

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> | 4 |
| 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 | 5 |
| Exception frame layout | 5 |
| Function <u>GetLastCommandStatusEx</u> | 6 |
| Description | 6 |
| Query frame layout | 6 |
| Response frame layout | 7 |
| 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 | 9 |
| Exception frame layout | 9 |
| Function <u>MXCommon GetModuleTypeEx</u> | 10 |
| Description | 10 |
| 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 | 12 |
| Response frame layout | 12 |
| Exception frame layout | 13 |
| 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> | 18 |
| <u>Exception frame layout</u> | 18 |
| <u>Function MSXE312x AnalogInputGetNumberOfChannels</u> | 19 |
| <u>Description</u> | 19 |
| <u>Query frame layout</u> | 19 |
| <u>Response frame layout</u> | 20 |
| <u>Exception frame layout</u> | 20 |
| <u>Function MSXE312x AnalogInputGetChannelsType</u> | 21 |
| <u>Description</u> | 21 |
| <u>Query frame layout</u> | 21 |
| <u>Response frame layout</u> | 22 |
| <u>Exception frame layout</u> | 22 |
| <u>Function MSXE312x AcquisitionGetNumberOfChannels</u> | 23 |
| <u>Description</u> | 23 |
| <u>Query frame layout</u> | 23 |
| <u>Response frame layout</u> | 24 |
| <u>Exception frame layout</u> | 24 |
| <u>Function MSXE312x AcquisitionGetChannelsInfo</u> | 25 |
| <u>Description</u> | 25 |
| <u>Query frame layout</u> | 25 |
| <u>Response frame layout</u> | 26 |
| <u>Exception frame layout</u> | 26 |
| <u>Function MSXE312x AcquisitionAutoRefreshGetValues</u> | 27 |
| <u>Description</u> | 27 |
| <u>Query frame layout</u> | 27 |
| <u>Response frame layout</u> | 28 |
| <u>Exception frame layout</u> | 29 |
| <u>Function MSXE312x AcquisitionAutoRefreshGetBlockingValues</u> | 29 |
| <u>Description</u> | 29 |
| <u>Query frame layout</u> | 30 |
| <u>Response frame layout</u> | 30 |
| <u>Exception frame layout</u> | 31 |
| <u>Function MSXE312x AcquisitionAutoRefreshGetConfiguration</u> | 31 |
| <u>Description</u> | 31 |
| <u>Query frame layout</u> | 33 |
| <u>Response frame layout</u> | 33 |
| <u>Exception frame layout</u> | 34 |
| <u>Function MSXE312x AcquisitionSequenceGetConfiguration</u> | 35 |
| <u>Description</u> | 35 |
| <u>Query frame layout</u> | 36 |
| <u>Response frame layout</u> | 37 |
| <u>Exception frame layout</u> | 38 |
| <u>Function MSXE312x AnalogOutputGetNumberOfChannels</u> | 38 |

Table of Contents

FC3 (read multiple register) Functions

| | |
|---|----|
| <u>Description</u> | 39 |
| <u>Query frame layout</u> | 39 |
| <u>Response frame layout</u> | 39 |
| <u>Exception frame layout</u> | 40 |
| <u>Function MSXE312x DigitalIOGetNumberOfChannels</u> | 40 |
| <u>Description</u> | 40 |
| <u>Query frame layout</u> | 41 |
| <u>Response frame layout</u> | 41 |
| <u>Exception frame layout</u> | 42 |
| <u>Function MSXE312x DigitalIOGetPortAvailableDirections</u> | 42 |
| <u>Description</u> | 42 |
| <u>Query frame layout</u> | 43 |
| <u>Response frame layout</u> | 43 |
| <u>Exception frame layout</u> | 44 |
| <u>Function MSXE312x DigitalIOGetPortDirections</u> | 44 |
| <u>Description</u> | 44 |
| <u>Query frame layout</u> | 45 |
| <u>Response frame layout</u> | 45 |
| <u>Exception frame layout</u> | 46 |
| <u>Function MSXE312x DigitalIOGetInputsFilterConfiguration</u> | 46 |
| <u>Description</u> | 46 |
| <u>Query frame layout</u> | 47 |
| <u>Response frame layout</u> | 47 |
| <u>Exception frame layout</u> | 48 |
| <u>Function MSXE312x DigitalIOTestOutputsShortCircuit</u> | 48 |
| <u>Description</u> | 48 |
| <u>Query frame layout</u> | 49 |
| <u>Response frame layout</u> | 49 |
| <u>Exception frame layout</u> | 50 |
| <u>Function MSXE312x DigitalIOReadChannel</u> | 50 |
| <u>Description</u> | 50 |
| <u>Query frame layout</u> | 51 |
| <u>Response frame layout</u> | 51 |
| <u>Exception frame layout</u> | 52 |
| <u>Function MSXE312x DigitalIOReadPort</u> | 52 |
| <u>Description</u> | 52 |
| <u>Query frame layout</u> | 52 |
| <u>Response frame layout</u> | 53 |
| <u>Exception frame layout</u> | 53 |

FC16 (write multiple register) Functions.....55

| | |
|--|----|
| <u>Function MXCommon SetHardwareTriggerFilterTime</u> | 55 |
| For new application(s) or automate communication it is recommended to use the function | |
| <u>MXCommon SetHardwareTriggerFilterTimeEx</u> | 55 |
| <u>Description</u> | 55 |
| <u>Query frame layout</u> | 56 |
| <u>Response frame layout</u> | 57 |

Table of Contents

FC16 (write multiple register) Functions

| | |
|--|----|
| Exception frame layout..... | 57 |
| Function MXCommon SetHardwareTriggerFilterTimeEx..... | 58 |
| Description..... | 58 |
| Query frame layout..... | 58 |
| Response frame layout..... | 59 |
| Exception frame layout..... | 59 |
| Function MXCommon InitAndStartSynchroTimer..... | 60 |
| For new application(s) or automate communication it is recommended to use the function | |
| MXCommon InitAndStartSynchroTimerEx..... | 60 |
| Description..... | 60 |
| Query frame layout..... | 61 |
| Response frame layout..... | 62 |
| Exception frame layout..... | 62 |
| Function MXCommon InitAndStartSynchroTimerEx..... | 63 |
| Description..... | 63 |
| Query frame layout..... | 64 |
| Response frame layout..... | 65 |
| Exception frame layout..... | 66 |
| Function MXCommon StopAndReleaseSynchroTimer..... | 66 |
| For new application(s) or automate communication it is recommended to use the function | |
| MXCommon StopAndReleaseSynchroTimerEx..... | 66 |
| Description..... | 66 |
| Query frame layout..... | 67 |
| Response frame layout..... | 67 |
| Exception frame layout..... | 68 |
| Function MXCommon StopAndReleaseSynchroTimerEx..... | 68 |
| Description..... | 68 |
| Query frame layout..... | 69 |
| Response frame layout..... | 69 |
| Exception frame layout..... | 70 |
| Function MXCommon Reboot..... | 70 |
| For new application(s) or automate communication it is recommended to use the function | |
| MXCommon RebootEx..... | 70 |
| Description..... | 70 |
| Query frame layout..... | 71 |
| Response frame layout..... | 71 |
| Exception frame layout..... | 72 |
| Function MXCommon RebootEx..... | 72 |
| Description..... | 72 |
| Query frame layout..... | 73 |
| Response frame layout..... | 73 |
| Exception frame layout..... | 74 |
| Function MXCommon SetCustomerKey..... | 74 |
| For new application(s) or automate communication it is recommended to use the function | |
| MXCommon SetCustomerKeyEx..... | 74 |
| Description..... | 74 |
| Query frame layout..... | 75 |

