MODBUS interface description

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

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/atonl 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.

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

Function GetLastCommandStatus

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

Description

Return the result of the last remote function call

Parameters:

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

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

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

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

Query frame layout

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

Response frame layout

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

Exception frame layout

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

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

Function GetLastCommandStatusEx

Description

Return the result of the last remote function call

Parameters:

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

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

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

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

Query frame layout

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

Response frame layout

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

Exception frame layout

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

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

Function MXCommon__GetModuleType

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

Description

Returns the type of the MSX-E Module

Parameters:

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

Query frame layout

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

Response frame layout

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

transaction identifier	2	16-bit 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
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

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

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

Response frame layout

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

Exception frame layout

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

Function MXCommon__GetTimeEx

Description

Get the time on the module

Parameters:

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

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

Query frame layout

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

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

Response frame layout

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

Exception frame layout

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

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

Function MXCommon__TestCustomerID

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

Description

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

Parameters:

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

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

Query frame layout

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

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

Response frame layout

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

Exception frame layout

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

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

Function MXCommon__TestCustomerIDEx

Description

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

Parameters:

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

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

Query frame layout

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

Response frame layout

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

Exception frame layout

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

Function MSXE321x__TemperatureGetNumberOfChannels

Description

Return the number of temperature channels.

Parameters:

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

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	19950	0xEE4D	0x4DEE
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	0x0000	0x0000

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

Exception frame layout

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

Function MSXE321x__TemperatureGetChannelSensorClass

Description

Return the type of the temperature channels.

Parameters:

[Response frame layout] ***ulType*** Array that contain the channels class (0 : RTD, 1 : TC, 2 : NTC)

- ulType [0] : Channel 0 type
- ...
- ulType [15] : Channel 15 type

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	20000	0x204E	0x4E20
word count	2	16-bit integer	32	0x2000	0x0020

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	67	0x4300	0x0043
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01

MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	64	0x40	0x40
ulType	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)
transaction identifier	2	16-bit 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 MSXE321x__TemperatureDiagnostics

Description

Temperature lines diagnostic

Parameters:

[Response frame layout] ***ulType*** Array that contain the channels class (0 : RTD, 1 : TC, 2 : NTC)

- ulStatus [0] : Channel 0 status
  - 0 : OK
  - 1 : RTD voltage line open
  - 2 : RTD voltage line short circuit
  - 3 : RTD current line open
  - 4 : RTD current line short circuit
  - 5 : TC disconnected
  - 6 : CJC disconnected (M12 connector not present)
  - 7 : RTD sensor front-end disconnected (wiring properly linked)

◊ ...
◊ ulStatus [15] : Channel 15 ulStatus

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	20200	0xE84E	0x4EE8
word count	2	16-bit integer	32	0x2000	0x0020

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	67	0x4300	0x0043
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function	1	8-bit integer	0x03	0x03	0x03

code					
Byte count	1	8-bit integer	64	0x40	0x40
ulStatus	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)
transaction identifier	2	16-bit 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 MSXE321x__TemperatureCalibrationGetCurrentStatus

Description

Return the type of the analog input channels

Parameters:

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

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

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

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

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	20400	0xB04F	0x4FB0
word count	2	16-bit integer	8	0x0800	0x0008

Response frame layout

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

		8-bit integer		0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	16	0x10	0x10
ulStatus	4	32-bit integer	See the description above	0x????????	0x????????
ulChannel	4	32-bit integer	See the description above	0x????????	0x????????
fRefValue	4	32-bit floating point	See the description above	0x????????	0x????????
ulDigitalValue	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
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 MSXE321x__AcquisitionGetNumberOfChannels

Description

Return the number of acquisition channels.

Parameters

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

Returns

Possible return value on the remote system (read them with 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 MSXE321x__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
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	15050	0xCA3A	0x3ACA
word count	2	16-bit integer	96	0x6000	0x0060

Response frame layout

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

			copied by server - usually 0		
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)
transaction identifier	2	16-bit 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 MSXE321x__AcquisitionAutoRefreshGetValues

Description

Returns the values acquired in auto refresh mode

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	15300	0xC43B	0x3BC4
word count	2	16-bit integer	38	0x2600	0x0026

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	79	0x4F00	0x004F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	76	0x4C	0x4C
ulTimeStampLow	4	32-bit integer	See the description above	0x????????	0x????????
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
	1		0x83	0x83	0x83

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

Function MSXE321x__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
· **-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
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	big endian (Motorola)

				(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 MSXE321x__AcquisitionAutoRefreshGetConfigurati

Description

Get the current auto refresh acquisition configuration.

