Reference number (=register)	2	16-bit integer	8100	0xA41F	0x1FA4
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function

MSXE170x__MFIncCounterInit

Description

Initialise the counter

For new application(s) or automat communication is it recommended to use the function MSXE17xx__MFIncCounterInit.

[Query frame layout]

◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout]

◇ **ulCounterMode** : Set the counter mode : either

MODBUS interface description

- MSXE170X_COUNTER_QUADRUPLE_MODE
(0x4)
- MSXE170X_COUNTER_DOUBLE_MODE
(0x2)
- MSXE170X_COUNTER_SIMPLE_MODE
(0x1)
- MSXE170X_COUNTER_DIRECT_MODE
(0x0)

[Query frame layout]

- ◇ **ulCounterOption** : Set the counter option
if in QUADRUPLE/DOUBLE/SIMPLE mode :
either

- MSXE170X_COUNTER_HYSTERESIS_ON
(0x1)
- MSXE170X_COUNTER_HYSTERESIS_OFF
(0x0)

if in DIRECT mode :

- MSXE170X_COUNTER_INCREMENT
(0x0)
- MSXE170X_COUNTER_DECREMENT
(0x1)

[Query frame layout]

- ◇ **ulOption01** : Set it to 0 [Query frame layout]
- ◇ **ulOption02** : Set it to 0 [Query frame layout]
- ◇ **ulOption03** : Set it to 0 [Query frame layout]
- ◇ **ulOption04** : Set it to 0

[Response frame layout]

◆ Possible return value on the remote system:

- ◇ 0: means the remote function performed OK
- ◇ -1: means an system error occurred
- ◇ -2: Multifunction sub module index selection error
- ◇ -3: Counter mode selection error
- ◇ -4: Counter option selection error
- ◇ -5: Counter filter selection error
- ◇ -100: Init counter kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endi (Motoro
transaction identifier	2	16-bit 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		0 or 1		

For new application(s) or automatcommunication is it recommended tuuse the functionMSXE17x132MFIncC

MODBUS interface description

		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	1	0x0100	0x0001
word count	2	16-bit integer	14	0x0E00	0x000E
byte count	2	16-bit integer	28	0x1C00	0x001C
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulCounterMode	4	32-bit integer	See the description above	0x???????	0x???????
ulCounterOption	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????
ulOption02	4	32-bit integer	See the description above	0x???????	0x???????
ulOption03	4	32-bit integer	See the description above	0x???????	0x???????
ulOption04	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

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

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1	0x0100	0x0001
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

MSXE170x__MFIncCounterRelease

Description

Release the counter

For new application(s) or automat communication is it recommended to use the function MSXE17xx__MFIncCounterRelease.

[Query frame layout]

◊ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Response frame layout]

◆ **Possible return value on the remote system:**

MODBUS interface description

◇ 0: means the remote function performed OK

◇ -1: means an system error occurred

◇ -2: Multifunction sub module index selection 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	2	0x0200	0x0002
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x???????

Response frame layout

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

For new application(s) or automatcommunication is it recommended tuuse the functionMSXE17x135MFIncC

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2	0x0200	0x0002
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

MSXE170x__MFIncCounterClear

Description

Clear the 32 bits counter

For new application(s) or automat communication is it recommended to use the function MSXE17xx__MFIncCounterClear.

[Query frame layout]

◊ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Response frame layout]

◆ **Possible return value on the remote system:**

MODBUS interface description

- ◇ 0: means the remote function performed OK
- ◇ -1: means an system error occurred
- ◇ -2: Multifunction sub module index selection error
- ◇ -100: Write counter value kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	3	0x0300	0x0003
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

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

For new application(s) or automatic communication it is recommended to use the function MSXE17x137MFIncC

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	3	0x0300	0x0003
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

MSXE170x__MFINcCounterInitAndEna

Description

Init and enable a counter latch register

For new application(s) or automat communication is it recommended to use the function

MSXE17xx__MFINcCounterInitAndEnableLatch

[Query frame layout]

◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout]

MODBUS interface description

- ◇ **ulLatchRegister** : 0: First Latch Register used with trigger input, 1: Second Latch Register used with synchro input [Query frame layout]
- ◇ **ulHardwareTriggerEdgeSelection** : not used for the synchro input
 - 01 : rising front (Only if trigger input selected)
 - 10 : falling front (Only if trigger input selected)
 - 11 : Both front (Only if trigger input selected)
 [Query frame layout]
- ◇ **ulHardwareTriggerCount** : not used for the synchro input
 Define the number of trigger events before the action occur
 1 : all trigger event start the action
 max value : 65535 [Query frame layout]
- ◇ **ulOption01** : set it to 0