Table of Contents

FC16 (write multiple register) Functions

| | |
|---|-----|
| <u>Response frame layout</u> | 75 |
| <u>Exception frame layout</u> | 76 |
| <u>Function MXCommon SetCustomerKeyEx</u> | 76 |
| <u>Description</u> | 76 |
| <u>Query frame layout</u> | 77 |
| <u>Response frame layout</u> | 77 |
| <u>Exception frame layout</u> | 78 |
| <u>Function MSXE312x SetDataCursor</u> | 78 |
| <u>Description</u> | 78 |
| <u>Query frame layout</u> | 79 |
| <u>Response frame layout</u> | 79 |
| <u>Exception frame layout</u> | 80 |
| <u>Function MSXE312x AnalogInputSetChannelConfiguration</u> | 80 |
| <u>Description</u> | 80 |
| <u>Query frame layout</u> | 81 |
| <u>Response frame layout</u> | 82 |
| <u>Exception frame layout</u> | 83 |
| <u>Function MSXE312x AnalogInputSetSamplingRate</u> | 83 |
| <u>Description</u> | 83 |
| <u>Query frame layout</u> | 84 |
| <u>Response frame layout</u> | 85 |
| <u>Exception frame layout</u> | 85 |
| <u>Function MSXE312x AcquisitionAutoRefreshInitAndStart</u> | 86 |
| <u>Description</u> | 86 |
| <u>Query frame layout</u> | 88 |
| <u>Response frame layout</u> | 89 |
| <u>Exception frame layout</u> | 90 |
| <u>Function MSXE312x AcquisitionAutoRefreshStopAndRelease</u> | 90 |
| <u>Description</u> | 90 |
| <u>Query frame layout</u> | 90 |
| <u>Response frame layout</u> | 91 |
| <u>Exception frame layout</u> | 92 |
| <u>Function MSXE312x AcquisitionSequenceInitAndStart</u> | 92 |
| <u>Description</u> | 92 |
| <u>Query frame layout</u> | 94 |
| <u>Response frame layout</u> | 95 |
| <u>Exception frame layout</u> | 96 |
| <u>Function MSXE312x AcquisitionSequenceStopAndRelease</u> | 96 |
| <u>Description</u> | 97 |
| <u>Query frame layout</u> | 97 |
| <u>Response frame layout</u> | 97 |
| <u>Exception frame layout</u> | 98 |
| <u>Function MSXE312x AnalogOutputWrite1Value</u> | 98 |
| <u>Description</u> | 99 |
| <u>Query frame layout</u> | 100 |
| <u>Response frame layout</u> | 101 |
| <u>Exception frame layout</u> | 101 |

Table of Contents

FC16 (write multiple register) Functions

| | |
|--|-----|
| <u>Function MSXE312x AnalogOutputTriggerOutputs</u> | 102 |
| <u>Description</u> | 102 |
| <u>Query frame layout</u> | 102 |
| <u>Response frame layout</u> | 103 |
| <u>Exception frame layout</u> | 103 |
| <u>Function MSXE312x DigitalIOSetPortDirections</u> | 104 |
| <u>Description</u> | 104 |
| <u>Query frame layout</u> | 104 |
| <u>Response frame layout</u> | 105 |
| <u>Exception frame layout</u> | 106 |
| <u>Function MSXE312x DigitalIOSetInputsFilterTime</u> | 106 |
| <u>Description</u> | 106 |
| <u>Query frame layout</u> | 107 |
| <u>Response frame layout</u> | 107 |
| <u>Exception frame layout</u> | 108 |
| <u>Function MSXE312x DigitalIOEnableDisableInputsFilter</u> | 108 |
| <u>Description</u> | 108 |
| <u>Query frame layout</u> | 109 |
| <u>Response frame layout</u> | 109 |
| <u>Exception frame layout</u> | 110 |
| <u>Function MSXE312x DigitalIORearmOutputsShortCircuit</u> | 110 |
| <u>Description</u> | 111 |
| <u>Query frame layout</u> | 111 |
| <u>Response frame layout</u> | 112 |
| <u>Exception frame layout</u> | 112 |
| <u>Function MSXE312x DigitalIOWriteChannel</u> | 113 |
| <u>Description</u> | 113 |
| <u>Query frame layout</u> | 113 |
| <u>Response frame layout</u> | 114 |
| <u>Exception frame layout</u> | 114 |
| <u>Function MSXE312x DigitalIOWritePort</u> | 115 |
| <u>Description</u> | 115 |
| <u>Query frame layout</u> | 115 |
| <u>Response frame layout</u> | 116 |
| <u>Exception frame layout</u> | 117 |

FC23 (read/write registers) Functions.....118

| | |
|-------------------------------------|-----|
| <u>Query frame layout</u> | 118 |
| <u>Response frame layout</u> | 119 |
| <u>Exception frame layout</u> | 119 |

Exception code description.....120

Siemens Step 7 compatibility information (AWL/SDF code).....121

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>MSXE312x_AnalogInputGetNumberOfChannels</u> | Register: 23000 |
| • <u>MSXE312x_AnalogInputGetChannelsType</u> | Register: 23050 |
| • <u>MSXE312x_AcquisitionGetNumberOfChannels</u> | Register: 15000 |
| • <u>MSXE312x_AcquisitionGetChannelsInfo</u> | Register: 15050 |
| • <u>MSXE312x_AcquisitionAutoRefreshGetValues</u> | Register: 15300 |
| • <u>MSXE312x_AcquisitionAutoRefreshGetBlockingValues</u> | Register: 15450 |
| • <u>MSXE312x_AcquisitionAutoRefreshGetConfiguration</u> | Register: 15600 |
| • <u>MSXE312x_AcquisitionSequenceGetConfiguration</u> | Register: 15650 |
| • <u>MSXE312x_AnalogOutputGetNumberOfChannels</u> | Register: 35000 |
| • <u>MSXE312x_DigitalIOGetNumberOfChannels</u> | Register: 26000 |
| • <u>MSXE312x_DigitalIOGetPortAvailableDirections</u> | Register: 26100 |
| • <u>MSXE312x_DigitalIOGetPortDirections</u> | Register: 26150 |

- MSXE312x DigitalIOGetInputsFilterConfiguration Register: **26200**
- MSXE312x DigitalIOTestOutputsShortCircuit Register: **26250**
- MSXE312x DigitalIOReadChannel Register: **26300**
- MSXE312x DigitalIOReadPort Register: **26350**

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 | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|----|--------|--------|
| code | | | | | |
| Reference number (=register) | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| word count | 2 | 16-bit integer | 54 | 0x3600 | 0x0036 |

Response frame layout

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

Exception frame layout

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

Query frame layout

MODBUS interface description

| | | | | | |
|----------------------|---|----------------|---------------------------|--------------|--------------|
| | | 16-bit integer | | | |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 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 |

Exception frame layout

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|-------|--------|--------|
| Reference number (=register) | 2 | 16-bit integer | 10000 | 0x1027 | 0x2710 |
| word count | 2 | 16-bit integer | 54 | 0x3600 | 0x0036 |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|---------------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 111 | 0x6F00 | 0x006F |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Byte count | 1 | 8-bit integer | 108 | 0x6C | 0x6C |
| 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 |

Query frame layout

MODBUS interface description

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

Function MXCommon__GetModuleType

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

Description

Returns the type of the MSX-E Module

Parameters:

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

Query frame layout

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

Exception frame layout

Response frame layout

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

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 |

MODBUS interface description

| | | | | | |
|----------------------|-----|---------------------|---------------------------|--------------|--------------|
| length | 2 | 16-bit integer | 203 | 0xCB00 | 0x00CB |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Byte count | 1 | 8-bit integer | 200 | 0xC8 | 0xC8 |
| str | 200 | 8-bit integer array | See the description above | 0x??[200] | 0x??[200] |

Exception frame layout

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

Function MXCommon__GetTime

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

Description

Get the time on the module

Parameters:

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

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

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 | 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 | See the | 0x??????? | 0x??????? |

MODBUS interface description

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

Exception frame layout

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

Function MXCommon__GetTimeEx

Description

Get the time on the module

Parameters:

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

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

Query frame layout

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

Exception frame layout

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|--------|--------------|--------------|
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 3 | 0x0300 | 0x0003 |
| word count | 2 | 16-bit integer | 16 | 0x1000 | 0x0010 |

Response frame layout

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

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function 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 |
| | 2 | | 10550 | 0x3629 | 0x2936 |

Exception frame layout

MODBUS interface description

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

Response frame layout

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

Exception frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit 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 | 1 | 8-bit integer | 0x83 | 0x83 | 0x83 |

Query frame layout

MODBUS interface description

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

Function MSXE312x__AnalogInputGetNumberOfChannels

Description

Return the number of analog input channels.

Parameters

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

Returns

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - 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 | 23000 | 0xD859 | 0x59D8 |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Response frame layout

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

Exception frame layout

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

Function MSXE312x__AnalogInputGetChannelsType

Description

Return the type of the analog input channels.