Parameters:

- [Response frame layout] ***ulChannelMask*** Mask of the channel acquired by the auto refresh (1 bit = 1 Channel). If the value returned is 0, then the auto refresh acquisition is not initialised.
- [Response frame layout] ***ulAverageValue*** Average value
  - 1 : not used
  - max value : 255
- [Response frame layout] ***ulRefreshTimeUnit*** Refresh Time Unit
  - 0 : microsecond
  - 1 : millisecond
  - 2 : second
- [Response frame layout] ***ulRefreshTime*** Refresh Time
  - range from min 1000 to 65535 when the unit is the microsecond
  - range from min 1 to 65535 when the unit is the millisecond
  - range from min 1 to 65535 when the unit is the second
- [Response frame layout] ***ulTriggerMask*** Define the source of the trigger
  - 0 : trigger disabled
  - 1 : Enable Hardware Digital Input Trigger
  - 2 : Enable Synchro Trigger
  - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/Cosine input)
  - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)

· [Response frame layout] ***ulTriggerMode*** Define the trigger mode
  - 1 : One shot trigger
  - 2 : Sequence trigger

· [Response frame layout] ***ulHardwareTriggerEdge*** Define the edge of the hardware trigger who generates a trigger action
  - 1 : rising edge (Only if hardware trigger selected)
  - 2 : falling edge (Only if hardware trigger selected)
  - 3 : Both front (Only if hardware trigger selected)

· [Response frame layout] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
  - 1 : all trigger event start the action
  - max value : 65535

· [Response frame layout] ***ulByTriggerNbrOfSeqToAcquire*** Define the number of sequence to acquire by each trigger event
  - 0 : continuous mode
  - 0 : number of sequence : (1..0xFFFFFFFF)

· [Response frame layout] ***ulDataFormat***
  - D0 : Absolute time stamp information
    - ♦ 0: no time stamp information
    - ♦ 1: time stamp information
  - D1 : Relative time stamp information
    - ♦ 0: no time stamp information
    - ♦ 1: time stamp information
  - D2 : Auto refresh counter information
    - ♦ 0 : No auto refresh counter information
    - ♦ 1 : Auto refresh counter information
  - D3 : System status information
    - ♦ 0 : No system status information required
    - ♦ 1 : System status information required
  - D4 : Data format
    - ♦ 0: Digital value
    - ♦ 1: Analog value

· [Response frame layout] ***ulRunning*** Auto refresh acquisition running state
  - 0 : Not running
  - 1 : Auto refresh acquisition running

Returns:

Possible return value on the remote system (read them with 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	0x0000	0x0000

			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	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	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	44	0x2C	0x2C
ulChannelMask	4	32-bit integer	See the description above	0x????????	0x????????
ulAverageValue	4	32-bit integer	See the description above	0x????????	0x????????
ulRefreshTime	4	32-bit integer	See the description above	0x????????	0x????????
ulRefreshTimeUnit	4	32-bit	See the	0x????????	0x????????

		integer	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 MSXE321x__AcquisitionSequenceGetConfiguration

Description

Get the current sequence acquisition configuration.

Parameters

- [Response frame layout] ***ulNbrOfChannel*** Nbr of channel in the sequence. If 0 no sequence initialised
- [Response frame layout] ***psChannelList*** List of the channel who compose the sequence.
- [Response frame layout] ***ulAcquisitionTime*** Acquisition Time
  - range from min 1000 to 65535 when the unit is the microsecond
  - range from min 1 to 65535 when the unit is the millisecond
  - range from min 1 to 65535 when the unit is the second
- [Response frame layout] ***ulAcquisitionTimeUnit*** Acquisition Time Unit
  - 0 : us
  - 1 : ms
  - 2 : s
- [Response frame layout] ***ulNbrOfSequence*** Number of sequence to acquire
  - 0 : continuous mode
  - Superior to 0 : number of sequence
- [Response frame layout] ***ulNbrMaxSequenceToTransfer*** Max number of sequence to acquire before a data transfer. (from 1 to 65535)
- [Response frame layout] ***ulTriggerMask*** Define the source of the trigger
  - 0 : trigger disabled
  - 1 : Enable Hardware Digital Input Trigger
  - 2 : Enable Synchro Trigger
  - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/Cosine input)
  - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- [Response frame layout] ***ulTriggerMode*** Define the trigger mode
  - 1 : One shot trigger
  - 2 : Sequence trigger
- [Response frame layout] ***ulHardwareTriggerEdge*** Define the edge of the hardware trigger who generate a trigger action
  - 1 : rising front (Only if hardware trigger selected)
  - 2 : falling front (Only if hardware trigger selected)
  - 3 : Both front (Only if hardware trigger selected)
- [Response frame layout] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
  - 1 : all trigger event start the action
  - max value : 65535
- [Response frame layout] ***ulByTriggerNbrOfSeqToAcquire*** define the number of sequence to acquire by each trigger event
  - 0 : continuous mode
  - Superior to 0 : number of sequence : (1..0xFFFFFFFF)
- [Response frame layout] ***ulDataFormat*** Data format option
  - D0 : Absolute time stamp information