[Response frame layout]

◆ Possible return value on the remote system:

- ◇ 0: means the remote function performed OK
- ◇ -1: means an system error occurred
- ◇ -2: Multifunction sub module index selection error
- ◇ -3: Latch register selection error
- ◇ -4: Trigger edge selection error
- ◇ -5: Trigger count selection error
- ◇ -100: Init and enable counter latch register kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Int)
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	28	0x1C00
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	4	0x0400
word count	2		10	0x0A00

For new application(s) or automatic communication it is recommended to use the function `MSX017x139MFI` in C.

MODBUS interface description

		16-bit integer		
byte count	2	16-bit integer	20	0x140
ulMFModuleIndex	4	32-bit integer	See the description above	0x???
ulLatchRegister	4	32-bit integer	See the description above	0x???
ulHardwareTriggerEdgeSelection	4	32-bit integer	See the description above	0x???
ulHardwareTriggerCount	4	32-bit integer	See the description above	0x???
ulOption01	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 - 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	4	0x0400	0x0004
word count	2	16-bit integer	10	0x0A00	0x000A

Exception frame layout

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

Function

MSXE170x__MFIncCounterDisableAndRelease

Description

Disable and Release a counter latch register

For new application(s) or automat communication is it recommended to use the function

MSXE17xx__MFIncCounterDisableAndRelease

[Query frame layout]

◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout]

◇ **ulLatchRegister** : 0: First Lath Register used with 24 V input, 1: Second Lath Register used with synchro input [Query frame layout]

◇ **ulOption01** : set it to 0

[Response frame layout]

◆ **Possible return value on the remote system:**

◇ 0: means the remote function performed OK

◇ -1: means an system error occurred

◇ -2: Multifunction sub module index selection error

MODBUS interface description

- ◇ -3: Latch register selection error
- ◇ -100: Disable and release counter latchd register kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	20	0x1400	0x0014
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	5	0x0500	0x0005
word count	2	16-bit integer	6	0x0600	0x0006
byte count	2	16-bit integer	12	0x0C00	0x000C
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulLatchRegister	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

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

For new application(s) or automatcommunication is it recommended tuuse the functionMSXE17x142MFIncC

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	5	0x0500	0x0005
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

MSXE170x__MFIncCounterWrite32BitV

Description

write a 32 bits counter value

For new application(s) or automat communication is it recommended to use the function MSXE17xx__MFIncCounterWrite32BitValue.

MODBUS interface description

[Query frame layout]

◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout]

◇ **ulCounterValue** : Counter value

[Response frame layout]

◆ Possible return value on the remote system:

◇ 0: means the remote function performed OK

◇ -1: means an system error occurred

◇ -2: Multifunction sub module index selection error

◇ -3: Counter value error

◇ -100: Write counter value kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	16	0x1000	0x0010
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6	0x0600	0x0006
word count	2	16-bit integer	4	0x0400	0x0004
byte count	2	16-bit integer	8	0x0800	0x0008
ulMFModuleIndex	4	32-bit integer	See the description above	0x???????	0x???????
ulCounterValue	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	0x0000	0x0000

For new application(s) or automatcommunication is it recommended tuuse the functionMSXE17x144MFIncC

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	6	0x0600	0x0006
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

MSXE170x__MFIncCounterSetFIFO0L

Description

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

For new application(s) or automat communication is it recommended to use the function MSXE17xx__MFCommonSetFIFO0Level.

[Query frame layout]

◇ **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).

[Query frame layout]

◇ **ulFIFOLevel** : Define the FIFO level (1 to 200)

[Query frame layout]

◇ **ulOption01** : Define a Time out : permit to receive the data from the FIFO before the FIFO level is reached

D31-D16 : Time base: 1:us, 2:ms, 3:s

D15-D0 : Time value

if this value is 0, then no time out used

[Query frame layout]

◇ **ulOption02** : set it to 0

[Response frame layout]

◆ **Possible return value on the remote system:**

◇ 0: means the remote function performed OK

◇ -1: means an system error occurred

◇ -2: Multifunction sub module index selection error

◇ -3: FIFO level value is wrong

◇ -4: Time out time base selection error

◇ -5: Time out value can not be null, if a time base is selected

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	24	0x1800	0x0018
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	8	0x0800	0x0008

MODBUS interface description

word count	2	16-bit integer	8	0x0800	0x0008
byte count	2	16-bit integer	16	0x1000	0x0010
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x??????
ulFIFOLevel	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	8	0x0800	0x0008
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

MSXE170x__DigitalIOWriteAllChannel

Description

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

For new application(s) or automat communication is it recommended to use the function

MSXE17xx__DigitalIOWriteAllChannelsValue.