Parameters

- [Response frame layout] **ulType** Array that contain the channels type (0 : Voltage, 1 : Current)
 - ◆ ulType [0] : Channel 0 type
 - ◆ ...
 - ◆ ulType [7] : Channel 7 type
- [Response frame layout] **ulICP** Array that contain if ICP available (0 : not available, 1 : available)
 - ◆ ulICP [0] : Channel 0 ICP information
 - ◆ ...
 - ◆ ulICP [7] : Channel 7 ICP information

Returns

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 23050 | 0x0A5A | 0x5A0A |
| word count | 2 | | 32 | 0x2000 | 0x0020 |

MODBUS interface description

| | | | | | |
|--|--|----------------|--|--|--|
| | | 16-bit integer | | | |
|--|--|----------------|--|--|--|

Response frame layout

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

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 | ?? | ?? |

Query frame layout

Function MSXE312x__AcquisitionGetNumberOfChannels

Description

Return the number of acquisition channels.

Parameters

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

Returns

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - 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 | 15000 | 0x983A | 0x3A98 |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Response frame layout

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

Exception frame layout

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

Function MSXE312x__AcquisitionGetChannelsInfo

Description

Return the acquisition channels type and hardware position.

Parameters:

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

Returns

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

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

Query frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|--------|--------------|--------------|
| protocol identifier | | 16-bit integer | | | |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 15050 | 0xCA3A | 0x3ACA |
| word count | 2 | 16-bit integer | 96 | 0x6000 | 0x0060 |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|----------------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 195 | 0xC300 | 0x00C3 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Byte count | 1 | 8-bit integer | 192 | 0xC0 | 0xC0 |
| ulType | 64 | 32-bit integer array | See the description above | 0x????????[16] | 0x????????[16] |
| ulHwPosition | 64 | 32-bit integer array | See the description above | 0x????????[16] | 0x????????[16] |
| ulChannelIndex | 64 | 32-bit integer array | See the description above | 0x????????[16] | 0x????????[16] |

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MSXE312x__AcquisitionAutoRefreshGetValues

Description

Returns the values acquired in auto refresh mode

Parameters:

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

Returns:

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

- ◆ **0** The remote function performed OK
- ◆ **-2** No Acquisition in progress
- ◆ **-3** 2s timeout occur (If you have enabled the blocking mode).
- ◆ **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

Exception frame layout

MODBUS interface description

| | | | | | |
|------------------------------------|---|-------------------|--------------------------------|-----------------|-----------------|
| | | | by server - usually 0 | | |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 15300 | 0xC43B | 0x3BC4 |
| word count | 2 | 16-bit integer | 38 | 0x2600 | 0x0026 |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|---------------------------|-----------------|--------------------|---|--------------------------|--------------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 79 | 0x4F00 | 0x004F |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Byte count | 1 | 8-bit integer | 76 | 0x4C | 0x4C |
| ulTimeStampLow | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulTimeStampHigh | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulCounterValue | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| fValues | 64 | 32-bit floating | See the description | 0x???????[16] | 0x???????[16] |

MODBUS interface description

| | | | | | |
|--|--|----------------|-------|--|--|
| | | point array | 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

MSXE312x__AcquisitionAutoRefreshGetBlockingValues

Description

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

Parameters:

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

Returns:

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

- ◇ 0 The remote function performed OK

MODBUS interface description

- ◇ -2 No Acquisition in progress
- ◇ -3 2s timeout occur (If you have enabled the blocking mode).
- ◇ -100 Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 15450 | 0x5A3C | 0x3C5A |
| word count | 2 | 16-bit integer | 38 | 0x2600 | 0x0026 |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 79 | 0x4F00 | 0x004F |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Byte count | 1 | 8-bit integer | 76 | 0x4C | 0x4C |

MODBUS interface description

| | | | | | |
|-----------------|----|-----------------------------|---------------------------|----------------|----------------|
| ulTimeStampLow | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulTimeStampHigh | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulCounterValue | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| fValues | 64 | 32-bit floating point array | See the description above | 0x????????[16] | 0x????????[16] |

Exception frame layout

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

Function

MSXE312x__AcquisitionAutoRefreshGetConfiguration

Description

Get the current auto refresh acquisition configuration.

Parameters:

- ◇ [Response frame layout] **ulChannelMask** Mask of the channel acquired by the auto refresh (1 bit = 1 Channel). If the value returned is 0, then the auto refresh acquisition is not initialised.
- ◇ [Response frame layout] **ulAverageValue** Average value

MODBUS interface description

- 1 : not used
- max value : 255
- ◇ [Response frame layout] **ulRefreshTimeUnit** Refresh Time Unit
 - 0 : microsecond
 - 1 : millisecond
 - 2 : second
- ◇ [Response frame layout] **ulRefreshTime** Refresh Time
 - range from min 10 to 65535 when the unit is the microsecond
 - range from min 1 to 65535 when the unit is the millisecond
 - range from min 1 to 65535 when the unit is the second
- ◇ [Response frame layout] **ulTriggerMask** Define the source of the trigger
 - 0 : trigger disabled
 - 1 : Enable Hardware Digital Input Trigger
 - 2 : Enable Synchro Trigger
 - 3 : Enable Compare Trigger (only useful if your system has incremental counter input)
- ◇ [Response frame layout] **ulTriggerMode** Define the trigger mode
 - 1 : One shot trigger
 - 2 : Sequence trigger
- ◇ [Response frame layout] **ulHardwareTriggerEdge** Define the edge of the hardware trigger who generates a trigger action
 - 1 : rising edge (Only if hardware trigger selected)
 - 2 : falling edge (Only if hardware trigger selected)
 - 3 : Both front (Only if hardware trigger selected)
- ◇ [Response frame layout] **ulHardwareTriggerCount** Define the number of trigger events before the action occur
 - 1 : all trigger event start the action
 - max value : 65535
- ◇ [Response frame layout] **ulByTriggerNbrOfSeqToAcquire** Define the number of sequence to acquire by each trigger event
 - 0 : continuous mode
 - 0 : number of sequence : (1..0xFFFFFFFF)
- ◇ [Response frame layout] **ulDataFormat**
 - D0 : Absolute time stamp information
 - 0: no time stamp information
 - 1: time stamp information
 - D1 : Relative time stamp information
 - 0: no time stamp information
 - 1: time stamp information
 - D2 : Auto refresh counter information
 - 0 : No auto refresh counter information
 - 1 : Auto refresh counter information
 - D3 : Hardware trigger information
 - 0 : No hardware trigger information required
 - 1 : Hardware trigger information required
 - D4 : Data format
 - 0: Digital value
 - 1: Analog value
- ◇ [Response frame layout] **ulRunning** Auto refresh acquisition running state
 - 0 : Not running
 - 1 : Auto refresh acquisition running

MODBUS interface description

Returns:

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 15600 | 0xF03C | 0x3CF0 |
| word count | 2 | 16-bit integer | 22 | 0x1600 | 0x0016 |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 47 | 0x2F00 | 0x002F |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | | 0x03 | 0x03 | 0x03 |

MODBUS interface description

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

Exception frame layout

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

Response frame layout

MODBUS interface description

| | | | | | |
|----------------------|---|----------------|---------------------------|--------------|--------------|
| | | 16-bit integer | | | |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x83 | 0x83 | 0x83 |
| Exception code | 1 | 8-bit integer | See corresponding chapter | ?? | ?? |

Function

MSXE312x__AcquisitionSequenceGetConfiguration

Description

Get the current sequence acquisition configuration.