- ♦ 0: no time stamp information
- ♦ 1: time stamp information

• D1 : Relative time stamp information
- ♦ 0: no time stamp information
- ♦ 1: time stamp information

• D2 : Sequence counter information
- ♦ 0 : No sequence counter information
- ♦ 1 : Sequence counter information

• D3 : System status information
- ♦ 0 : No system status information required
- ♦ 1 : System status information required

• D4 : Data format
- ♦ 0: Digital value
- ♦ 1: Analog value

· [Response frame layout] ***ulRunning*** Sequence acquisition running state
- • 0 : Not running
- • 1 : Sequence acquisition running

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	15650	0x223D	0x3D22
word count	2	16-bit integer	56	0x3800	0x0038

Response frame layout

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

ulByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????	0x????????
ulDataFormat	4	32-bit integer	See the description above	0x????????	0x????????
ulRunning	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

Function MXCommon__SetHardwareTriggerFilterTime

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

Description

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

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

Parameters

· [Query frame layout] ***ulFilterTime*** Filter time for the hardware trigger input

in 250ns step (max value : 65535).

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

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

Returns

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

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

Query frame layout

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

Response frame layout

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

Exception frame layout

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

Function MXCommon__SetHardwareTriggerFilterTimeEx

Description

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

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

Parameters

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

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

Returns

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

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

Query frame layout

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

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

Response frame layout

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

Exception frame layout

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

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

Function MXCommon__InitAndStartSynchroTimer

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

Description

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

Parameters:

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

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

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

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

[Query frame layout] ***ulGenerateTriggerMode:***

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

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

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

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

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

- 3 : Both front (Only if hardware trigger selected)

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

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

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

Returns:

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

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

Query frame layout

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

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

Response frame layout

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

Exception frame layout

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

Function MXCommon__InitAndStartSynchroTimerEx

Description

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

Parameters:

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

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

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

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

[Query frame layout] ***ulGenerateTriggerMode:***

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

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

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

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

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

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

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

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

Returns:

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

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

Query frame layout

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

		16-bit integer			
byte count	1	8-bit integer	32	0x20	0x20
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulReloadValue	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrOfCycle	4	32-bit integer	See the description above	0x????????	0x????????
ulGenerateTriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????
ulOption03	4	32-bit integer	See the description above	0x????????	0x????????
ulOption04	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

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	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 occured
- -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
	1		0 or 1		

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	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
- -1: means an system error occured (probably EPERM)

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	12	0x0C00	0x000C
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	103	0x6700	0x0067
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
Dummy	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

		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	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 occured (probably EPERM)

Query frame layout

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

Response frame layout

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

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11150	0x8E2B	0x2B8E
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function MXCommon__SetCustomerKey

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

Description

Permit to set the Customer key

Parameters:

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

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

Returns:

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

- 0 : means the remote function performed OK
- -1: means an system error 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	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 -	0x0000	0x0000

			usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	104	0x6800	0x0068
word count	2	16-bit integer	24	0x1800	0x0018

Exception frame layout

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

Function MXCommon__SetCustomerKeyEx

Description

Permit to set the Customer key

Parameters:

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

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

Returns:

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

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

Query frame layout

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

Response frame layout

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

			- 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 MSXE321x__SetDataCursor

Description

Change the active data cursor cursor

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	65530	0xFAFF	0xFFFA
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

			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	0xFFFA
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 MSXE321x__TemperatureSetChannelType

Description

Temperature sensor type selection for the selected channel

Parameters:

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

[Query frame layout] ***ulType*** Type selection.