[Query frame layout]

◇ **ulValue** : Channels value

[Response frame layout]

◆ Possible return value on the remote system:

◇ 0: means the remote function performed OK

◇ -1: means an system error occurred

◇ -100: Write digital I/O kernel function error

Query frame layout

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

MODBUS interface description

protocol identifier		16-bit integer			
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	7	0x0700	0x0007
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7	0x0700	0x0007
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

FC23 (read/write registers) Functions

[Top](#)

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

This functions permits to call a write (FC16) and then a read(FC3) function in one command.

Query frame layout

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

MODBUS interface description

		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
MODBUS_ILLEGAL_FUNCTION
MODBUS_ILLEGAL_DATA_ADDRESS
MODBUS_ILLEGAL_DATA_VALUE
MODBUS_ILLEGAL_DATA_RESPONSE_LENGTH
MODBUS_ACKNOWLEDGE
MODBUS_DSLAVE_DEVICE_BUSY
MODBUS_NEGATIVE_ACKNOWLEDGE
MODBUS_MEMORY_PARITY_ERROR
MODBUS_REMOTE_EXECUTION_ERROR
MODBUS_GATEWAY_PATH_UNAVAILABLE
MODBUS_GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND

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
MXCommon__GetModuleType
MXCommon__GetTime
MXCommon__TestCustomerID
MSXE17xx__MFINcCounterRead32BitsValueAll
MSXE17xx__MFINcCounterRead32BitsValue0
MSXE17xx__MFINcCounterRead32BitsValue1
MSXE17xx__MFINcCounterRead32BitsValue2
MSXE17xx__MFINcCounterRead32BitsValue3
MSXE17xx__DigitalIOReadAllChannelsValue
MSXE17xx__DigitalIOTestShortCircuit
MSXE17xx__IOWatchdogGetStatusAndValue
MSXE170x__DigitalIOReadAllChannelsValue
MXCommon__SetHardwareTriggerFilterTime
MXCommon__InitAndStartSynchroTimer
MXCommon__StopAndReleaseSynchroTimer
MXCommon__Reboot
MXCommon__SetCustomerKey
MXCommon__SetFilterChannels
MSXE17xx__MFCommonSetInputsFilter
MSXE17xx__MFCommonReferenceVoltageActivation
MSXE17xx__MFCommonSetFIFO0Level
MSXE17xx__MFCommonEnableDisableTriggerGate
MSXE17xx__MFINcCounterInit
MSXE17xx__MFINcCounterRelease
MSXE17xx__MFINcCounterClear
MSXE17xx__MFINcCounterWrite32BitValue
MSXE17xx__MFINcCounterInitAndEnableCompareLogic
MSXE17xx__MFINcCounterDisableAndReleaseCompareLogic
MSXE17xx__MFINcCounterInitHardwareTrigger
MSXE17xx__MFINcCounterReleaseHardwareTrigger
MSXE17xx__MFINcCounterInitIndex
MSXE17xx__MFINcCounterReleaseIndex
MSXE17xx__MFINcCounterInitAndEnableLatch

MODBUS interface description

MSXE17xx__MFIncCounterDisableAndReleaseLatch
MSXE17xx__MFIncCounterInitAndEnableClear
MSXE17xx__MFIncCounterDisableAndReleaseClear
MSXE17xx__MFIncCounterInitAndEnableFrequencyMeasurement
MSXE17xx__MFIncCounterDisableAndReleaseFrequencyMeasurement
MSXE17xx__DigitalIOWriteAllChannelsValue
MSXE17xx__DigitalIORearmShortCircuit
MSXE17xx__DigitalIOInitPort
MSXE17xx__IOWatchdogInitAndStart
MSXE17xx__IOWatchdogStopAndRelease
MSXE170x__MFIncCounterInit
MSXE170x__MFIncCounterRelease
MSXE170x__MFIncCounterClear
MSXE170x__MFIncCounterInitAndEnableLatchRegister
MSXE170x__MFIncCounterDisableAndReleaseLatchRegister
MSXE170x__MFIncCounterWrite32BitValue
MSXE170x__MFIncCounterSetFIFO0Level
MSXE170x__DigitalIOWriteAllChannelsValue