Parameters

- ◇ [Response frame layout] **ulNbrOfChannel** Nbr of channel in the sequence. If 0 no sequence initialised
- ◇ [Response frame layout] **psChannelList** List of the channel who compose the sequence.
- ◇ [Response frame layout] **ulAcquisitionTime** Acquisition Time
 - range from min 10 to 65535 when the unit is the microsecond
 - range from min 1 to 65535 when the unit is the millisecond
 - range from min 1 to 65535 when the unit is the second
- ◇ [Response frame layout] **ulAcquisitionTimeUnit** Acquisition Time Unit
 - 0 : us
 - 1 : ms
 - 2 : s
- ◇ [Response frame layout] **ulNbrOfSequence** Number of sequence to acquire
 - 0 : continuous mode
 - Superior to 0 : number of sequence
- ◇ [Response frame layout] **ulNbrMaxSequenceToTransfer** Max number of sequence to acquire before a data transfer. (from 1 to 65535)
- ◇ [Response frame layout] **ulTriggerMask** Define the source of the trigger
 - 0 : trigger disabled
 - 1 : Enable Hardware Digital Input Trigger
 - 2 : Enable Synchro Trigger
 - 3 : Enable Compare Trigger (only useful if your system has incremental counter input)
- ◇ [Response frame layout] **ulTriggerMode** Define the trigger mode
 - 1 : One shot trigger
 - 2 : Sequence trigger
- ◇ [Response frame layout] **ulHardwareTriggerEdge** Define the edge of the hardware trigger who generate a trigger action
 - 1 : rising front (Only if hardware trigger selected)

MODBUS interface description

- 2 : falling front (Only if hardware trigger selected)
- 3 : Both front (Only if hardware trigger selected)
- ◇ [Response frame layout] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
 - 1 : all trigger event start the action
 - max value : 65535
- ◇ [Response frame layout] ***ulByTriggerNbrOfSeqToAcquire*** define the number of sequence to acquire by each trigger event
 - 0 : continuous mode
 - Superior to 0 : number of sequence : (1..0xFFFFFFFF)
- ◇ [Response frame layout] ***ulDataFormat*** Data format option
 - D0 : Absolute time stamp information
 - 0: no time stamp information
 - 1: time stamp information
 - D1 : Relative time stamp information
 - 0: no time stamp information
 - 1: time stamp information
 - D2 : Sequence counter information
 - 0 : No sequence counter information
 - 1 : Sequence counter information
 - D3 : Hardware trigger information
 - 0 : No hardware trigger information required
 - 1 : Hardware trigger information required
 - D4 : Data format
 - 0: Digital value
 - 1: Analog value
- ◇ [Response frame layout] ***ulRunning*** Sequence acquisition running state
 - 0 : Not running
 - 1 : Sequence acquisition running

Returns:

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

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

Query frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|--------|--------------|--------------|
| | | 16-bit integer | | | |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 15650 | 0x223D | 0x3D22 |
| word count | 2 | 16-bit integer | 56 | 0x3800 | 0x0038 |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|----------------------------|--------------|----------------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 115 | 0x7300 | 0x0073 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Byte count | 1 | 8-bit integer | 112 | 0x70 | 0x70 |
| ulNbrOfChannel | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulChannelList | 64 | 32-bit integer array | See the description above | 0x???????[16] | 0x???????[16] |
| ulAcquisitionTime | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulAcquisitionTimeUnit | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulNbrOfSequence | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulNbrMaxSequenceToTransfer | 4 | 32-bit integer | See the description | 0x??????? | 0x??????? |

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|---------------------------|------------|------------|
| | | | above | | |
| ulTriggerMask | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulTriggerMode | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulHardwareTriggerEdge | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulHardwareTriggerCount | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulByTriggerNbrOfSeqToAcquire | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulDataFormat | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulRunning | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |

Exception frame layout

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

Function

MSXE312x__AnalogOutputGetNumberOfChannels

Description

Return the number of analog output channels.

Parameters

◇ [Response frame layout] **ulNumber** Return the number of available analog output channels

Returns

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

◇ **0** The remote function performed OK

◇ **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - 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 | 35000 | 0xB888 | 0x88B8 |
| 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 - | 0x0000 | 0x0000 |

MODBUS interface description

| | | | | | |
|----------------------|---|----------------|---------------------------|--------------|--------------|
| | | | usually 0 | | |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 7 | 0x0700 | 0x0007 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| ulNumber | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Exception frame layout

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

Function

MSXE312x__DigitalIOGetNumberOfChannels

Description

Returns the number of digital I/O channels.

Parameters

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

Returns

MODBUS interface description

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

◇ 0 The remote function performed OK

◇ -100 Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 26000 | 0x9065 | 0x6590 |
| 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 |

MODBUS interface description

| | | | | | |
|------------|---|----------------|---------------------------|------------|------------|
| Byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| ulNumber | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |

Exception frame layout

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

Function

MSXE312x__DigitalIOGetPortAvailableDirections

Description

Returns the available directions for the selected port (input or output). The port is selected via MSXE312x__SetDataCursor.

Parameters

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

Returns

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The selected port (selected via MSXE312x__SetDataCursor) is wrong
- ◇ **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 26100 | 0xF465 | 0x65F4 |
| word count | 2 | 16-bit integer | 4 | 0x0400 | 0x0004 |

Response frame layout

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

MODBUS interface description

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

Exception frame layout

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

Function MSXE312x__DigitalIOGetPortDirections

Description

Reads the current digital I/O direction for the selected port. The port is selected via MSXE312x__SetDataCursor.

Parameters

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

Returns

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The selected port (selected via MSXE312x__SetDataCursor) is wrong
- ◇ **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 26150 | 0x2666 | 0x6626 |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Response frame layout

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

MODBUS interface description

| | | | | | |
|--|--|---------|-------------------|--|--|
| | | integer | 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

MSXE312x__DigitalIOGetInputsFilterConfiguration

Description

Reads the digital inputs filter configuration for the selected port. The port is selected via MSXE312x__SetDataCursor.

Parameters

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

Returns

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The selected port (selected via MSXE312x__SetDataCursor) is wrong
- ◇ **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 26200 | 0x5866 | 0x6658 |
| word count | 2 | 16-bit integer | 4 | 0x0400 | 0x0004 |

Response frame layout

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

MODBUS interface description

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

Exception frame layout

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

Function

MSXE312x__DigitalIOTestOutputsShortCircuit

Description

Get the short-circuit status of the outputs of the selected port. The port is selected via MSXE312x__SetDataCursor.

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

If you detect a short circuit, first solve it, and then, call the MSXE312x__DigitalIORearmOutputsShortCircuit function.

Parameters

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

Returns

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

MODBUS interface description

- ◇ 0 The remote function performed OK
- ◇ -2 The selected port (selected via MSXE312x__SetDataCursor) is wrong
- ◇ -100 Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 26250 | 0x8A66 | 0x668A |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Response frame layout

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

MODBUS interface description

| | | | | | |
|---------|---|----------------|---------------------------|------------|------------|
| | | 8-bit integer | | | |
| ulState | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |

Exception frame layout

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

Function MSXE312x__DigitalIOReadChannel

Description

Read the selected digital I/O channel. The channel is selected via MSXE312x__SetDataCursor.

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

Parameters

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

Returns

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The selected channel (selected via MSXE312x__SetDataCursor) is wrong
- ◇ **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 26300 | 0xBC66 | 0x66BC |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Response frame layout

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

MODBUS interface description

| | | | | | |
|--|--|---------|-------------------|--|--|
| | | integer | 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 MSXE312x__DigitalIOReadPort

Description

Read the selected digital I/O port. The port is selected via MSXE312x__SetDataCursor.

Parameters

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

Returns

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

- ◇ **0** The remote function performed OK
- ◇ **-2** The selected port (selected via MSXE312x__SetDataCursor) is wrong
- ◇ **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|--------------------------------|--------------|--------------|
| | | | - copied by server - usually 0 | | |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x03 | 0x03 | 0x03 |
| Reference number (=register) | 2 | 16-bit integer | 26350 | 0xEE66 | 0x66EE |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Response frame layout

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

Exception frame layout

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

MODBUS interface description

| | | | | | |
|------------------------|---|----------------|---|--------------|--------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 3 | 0x0300 | 0x0003 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 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.

| | |
|--|------------------------|
| ◇ <u>MXCommon_SetHardwareTriggerFilterTime</u> | Register: 100 |
| ◇ <u>MXCommon_SetHardwareTriggerFilterTimeEx</u> | Register: 11000 |
| ◇ <u>MXCommon_InitAndStartSynchroTimer</u> | Register: 101 |
| ◇ <u>MXCommon_InitAndStartSynchroTimerEx</u> | Register: 11050 |
| ◇ <u>MXCommon_StopAndReleaseSynchroTimer</u> | Register: 102 |
| ◇ <u>MXCommon_StopAndReleaseSynchroTimerEx</u> | Register: 11100 |
| ◇ <u>MXCommon_Reboot</u> | Register: 103 |
| ◇ <u>MXCommon_RebootEx</u> | Register: 11150 |
| ◇ <u>MXCommon_SetCustomerKey</u> | Register: 104 |
| ◇ <u>MXCommon_SetCustomerKeyEx</u> | Register: 11200 |
| ◇ <u>MSXE312x_SetDataCursor</u> | Register: 65530 |
| ◇ <u>MSXE312x_AnalogInputSetChannelConfiguration</u> | Register: 23100 |
| ◇ <u>MSXE312x_AnalogInputSetSamplingRate</u> | Register: 23200 |
| ◇ <u>MSXE312x_AcquisitionAutoRefreshInitAndStart</u> | Register: 15750 |
| ◇ <u>MSXE312x_AcquisitionAutoRefreshStopAndRelease</u> | Register: 15850 |
| ◇ <u>MSXE312x_AcquisitionSequenceInitAndStart</u> | Register: 15900 |
| ◇ <u>MSXE312x_AcquisitionSequenceStopAndRelease</u> | Register: 15950 |
| ◇ <u>MSXE312x_AnalogOutputWrite1Value</u> | Register: 35100 |
| ◇ <u>MSXE312x_AnalogOutputTriggerOutputs</u> | Register: 35200 |
| ◇ <u>MSXE312x_DigitalIOSetPortDirections</u> | Register: 26500 |
| ◇ <u>MSXE312x_DigitalIOSetInputsFilterTime</u> | Register: 26550 |
| ◇ <u>MSXE312x_DigitalIOEnableDisableInputsFilter</u> | Register: 26600 |
| ◇ <u>MSXE312x_DigitalIORearmOutputsShortCircuit</u> | Register: 26650 |
| ◇ <u>MSXE312x_DigitalIOWriteChannel</u> | Register: 26700 |
| ◇ <u>MSXE312x_DigitalIOWritePort</u> | Register: 26750 |