- For TC
  - 1: Type B
  - 5: Type E
  - 10: Type J
  - 11: Type K
  - 14: Type N
  - 18: Type R
  - 19: Type S
  - 20: Type T
- For RTD type PT
  - 100: PT100
  - 200: PT200
  - 500: PT500
  - 1000: PT1000
  - ...
- For RTD type Ni
  - 101: Ni100
  - 201: Ni200
  - 501: Ni500
  - 1001: Ni1000
  - ...
- For NTC
  - 1: G10K4D453
  - ...

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

Returns:

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

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

Query frame layout

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

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	20450	0xE24F	0x4FE2
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannel	4	32-bit integer	See the description above	0x????????	0x????????
ulType	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	20450	0xE24F	0x4FE2
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 MSXE321x__TemperatureSetSamplingRate

Description

Temperature acquisition sampling rate selection

Parameters:

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

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

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

- 5 for 5Hz
- 10 for 10Hz

- 20 for 20Hz
- 40 for 40Hz
- 80 for 80Hz
- 160 for 160Hz
- 320 for 320Hz
- 640 for 640Hz
- 1000 for 1000Hz
- 2000 for 2000Hz

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	20500	0x1450	0x5014
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulChannelGroup	4	32-bit integer	See the description above	0x????????	0x????????
ulBaseSamplingRate	4	32-bit integer	See the description above	0x????????	0x????????
ulOption1	4			0x????????	0x????????

		32-bit integer	See the description above		

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	20500	0x1450	0x5014
word count	2	16-bit integer	6	0x0600	0x0006

Exception frame layout

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

code		integer	corresponding chapter		

Function MSXE321x__TemperatureCalibrationStart

Description

This function start the calibration thread

Parameters:

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

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

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

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

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

Returns:

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

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

Query frame layout

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

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	83	0x5300	0x0053
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	20550	0x4650	0x5046
word count	2	16-bit integer	38	0x2600	0x0026
byte count	1	8-bit integer	76	0x4C	0x4C
ulChannelGroup	4	32-bit integer	See the description above	0x????????	0x????????
ulGainSelection	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrOfRefValues	4	32-bit integer	See the description above	0x????????	0x????????
fRefValues	64	32-bit floating point array	See the description above	0x????????[16]	0x????????[16]

Response frame layout

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

Reference number (=register)	2	16-bit integer	20550	0x4650	0x5046
word count	2	16-bit integer	38	0x2600	0x0026

Exception frame layout

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

Function MSXE321x__TemperatureCalibrationNextStep

Description

This function start the next calibration step

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit 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	20600	0x7850	0x5078
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

Function code		integer			
Reference number (=register)	2	16-bit integer	20600	0x7850	0x5078
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 MSXE321x__TemperatureCalibrationBreak

Description

This function break the current calibration

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit 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	20650	0xAA50	0x50AA
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

Function code		integer			
Reference number (=register)	2	16-bit integer	20650	0xAA50	0x50AA
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 MSXE321x__AcquisitionAutoRefreshInitAndSta

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 1000 to 65535 when the unit is the microsecond
  - ♦ range from min 1 to 65535 when the unit is the millisecond
  - ♦ range from min 1 to 65535 when the unit is the second
- [Query frame layout] ***ulTriggerMask*** Define the source of the trigger
  - ♦ 0 : trigger disabled
  - ♦ 1 : Enable Hardware Digital Input Trigger
  - ♦ 2 : Enable Synchro Trigger
  - ♦ 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/Cosine input)
  - ♦ 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- [Query frame layout] ***ulTriggerMode*** Define the trigger mode
  - ♦ 1 : One shot trigger
  - ♦ 2 : Sequence trigger
- [Query frame layout] ***ulHardwareTriggerEdge*** Define the edge of the hardware trigger who generates a trigger action
  - ♦ 1 : rising edge (Only if hardware trigger selected)
  - ♦ 2 : falling edge (Only if hardware trigger selected)
  - ♦ 3 : Both front (Only if hardware trigger selected)
- [Query frame layout] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
  - ♦ 1 : all trigger event start the action
  - ♦ max value : 65535
- [Query frame layout] ***ulByTriggerNbrOfSeqToAcquire*** Define the number of sequence to acquire by each trigger event
  - ♦ 0 : continuous mode
  - ♦ 0 : number of sequence : (1..0xFFFFFFFF)
- [Query frame layout] ***ulDataFormat***
  - ♦ D0 : Absolute time stamp information
    - ◊ 0: no time stamp information
    - ◊ 1: time stamp information
  - ♦ D1 : Relative time stamp information
    - ◊ 0: no time stamp information
    - ◊ 1: time stamp information
  - ♦ D2 : Auto refresh counter information
    - ◊ 0 : No auto refresh counter information
    - ◊ 1 : Auto refresh counter information
  - ♦ D3 : System status information
    - ◊ 0 : No system status information required
    - ◊ 1 : System status information required
  - ♦ D4 : Data format
    - ◊ 0: Digital value
    - ◊ 1: Analog value
- [Query frame layout] ***ulForceStart***
  - ♦ 0 : Function return a error if any acquisition already in progress
  - ♦ 1 : If any acquisition in progress then stop this
- [Query frame layout] ***ulOption1*** Reserved. Set to 0