Function

MXCommon__SetHardwareTriggerFilterTime

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

MXCommon__SetHardwareTriggerFilterTimeEx.

Description

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

MODBUS interface description

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

Parameters

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

Returns

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 16 | 0x1000 | 0x0010 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| 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 | 0x???????? | 0x???????? |

MODBUS interface 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 | 100 | 0x6400 | 0x0064 |
| word count | 2 | 16-bit integer | 4 | 0x0400 | 0x0004 |

Exception frame layout

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

Function

MXCommon__SetHardwareTriggerFilterTimeEx

Description

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

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

Parameters

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

- **0**: disable the filter
- **1**: filter of 250ns
- **2**: filter of 500ns
- ...
- **65535**: filter of 16ms

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

Returns

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

◇ **0** The remote function performed OK

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

Query frame layout

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

MODBUS interface description

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

Response frame layout

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

Exception frame layout

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

MODBUS interface description

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

Function MXCommon__InitAndStartSynchroTimer

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

Description

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

Parameters:

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

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

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

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

[Query frame layout] **ulGenerateTriggerMode**:

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

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

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

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

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

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

MODBUS interface description

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

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

Returns:

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

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

Query frame layout

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

MODBUS interface description

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

MODBUS interface description

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

Function 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)

MODBUS interface description

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

MODBUS interface description

| | | | | | |
|-----------------------|---|----------------|---------------------------|------------|------------|
| | | | above | | |
| 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 |
| Reference number (=register) | 2 | 16-bit integer | 11050 | 0x2A2B | 0x2B2A |
| word count | 2 | 16-bit integer | 16 | 0x1000 | 0x0010 |

Exception frame layout

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

Function

MXCommon__StopAndReleaseSynchroTimer

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

MXCommon__StopAndReleaseSynchroTimerEx.

Description

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

Parameters:

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

Returns:

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 12 | 0x0C00 | 0x000C |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 102 | 0x6600 | 0x0066 |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |
| byte count | 2 | 16-bit integer | 4 | 0x0400 | 0x0004 |
| ulOption01 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied 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 | 102 | 0x6600 | 0x0066 |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Exception frame layout

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

Function

MXCommon__StopAndReleaseSynchroTimerEx

Description

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

Parameters:

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

Returns:

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 11 | 0x0B00 | 0x000B |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 11100 | 0x5C2B | 0x2B5C |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |
| byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| ulOption01 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

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

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

Returns:

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

· 0 : means the remote function performed OK

MODBUS interface description

· -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 - 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 | 103 | 0x6700 | 0x0067 |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Exception frame layout

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

Function MXCommon__RebootEx

Description

Ask the MSX-E module to reboot

Parameters:

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

Returns:

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 11 | 0x0B00 | 0x000B |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 11150 | 0x8E2B | 0x2B8E |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |
| byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| Dummy | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

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

Response frame layout

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 (probably EPERM)

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|---------------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 56 | 0x3800 | 0x0038 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 104 | 0x6800 | 0x0068 |
| word count | 2 | 16-bit integer | 24 | 0x1800 | 0x0018 |
| byte count | 2 | 16-bit integer | 48 | 0x3000 | 0x0030 |
| bKey | 32 | 8-bit integer array | See the description above | 0x??[32] | 0x??[32] |
| bPublicKey | 16 | 8-bit integer array | See the description above | 0x??[16] | 0x??[16] |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - 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 | 104 | 0x6800 | 0x0068 |
| word count | 2 | 16-bit integer | 24 | 0x1800 | 0x0018 |

Exception frame layout

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

Function MXCommon__SetCustomerKeyEx

Description

Permit to set the Customer key

Parameters:

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

MODBUS interface description

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

Returns:

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

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

Query frame layout

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

Response frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|--------------------------------|--------------|--------------|
| | | | - copied by server - usually 0 | | |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 11200 | 0xC02B | 0x2BC0 |
| word count | 2 | 16-bit integer | 24 | 0x1800 | 0x0018 |

Exception frame layout

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

Function MSXE312x__SetDataCursor

Description

Change the active data cursor cursor

Parameters:

MODBUS interface description

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

Returns:

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

- 0: success
- -1: otherwise : internal error

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------------|-----------------|-------------------|---|-----------------------------|--------------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 11 | 0x0B00 | 0x000B |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 65530 | 0xFAFF | 0xFFFFA |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |
| byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| ulCursor | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|--------|--------------|--------------|
| | | | 0 | | |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 65530 | 0xFAFF | 0xFFFFA |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Exception frame layout

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

Function

MSXE312x__AnalogInputSetChannelConfiguration

Description

Analog input channel configuration

Parameters

- [Query frame layout] **ulChannel** Channel selection. 0 to 7
- [Query frame layout] **ulAcDc** AC/DC coupling selection

MODBUS interface description

- 0 : DC
- 1 : AC
- [Query frame layout] **ulSingleDiff** Single/differential selection
 - 0 : Single
 - 1 : Differential
- [Query frame layout] **ulPolarity** Polarity selection
 - 0 : Bipolar
 - 1 : Unipolar
- [Query frame layout] **ulGain** Gain selection
 - 1 : Gain 1x
 - 10 : Gain 10x
 - 100 : Gain 100x
 - 1000 : Gain 1000x
- [Query frame layout] **ulICP** ICP source
 - 0 : Disable the ICP source
 - 1 : Enable the ICP source
- [Query frame layout] **ulOption1** Reserved. Set to 0
- [Query frame layout] **ulOption2** Reserved. Set to 0

Returns

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

- **0** The remote function performed OK
- **-2** Channel selection wrong
- **-3** AC/DC coupling selection wrong
- **-4** Single/differential selection wrong
- **-5** Polarity selection wrong
- **-6** Gain selection wrong
- **-7** ICP selection wrong
- **-8** ICP not available
- **-9** Acquisition in progress. Can not change the configuration
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 39 | 0x2700 | 0x0027 |
| 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 | 23100 | 0x3C5A | 0x5A3C |
| word count | 2 | 16-bit integer | 16 | 0x1000 | 0x0010 |
| byte count | 1 | 8-bit integer | 32 | 0x20 | 0x20 |
| ulChannel | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulAcDc | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulSingleDiff | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulPolarity | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulGain | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulICP | 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 | | 6 | 0x0600 | 0x0006 |

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|--------|--------------|--------------|
| | | 16-bit integer | | | |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 23100 | 0x3C5A | 0x5A3C |
| 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

MSXE312x__AnalogInputSetSamplingRate

Description

Sets the sampling rate (or the sampling period) for the analog inputs.

- If you select **5** for the parameter **ulTimeBaseUnity**, then this function sets the **sampling rate**, i.e. the number of samples per second.
- If you select **0** for the parameter **ulTimeBaseUnity**, then this function sets the **sampling period**, i.e. the time between two samples.