- [Query frame layout] ***ulOption2*** Reserved. Set to 0
- [Query frame layout] ***ulOption3*** Reserved. Set to 0

Returns:

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

- **0** The remote function performed OK
- **-2** Any acquisition already in progress
- **-3** Any selected channel not OK, call the diagnostic function for more information
- **-4** Channel Mask error
- **-5** Not available average value
- **-6** Not available refresh time unit
- **-7** The minimal refresh time is 1000 us
- **-8** The maximal refresh time is 65535
- **-9** Trigger mask not available
- **-10** Trigger mask : 2 different trigger source cannot be simultaneously be activated
- **-11** Trigger mode not available
- **-12** Trigger mask : 2 trigger mode cannot be simultaneously activated
- **-13** Hardware trigger : front definition error
- **-14** Hardware trigger count value not available
- **-15** Nbr of sequence to acquire by trigger mode not available
- **-16** Data format not available
- **-17** Selected channels combination not available
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	63	0x3F00	0x003F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
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????????
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 MSXE321x__AcquisitionAutoRefreshStopAndR

Description

Stops the current auto refresh acquisition

Parameters

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	15850	0xEA3D	0x3DEA
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption1	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

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	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 MSXE321x__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 1000 to 65535 when the unit is the microsecond
    - range from min 1 to 65535 when the unit is the millisecond
    - range from min 1 to 65535 when the unit is the second
- [Query frame layout] ***ulAcquisitionTimeUnit*** Acquisition Time Unit
    - 0 : us
    - 1 : ms
    - 2 : s
- [Query frame layout] ***ulNbrOfSequence*** Number of sequence to acquire
    - 0 : continuous mode
    - Superior to 0 : number of sequence
- [Query frame layout] ***ulNbrMaxSequenceToTransfer*** Max number of sequence to acquire before a data transfer. (from 1 to 65535)
- [Query frame layout] ***ulTriggerMask*** Define the source of the trigger
    - 0 : trigger disabled
    - 1 : Enable Hardware Digital Input Trigger
    - 2 : Enable Synchro Trigger
    - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/Cosine input)
    - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- [Query frame layout] ***ulTriggerMode*** Define the trigger mode
    - 1 : One shot trigger
    - 2 : Sequence trigger
- [Query frame layout] ***ulHardwareTriggerEdge*** Define the edge of the hardware trigger who generate a trigger action
    - 1 : rising front (Only if hardware trigger selected)
    - 2 : falling front (Only if hardware trigger selected)
    - 3 : Both front (Only if hardware trigger selected)
- [Query frame layout] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
    - 1 : all trigger event start the action
    - max value : 65535
- [Query frame layout] ***ulByTriggerNbrOfSeqToAcquire*** define the number of sequence to acquire by each trigger event

- ♦ 0 : continuous mode
- ♦ Superior to 0 : number of sequence : (1..0xFFFFFFFF)

• [Query frame layout] ***ulDataFormat*** Data format option
- ♦ D0 : Absolute time stamp information
  - ◊ 0: no time stamp information
  - ◊ 1: time stamp information
- ♦ D1 : Relative time stamp information
  - ◊ 0: no time stamp information
  - ◊ 1: time stamp information
- ♦ D2 : Sequence counter information
  - ◊ 0 : No sequence counter information
  - ◊ 1 : Sequence counter information
- ♦ D3 : System status information
  - ◊ 0 : No system status information required
  - ◊ 1 : System status information required
- ♦ D4 : Data format
  - ◊ 0: Digital value
  - ◊ 1: Analog value

• [Query frame layout] ***ulForceStart***
- ♦ 0 : Function return a error if any acquisition already in progress
- ♦ 1 : If any acquisition in progress then stop this

• [Query frame layout] ***ulOption1*** Reserved. Set to 0
• [Query frame layout] ***ulOption2*** Reserved. Set to 0
• [Query frame layout] ***ulOption3*** Reserved. Set to 0