Parameters

- [Query frame layout] **ulTimeBase** Time base selection

MODBUS interface description

- 0 for 1MHz
- 1 for 6.4MHz
- [Query frame layout] **ulTimeBaseUnity** Unity
 - 0 for μ s
 - 5 for Hz
- [Query frame layout] **ulSamplingRate** Sampling rate or sampling period according to the selected unity
- [Query frame layout] **ulOption1** Reserved. Set to 0
- [Query frame layout] **ulOption2** Reserved. Set to 0

Returns

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

- **0** The remote function performed OK
- **-2** Sampling rate time base selection wrong
- **-3** Sampling rate time base unity selection wrong
- **-4** Sampling rate selection wrong
- **-5** Acquisition in progress. Can not change the sampling rate
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 27 | 0x1B00 | 0x001B |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 23200 | 0xA05A | 0x5AA0 |
| word count | 2 | 16-bit integer | 10 | 0x0A00 | 0x000A |
| byte count | 1 | 8-bit integer | 20 | 0x14 | 0x14 |
| ulTimeBase | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulTimeBaseUnity | 4 | | | 0x??????? | 0x??????? |

MODBUS interface description

| | | | | | |
|----------------|---|----------------|---------------------------|------------|------------|
| | | 32-bit integer | See the description above | | |
| ulSamplingRate | 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 |
| Reference number (=register) | 2 | 16-bit integer | 23200 | 0xA05A | 0x5AA0 |
| word count | 2 | 16-bit integer | 10 | 0x0A00 | 0x000A |

Exception frame layout

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

MODBUS interface description

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

Function

MSXE312x__AcquisitionAutoRefreshInitAndStart

Description

Initialise and start an auto refresh acquisition using provided configuration

Parameters

- [Query frame layout] ***ulChannelMask*** Mask of the channel to acquire by the auto refresh (1 bit = 1 Channel). 0 for all available acquisition channels
- [Query frame layout] ***ulAverageValue*** Set the average value
 - 1 : not used
 - max value : 255
- [Query frame layout] ***ulRefreshTimeUnit*** Refresh Time Unit
 - 0 : microsecond
 - 1 : millisecond
 - 2 : second
- [Query frame layout] ***ulRefreshTime*** Refresh Time
 - range from min 10 to 65535 when the unit is the microsecond
 - range from min 1 to 65535 when the unit is the millisecond
 - range from min 1 to 65535 when the unit is the second
- [Query frame layout] ***ulTriggerMask*** Define the source of the trigger
 - 0 : trigger disabled
 - 1 : Enable Hardware Digital Input Trigger
 - 2 : Enable Synchro Trigger
 - 3 : Enable Compare Trigger (only useful if your system has incremental counter input)
- [Query frame layout] ***ulTriggerMode*** Define the trigger mode
 - 1 : One shot trigger
 - 2 : Sequence trigger
- [Query frame layout] ***ulHardwareTriggerEdge*** Define the edge of the hardware trigger who generates a trigger action
 - 1 : rising edge (Only if hardware trigger selected)
 - 2 : falling edge (Only if hardware trigger selected)
 - 3 : Both front (Only if hardware trigger selected)
- [Query frame layout] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur

MODBUS interface description

- 1 : all trigger event start the action
- max value : 65535
- [Query frame layout] ***ulByTriggerNbrOfSeqToAcquire*** Define the number of sequence to acquire by each trigger event
 - 0 : continuous mode
 - 0 : number of sequence : (1..0xFFFFFFFF)
- [Query frame layout] ***ulDataFormat***
 - D0 : Absolute time stamp information
 - ◆ 0: no time stamp information
 - ◆ 1: time stamp information
 - D1 : Relative time stamp information
 - ◆ 0: no time stamp information
 - ◆ 1: time stamp information
 - D2 : Auto refresh counter information
 - ◆ 0 : No auto refresh counter information
 - ◆ 1 : Auto refresh counter information
 - D3 : Hardware trigger information
 - ◆ 0 : No hardware trigger information required
 - ◆ 1 : Hardware trigger information required
 - D4 : Data format
 - ◆ 0: Digital value
 - ◆ 1: Analog value
- [Query frame layout] ***ulForceStart***
 - 0 : Function return a error if any acquisition already in progress
 - 1 : If any acquisition in progress then stop this
- [Query frame layout] ***ulOption1*** Reserved. Set to 0
- [Query frame layout] ***ulOption2*** Reserved. Set to 0
- [Query frame layout] ***ulOption3*** Reserved. Set to 0

Returns:

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

- **0** The remote function performed OK
- **-2** Any acquisition already in progress
- **-3** Any selected channel not OK, call the diagnostic function for more information
- **-4** Channel Mask error
- **-5** Not available average value
- **-6** Not available refresh time unit
- **-7** The minimal refresh time is 1000 us
- **-8** The maximal refresh time is 65535
- **-9** Trigger mask not available
- **-10** Trigger mask : 2 different trigger source cannot be simultaneously be activated
- **-11** Trigger mode not available
- **-12** Trigger mask : 2 trigger mode cannot be simultaneously activated
- **-13** Hardware trigger : front definition error
- **-14** Hardware trigger count value not available
- **-15** Nbr of sequence to acquire by trigger mode not available
- **-16** Data format not available

MODBUS interface description

- **-17** Selected channels combination not available
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 63 | 0x3F00 | 0x003F |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 15750 | 0x863D | 0x3D86 |
| word count | 2 | 16-bit integer | 28 | 0x1C00 | 0x001C |
| byte count | 1 | 8-bit integer | 56 | 0x38 | 0x38 |
| ulChannelMask | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulAverageValue | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulRefreshTime | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulRefreshTimeUnit | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulTriggerMask | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulTriggerMode | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulHardwareTriggerEdge | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulHardwareTriggerCount | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|---------------------------|------------|------------|
| | | | above | | |
| ulByTriggerNbrOfSeqToAcquire | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulDataFormat | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulForceStart | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulOption1 | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulOption2 | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulOption3 | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 15750 | 0x863D | 0x3D86 |
| word count | 2 | 16-bit integer | 28 | 0x1C00 | 0x001C |

Exception frame layout

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

Function

MSXE312x__AcquisitionAutoRefreshStopAndRelease

Description

Stops the current auto refresh acquisition

Parameters

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

Returns:

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

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

Query frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|---------------------------|--------------|--------------|
| | | | usually 0 | | |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 11 | 0x0B00 | 0x000B |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 15850 | 0xEA3D | 0x3DEA |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |
| byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| ulOption1 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 15850 | 0xEA3D | 0x3DEA |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Exception frame layout

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

Function

MSXE312x__AcquisitionSequenceInitAndStart

Description

Initialise and start the sequence acquisition mode

Parameters

- [Query frame layout] ***ulNbrOfChannel*** Number of channel in the sequence
- [Query frame layout] ***psChannelList*** List of the channel who compose the sequence.
- [Query frame layout] ***ulAcquisitionTime*** Acquisition Time
 - range from min 10 to 65535 when the unit is the microsecond
 - range from min 1 to 65535 when the unit is the millisecond
 - range from min 1 to 65535 when the unit is the second
- [Query frame layout] ***ulAcquisitionTimeUnit*** Acquisition Time Unit
 - 0 : us
 - 1 : ms
 - 2 : s
- [Query frame layout] ***ulNbrOfSequence*** Number of sequence to acquire
 - 0 : continuous mode
 - Superior to 0 : number of sequence
- [Query frame layout] ***ulNbrMaxSequenceToTransfer*** Max number of sequence to acquire before a data transfer. (from 1 to 65535)
- [Query frame layout] ***ulTriggerMask*** Define the source of the trigger

MODBUS interface description

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

Returns:

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

- **0** The remote function performed OK
- **-2** Any acquisition already in progress
- **-3** The number of channel in the sequence is null or to high

MODBUS interface description

- **-4** Channel index selection error
- **-5** Channel already selected
- **-6** Any selected channel not OK, call the diagnostic function for more information
- **-7** Not available acquisition time unit
- **-8** The minimal acquisition time is 1000 us !
- **-9** The maximal acquisition time is 65535 !
- **-10** Transfer sequence size error (1 to 4096) !
- **-11** The total number of sequences is not a multiple from number of sequences to transfer
- **-12** Trigger mask not available
- **-13** Trigger mask : 2 different trigger source cannot be simultaneously be activated
- **-14** Trigger mode not available
- **-15** Trigger mask : 2 trigger mode cannot be simultaneously be activated
- **-16** Hardware trigger : front definition error
- **-17** Hardware trigger count value not available
- **-18** Number of sequence to acquire by trigger mode not available
- **-19** Data format not available
- **-20** Selected channels combination not available
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 131 | 0x8300 | 0x0083 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 15900 | 0x1C3E | 0x3E1C |
| word count | 2 | 16-bit integer | 62 | 0x3E00 | 0x003E |
| byte count | 1 | 8-bit integer | 124 | 0x7C | 0x7C |
| ulNbrOfChannel | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulChannelList | 64 | 32-bit integer | See the description | 0x????????[16] | 0x????????[16] |

MODBUS interface description

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

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|-------|--------------|------|-------|-----------------------|-----------------------|
| | 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 | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 15900 | 0x1C3E | 0x3E1C |
| word count | 2 | 16-bit integer | 62 | 0x3E00 | 0x003E |

Exception frame layout

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

Function

MSXE312x__AcquisitionSequenceStopAndRelease

Description

Stop and release the sequence acquisition mode

Parameters

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

Returns:

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

- **0** The remote function performed OK
- **-2** No sequence acquisition in progress
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 11 | 0x0B00 | 0x000B |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 15950 | 0x4E3E | 0x3E4E |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |
| byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| ulOption01 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

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

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 | 15950 | 0x4E3E | 0x3E4E |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Exception frame layout

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

Function

MSXE312x__AnalogOutputWrite1Value

Description

Set the value of an analog output

Parameters

- [Query frame layout] **ulChannel** Index of the analog output (0 to AnalogOutputGetNumberOfChannels)
- [Query frame layout] **ulOutputType** Type of the output (0 for Voltage, 1 for Current, 2 system default)
- [Query frame layout] **ulPolarity** Polarity (0 for unipolar, 1 for bipolar)
- [Query frame layout] **ulTriggerMask** Trigger to use to set the outputs (Mask of bits)
 - **0** no trigger: output directly updated after a call to this function
 - **Bit 0 = 1**: Hardware trigger used
 - **Bit 1 = 1**: Synchro trigger used
 - **Bit 2 = 1**: Software trigger used: output updated after a call to AnalogOutputTriggerOutputs
- [Query frame layout] **ulTriggerEdgeSelection** Only used if you select the hardware trigger. Defines the detected edges (Mask of bits). The value cannot be null (if hardware trigger is selected).
 - **Bit 0 = 1**: Rising edges are detected
 - **Bit 1 = 1**: Falling edges are detected
 - **Bit 0 = 1 and Bit 1 = 1**: Rising and falling edges are detected
- [Query frame layout] **ulTriggerCount** Only used if you select the hardware trigger. Defines the number of hardware trigger to wait before updating the analog outputs. The value cannot be null (if hardware trigger is selected).
- [Query frame layout] **ulValue** Value that you want to set (digital format)
 - If unipolar, range is from 0 to 0x7fff (from 0V to +10V)
 - If bipolar, range is from 0 to 0xffff (from -10V to +10V)
- [Query frame layout] **ulOption0** Reserved. Set to 0
- [Query frame layout] **ulOption1** Reserved. Set to 0
- [Query frame layout] **ulOption2** Reserved. Set to 0

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulChannel parameter is wrong
- **-4** The ulOutputType parameter is wrong
- **-5** The ulPolarity parameter is wrong
- **-6** The ulTriggerMask parameter is wrong
- **-7** The ulValue parameter is wrong
- **-8** The ulTriggerEdgeSelection parameter is wrong
- **-9** The ulTriggerCount parameter is wrong
- **-10** Timeout while setting the value
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 47 | 0x2F00 | 0x002F |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 35100 | 0x1C89 | 0x891C |
| word count | 2 | 16-bit integer | 20 | 0x1400 | 0x0014 |
| byte count | 1 | 8-bit integer | 40 | 0x28 | 0x28 |
| ulChannel | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulOutputType | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulPolarity | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulTriggerMask | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulTriggerEdgeSelection | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulTriggerCount | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulValue | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulOption01 | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |

MODBUS interface description

| | | | | | |
|------------|---|----------------|---------------------------|------------|------------|
| ulOption02 | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulOption03 | 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 | 35100 | 0x1C89 | 0x891C |
| word count | 2 | 16-bit integer | 20 | 0x1400 | 0x0014 |

Exception frame layout

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

MODBUS interface description

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

Function MSXE312x__AnalogOutputTriggerOutputs

Description

Trigger the analog outputs

Set the voltage of the outputs to the last value written with AnalogOutputWrite1Value.

Only useful if you called AnalogOutputWrite1Value with the bit 2 of the trigger mask set to 1 (software trigger)

All the output values will be updated at the same time

Parameters

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

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** Timeout while triggering outputs
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|---------------------------|--------------|--------------|
| unit identifier | | 8-bit integer | | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 35200 | 0x8089 | 0x8980 |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |
| byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| ulOption01 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 35200 | 0x8089 | 0x8980 |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |

Exception frame layout

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

MODBUS interface description

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

Function MSXE312x__DigitalIOSetPortDirections

Description

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

Parameters

- [Query frame layout] **ulPort** Selected digital I/O port (0 to MSXE312x__DigitalIOGetNumberOfPorts).
- [Query frame layout] **ulDirection** Digital I/O direction. Each bit indicates if the channel is used as an input or an output. (0: input, 1: output)
- [Query frame layout] **ulOption1** Reserved. Set to 0

Returns

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

- **0** The remote function performed OK
- **-2** The ulPort parameter is wrong
- **-3** The ulDirection parameter is wrong
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|---------------------------|--------------|--------------|
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 19 | 0x1300 | 0x0013 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26500 | 0x8467 | 0x6784 |
| word count | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| byte count | 1 | 8-bit integer | 12 | 0x0C | 0x0C |
| ulPort | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulDirection | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulOption1 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26500 | 0x8467 | 0x6784 |

MODBUS interface description

| | | | | | |
|------------|---|----------------|---|--------|--------|
| 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

MSXE312x__DigitalIOSetInputsFilterTime

Description

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

Parameters

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

Returns

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

- **0** The remote function performed OK
- **-2** The ulFilterTime parameter is wrong

MODBUS interface description

· **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 15 | 0x0F00 | 0x000F |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26550 | 0xB667 | 0x67B6 |
| word count | 2 | 16-bit integer | 4 | 0x0400 | 0x0004 |
| byte count | 1 | 8-bit integer | 8 | 0x08 | 0x08 |
| ulFilterTime | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulOption1 | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |

Response frame layout

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

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|--------|--------------|--------------|
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26550 | 0xB667 | 0x67B6 |
| word count | 2 | 16-bit integer | 4 | 0x0400 | 0x0004 |

Exception frame layout

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

MSXE312x__DigitalIOEnableDisableInputsFilter

Description

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

Parameters

- [Query frame layout] **ulPort** Selected digital I/O port (0 to MSXE312x__DigitalIOGetNumberOfPorts).
- [Query frame layout] **ulFilter** Digital input filter selection. Each bit indicates if the filter is enabled on the input. (0: disabled, 1: enabled)
- [Query frame layout] **ulOption1** Reserved. Set to 0

Returns

MODBUS interface description

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

- **0** The remote function performed OK
- **-2** The ulPort parameter is wrong
- **-3** The ulFilterTime parameter is wrong
- **-4** Any selected input is not an input or a bidirectional channel
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 19 | 0x1300 | 0x0013 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26600 | 0xE867 | 0x67E8 |
| word count | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| byte count | 1 | 8-bit integer | 12 | 0x0C | 0x0C |
| ulPort | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulFilter | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulOption1 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|-------|--------------|------|-------|-----------------------|-----------------------|
| | 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 | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26600 | 0xE867 | 0x67E8 |
| word count | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |

Exception frame layout

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

Function

MSXE312x__DigitalIORearmOutputsShortCircuit

Description

Rearm the digital outputs short circuit.

Please use only this function if you detected a short circuit using the function MSXE312x__DigitalIOTestOutputsShortCircuit.

Parameters

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

Returns

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

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

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit 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 | 26650 | 0x1A68 | 0x681A |
| word count | 2 | 16-bit integer | 2 | 0x0200 | 0x0002 |
| byte count | 1 | 8-bit integer | 4 | 0x04 | 0x04 |
| ulOption1 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26650 | 0x1A68 | 0x681A |
| word count | 2 | 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 MSXE312x__DigitalIOWriteChannel

Description

Set the selected digital output channel to on or off.