Returns:

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

- **0** The remote function performed OK
- **-2** Any acquisition already in progress
- **-3** The number of channel in the sequence is null or to high
- **-4** Channel index selection error
- **-5** Channel already selected
- **-6** Any selected channel not OK, call the diagnostic function for more information
- **-7** Not available acquisition time unit
- **-8** The minimal acquisition time is 1000 us !
- **-9** The maximal acquisition time is 65535 !
- **-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 e (Mot
transaction identifier	2	16-bit 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 o
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	15900	0x1C3E	0x3E1
word count	2	16-bit integer	62	0x3E00	0x003
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 array	See the description above	0x????????[16]	0x????
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			0x????????	0x????

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

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
	2		15900	0x1C3E	0x3E1C

Reference number (=register)		16-bit integer			
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 MSXE321x__AcquisitionSequenceStopAndRele

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)
transaction identifier	2	16-bit 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

Function code		integer			
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??

FC23 (read/write registers) Functions

Top

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

Query frame layout

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

Response frame layout

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

Exception frame layout

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

Exception code description

Top

Name	Value	
MODBUS_ILLEGAL_FUNCTION	0x1	func allov slave
MODBUS_ILLEGAL_DATA_ADDRESS	0x2	data quer
MODBUS_ILLEGAL_DATA_VALUE	0x3	incor quer leng
MODBUS_ILLEGAL_DATA_RESPONSE_LENGTH	0x4	the r woul respo exce MOI
MODBUS_ACKNOWLEDGE	0x5	spec conj prog
MODBUS_DSLAVE_DEVICE_BUSY	0x6	spec conj prog
MODBUS_NEGATIVE_ACKNOWLEDGE	0x07	spec conj prog
MODBUS_MEMORY_PARITY_ERROR	0x08	the e faile cons
MODBUS_REMOTE_EXECUTION_ERROR	0x09	the r perfo (use GetL to kn
MODBUS_GATEWAY_PATH_UNAVAILABLE	0x0A	used gatev
MODBUS_GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND	0x0B	used gatev

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
MSXE321x__TemperatureGetNumberOfChannels	TempGetNbChan
MSXE321x__TemperatureGetChannelSensorClass	TempGetSensorClass
MSXE321x__TemperatureDiagnostics	TempDiagnostics
MSXE321x__TemperatureCalibrationGetCurrentStatus	TempCalibStatus
MSXE321x__AcquisitionGetNumberOfChannels	AcqGetNbChannels
MSXE321x__AcquisitionGetChannelsInfo	AcqGetChanInfo
MSXE321x__AcquisitionAutoRefreshGetValues	AcqGetAutoRefVal
MSXE321x__AcquisitionAutoRefreshGetBlockingValues	AcqGetAutoRefBlockVal
MSXE321x__AcquisitionAutoRefreshGetConfiguration	AcqAutoRefGetConfig
ulByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSeqToAcq
MSXE321x__AcquisitionSequenceGetConfiguration	AcqSequenceGetConfig
ulNbrMaxSequenceToTransfer	NbrMaxSequenceToTra
ulByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSeqToAcq
MXCommon__SetHardwareTriggerFilterTime	SetHwTrigFiltTime
MXCommon__InitAndStartSynchroTimer	InitStartSyncTimer
MXCommon__StopAndReleaseSynchroTimer	StopRelSyncTimer
MXCommon__Reboot	Reboot
MXCommon__SetCustomerKey	SetCustomerKey
MSXE321x__SetDataCursor	SetCursor
MSXE321x__TemperatureSetChannelType	TempSetChannelType
MSXE321x__TemperatureSetSamplingRate	TempSetSamplingRate
MSXE321x__TemperatureCalibrationStart	TempCalibStart
MSXE321x__TemperatureCalibrationNextStep	TempCalibNext
MSXE321x__TemperatureCalibrationBreak	TempCalibBreak
MSXE321x__AcquisitionAutoRefreshInitAndStart	AcqAutoRefStart
ulByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSeqToAcq
MSXE321x__AcquisitionAutoRefreshStopAndRelease	AcqAutoRefStop
MSXE321x__AcquisitionSequenceInitAndStart	AcqSequenceStart
ulNbrMaxSequenceToTransfer	NbrMaxSequenceToTra
ulByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSeqToAcq
MSXE321x__AcquisitionSequenceStopAndRelease	AcqSequenceStop