Parameters

- [Query frame layout] **ulChannel** Selected digital I/O channel (0 to MSXE312x__DigitalIOGetNumberOfChannels).
- [Query frame layout] **ulState** Digital I/O channel state (0: off, 1: on)
- [Query frame layout] **ulOption1** Reserved. Set to 0

Returns

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

- **0** The remote function performed OK
- **-2** The ulChannel parameter is wrong
- **-3** The ulState parameter is wrong
- **-4** The selected digital I/O is not an output
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 19 | 0x1300 | 0x0013 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26700 | 0x4C68 | 0x684C |
| word count | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| byte count | 1 | 8-bit integer | 12 | 0x0C | 0x0C |
| ulChannel | 4 | | | 0x??????? | 0x??????? |

MODBUS interface description

| | | | | | |
|-----------|---|----------------|---------------------------|------------|------------|
| | | 32-bit integer | See the description above | | |
| ulState | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |
| ulOption1 | 4 | 32-bit integer | See the description above | 0x???????? | 0x???????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26700 | 0x4C68 | 0x684C |
| word count | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |

Exception frame layout

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

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

Function MSXE312x__DigitalIOWritePort

Description

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

Parameters

- [Query frame layout] **ulPort** Selected digital I/O port (0 to MSXE312x__DigitalIOGetNumberOfPorts).
- [Query frame layout] **ulState** Digital I/O state. Each bit set the state for one digital I/O channel (0: off, 1: on)
- [Query frame layout] **ulOption1** Reserved. Set to 0

Returns

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

- **0** The remote function performed OK
- **-2** The ulPort parameter is wrong
- **-3** The ulState parameter is wrong
- **-4** Any digital I/O set to 1 is not an output channel
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 19 | 0x1300 | 0x0013 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| | 1 | | 0x10 | 0x10 | 0x10 |

MODBUS interface description

| | | | | | |
|------------------------------|---|----------------|---------------------------|-----------|-----------|
| MODBUS Function code | | 8-bit integer | | | |
| Reference number (=register) | 2 | 16-bit integer | 26750 | 0x7E68 | 0x687E |
| word count | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| byte count | 1 | 8-bit integer | 12 | 0x0C | 0x0C |
| ulPort | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulState | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |
| ulOption1 | 4 | 32-bit integer | See the description above | 0x??????? | 0x??????? |

Response frame layout

| Field | Size (Bytes) | Type | Value | little endian (Intel) | big endian (Motorola) |
|------------------------------|--------------|----------------|---|-----------------------|-----------------------|
| transaction identifier | 2 | 16-bit integer | User defined - copied by server - usually 0 | 0x0000 | 0x0000 |
| protocol identifier | 2 | 16-bit integer | 0 | 0x0000 | 0x0000 |
| length | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |
| unit identifier | 1 | 8-bit integer | 0 or 1 | 0x00 or 0x01 | 0x00 or 0x01 |
| MODBUS Function code | 1 | 8-bit integer | 0x10 | 0x10 | 0x10 |
| Reference number (=register) | 2 | 16-bit integer | 26750 | 0x7E68 | 0x687E |
| word count | 2 | 16-bit integer | 6 | 0x0600 | 0x0006 |

Exception frame layout

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

FC23 (read/write registers) Functions

[Top](#)

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

Query frame layout

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

Response frame layout

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

Exception frame layout

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

Exception code description

[Top](#)

| Name | Value | Description |
|--|-------|---|
| MODBUS_ILLEGAL_FUNCTION | 0x1 | function code not allowable for slave |
| MODBUS_ILLEGAL_DATA_ADDRESS | 0x2 | data address query is not allowed |
| MODBUS_ILLEGAL_DATA_VALUE | 0x3 | incorrect value query data field length is incorrect |
| MODBUS_ILLEGAL_DATA_RESPONSE_LENGTH | 0x4 | the request and response would generate a response which exceeds the allowed MODBUS data length |
| MODBUS_ACKNOWLEDGE | 0x5 | specialized use in conjunction with programming |
| MODBUS_DSLAVE_DEVICE_BUSY | 0x6 | specialized use in conjunction with programming |
| MODBUS_NEGATIVE_ACKNOWLEDGE | 0x07 | specialized use in conjunction with programming |
| MODBUS_MEMORY_PARITY_ERROR | 0x08 | the extended memory failed to pass consistency check |
| MODBUS_REMOTE_EXECUTION_ERROR | 0x09 | the remote function performed in error (use function GetLastCommand to know why) |
| MODBUS_GATEWAY_PATH_UNAVAILABLE | 0x0A | used with modbus gateway |
| MODBUS_GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND | 0x0B | used with modbus gateway |

Siemens Step 7 compatibility information (AWL/SDF code)

[Top](#)

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

| Function/Parameter | Renamed as |
|---|--------------------------|
| MXCommon__GetModuleType | GetModuleType |
| MXCommon__GetTime | GetTime |
| MXCommon__TestCustomerID | TestCustomerID |
| MSXE312x__AnalogInputGetNumberOfChannels | AnaInpGetNbChan |
| MSXE312x__AnalogInputGetChannelsType | AnaInpGetChlsType |
| MSXE312x__AcquisitionGetNumberOfChannels | AcqGetNbChannels |
| MSXE312x__AcquisitionGetChannelsInfo | AcqGetChanInfo |
| MSXE312x__AcquisitionAutoRefreshGetValues | AcqGetAutoRefVal |
| MSXE312x__AcquisitionAutoRefreshGetBlockingValues | AcqGetAutoRefBlockVal |
| MSXE312x__AcquisitionAutoRefreshGetConfiguration | AcqAutoRefGetConfig |
| ulByTriggerNbrOfSeqToAcquire | ByTrigNbrOfSeqToAcquire |
| MSXE312x__AcquisitionSequenceGetConfiguration | AcqSequenceGetConfig |
| ulNbrMaxSequenceToTransfer | NbrMaxSequenceToTransfer |
| ulByTriggerNbrOfSeqToAcquire | ByTrigNbrOfSeqToAcquire |
| MSXE312x__AnalogOutputGetNumberOfChannels | AnaOutGetNbChan |
| MSXE312x__DigitalIOGetNumberOfChannels | DigIOGetNbChannels |
| MSXE312x__DigitalIOGetPortAvailableDirections | DigIOGetAvDir |
| MSXE312x__DigitalIOGetPortDirections | DigIOGetDir |
| MSXE312x__DigitalIOGetInputsFilterConfiguration | DigIOGetFiltCfg |
| MSXE312x__DigitalIOTestOutputsShortCircuit | DigIOTestSC |
| MSXE312x__DigitalIORReadChannel | DigIORdChl |
| MSXE312x__DigitalIORReadPort | DigIORdPort |
| MXCommon__SetHardwareTriggerFilterTime | SetHwTrigFiltTime |
| MXCommon__InitAndStartSynchroTimer | InitStartSyncTimer |
| MXCommon__StopAndReleaseSynchroTimer | StopRelSyncTimer |
| MXCommon__Reboot | Reboot |
| MXCommon__SetCustomerKey | SetCustomerKey |
| MSXE312x__SetDataCursor | SetCursor |
| MSXE312x__AnalogInputSetChannelConfiguration | AnaInpSetChannelCfg |
| MSXE312x__AnalogInputSetSamplingRate | AnaInpSetSamplingRate |
| MSXE312x__AcquisitionAutoRefreshInitAndStart | AcqAutoRefStart |
| ulByTriggerNbrOfSeqToAcquire | ByTrigNbrOfSeqToAcquire |
| MSXE312x__AcquisitionAutoRefreshStopAndRelease | AcqAutoRefStop |
| MSXE312x__AcquisitionSequenceInitAndStart | AcqSequenceStart |
| ulNbrMaxSequenceToTransfer | NbrMaxSequenceToTransfer |

MODBUS interface description

| ulByTriggerNbrOfSeqToAcquire | ByTrigNbrOfSeqToAcquire |
|--|-------------------------|
| MSXE312x__AcquisitionSequenceStopAndRelease | AcqSequenceStop |
| MSXE312x__AnalogOutputWrite1Value | AnaOutWrite1Value |
| MSXE312x__AnalogOutputTriggerOutputs | AnaOutTriggerOut |
| MSXE312x__DigitalIOSetPortDirections | DigIOSetDir |
| MSXE312x__DigitalIOSetInputsFilterTime | DigIOSetFiltTime |
| MSXE312x__DigitalIOEnableDisableInputsFilter | DigIOEnaDisFilt |
| MSXE312x__DigitalIORearmOutputsShortCircuit | DigIORearmSC |
| MSXE312x__DigitalIOWriteChannel | DigIOWrChl |
| MSXE312x__DigitalIOWritePort | DigIOWrPort |