MODBUS interface description

Table of Contents

Table of Contents

Table of Contents

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.

- GetLastCommandStatus — Register: **0**
- GetLastCommandStatusEx — Register: **10000**
- MXCommon__GetModuleType — Register: **1**
- MXCommon__GetModuleTypeEx — Register: **10200**
- MXCommon__GetTime — Register: **2**
- MXCommon__GetTimeEx — Register: **10500**
- MXCommon__TestCustomerID — Register: **3**
- MXCommon__TestCustomerIDEx — Register: **10550**
- MSXE17xx__DigitalIOReadAllChannelsValue — Register: **7000**
- MSXE17xx__DigitalIOTestShortCircuit — Register: **7050**
- MSXE17xx__IOWatchdogGetStatusAndValue — Register: **8000**
- MSXE173x__EndatGetPosition0 — Register: **3100**
- MSXE173x__EndatGetPosition1 — Register: **3200**
- MSXE173x__EndatGetPosition2 — Register: **3300**
- MSXE173x__EndatGetPosition3 — Register: **3400**
- MSXE173x__EndatGetSensorProperties0 — Register: **3500**
- MSXE173x__EndatGetSensorProperties1 — Register: **3600**
- MSXE173x__EndatGetSensorProperties2 — Register: **3700**
- MSXE173x__EndatGetSensorProperties3 — Register: **3800**
- MSXE173x__EndatGetPositionWithAddData0 — Register: **3900**

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			0x??[100]	0x??[100]

		8-bit integer array	See the description above		

Exception frame layout

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

Function 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)
	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	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			0x??[100]	0x??[100]

		8-bit integer array	See the description above		

Exception frame layout

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

Function MXCommon__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 -	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	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 -	0x0000	0x0000

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

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

Response frame layout

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

			copied by server - usually 0		
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
	1		0x03	0x03	0x03

MODBUS Function code		8-bit integer			
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
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	3	0x0300	0x0003
word count	2	16-bit integer	16	0x1000	0x0010

Response frame layout

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

		array	above		
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 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	0x0000	0x0000

			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	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 MSXE17xx__DigitalIOReadAllChannelsValue

Description

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

Parameters:

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

Returns:

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

Query frame layout

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

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

Response frame layout

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

Exception frame layout

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

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

Function MSXE17xx__DigitalIOTestShortCircuit

Description

Parameters:

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

- ♦ 0 : no short circuit
- ♦ 1 : short circuit

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
	2		7050	0x8A1B	0x1B8A

Reference number (=register)		16-bit integer			
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Exception code		8-bit integer	See corresponding chapter		

Function MSXE17xx__IOWatchdogGetStatusAndValue

Description

Get watchdog current status and value information

Parameters:

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

- BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXX0: is stopped
- BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXX1: is running
- BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXX0X: is not run down
- BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXX1X: is run down

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number	2	16-bit integer	8000	0x401F	0x1F40

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

Response frame layout

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

Exception frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
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 MSXE173x__EndatGetPosition0

Description

Reads the position of the sensor 0.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Response frame layout]***ulPositionLow*** Position - low bits (LSB)
- [Response frame layout]***ulPositionHigh*** Position - high bits (MSB)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reading the position
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

		16-bit integer			
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	3100	0x1C0C	0x0C1C
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
ulPositionLow	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionHigh	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

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

Function MSXE173x__EndatGetPosition1

Description

Reads the position of the sensor 1.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Response frame layout]***ulPositionLow*** Position - low bits (LSB)
- [Response frame layout]***ulPositionHigh*** Position - high bits (MSB)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reading the position
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

			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	3200	0x800C	0x0C80
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
ulPositionLow	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionHigh	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 MSXE173x__EndatGetPosition2

Description

Reads the position of the sensor 2.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Response frame layout]***ulPositionLow*** Position - low bits (LSB)
- [Response frame layout]***ulPositionHigh*** Position - high bits (MSB)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reading the position
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	3300	0xE40C	0x0CE4
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
ulPositionLow	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionHigh	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 MSXE173x__EndatGetPosition3

Description

Reads the position of the sensor 3.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Response frame layout]***ulPositionLow*** Position - low bits (LSB)
- [Response frame layout]***ulPositionHigh*** Position - high bits (MSB)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reading the position
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	3400	0x480D	0x0D48
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
ulPositionLow	4	32-bit integer	See the description	0x????????	0x????????

			above		
ulPositionHigh	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 MSXE173x__EndatGetSensorProperties0

Description

Reads the properties of the sensor 0.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Response frame layout]***ulIDNumberLsb*** ID Number - low bits (see page 67/121 of EnDat specifications)
  - ♦ **bits 0 -> 15** Word 24 (last 2 digits of the ID number in ASCII format)
  - ♦ **bits 16 -> 31** Word 25 (first part of the first 6 digits of the ID number)
- [Response frame layout]***ulIDNumberMsb*** ID Number - high bits (see page 67/121 of EnDat specifications)
  - ♦ **bits 0 -> 15** Word 26 (last part of the first 6 digits of the ID number)
- [Response frame layout]***ulSerialNumberLsb*** Serialnumber - low bits (see page 68/121 of EnDat specifications)
  - ♦ **bits 0 -> 15** Word 27
    - ◊ **bits 0 -> 7** ASCII digit
    - ◊ **bits 8 -> 15** binary coded
  - ♦ **bits 16 -> 31** Word 28 binary coded

- [Response frame layout]***ulSerialNumberMsb*** Serialnumber - high bits (see page 68/121 of EnDat specifications)
    - **bits 0 -> 15** Word 29
        - **bits 0 -> 7** binary coded
        - **bits 8 -> 15** ASCII digit

  [Response frame layout]***ulModel*** Model/Type of the sensor (see page 79/84 of EnDat application notes 722024)
    - **0, 1, 2, 3** Incremental linear encoder
    - **4, 6** Absolute linear encoder
    - **8, 9, 10, 11** Incremental rotary encoder or angle encoder
    - **12** Single-turn encoder
    - **13, 14** Multi-turn encoder
- [Response frame layout]***ulMode*** Provide if sensor allow to use EnDat 2.2 commands
    - **0** Sensor does not support EnDat 2.2. commands
    - **1** Sensor supports EnDat 2.2 commands
- [Response frame layout]***ulPositionSize*** Size of the position value send by the sensor (in bits)
- [Response frame layout]***ulSignalPeriod*** Signal period length or signal periods per revolution for incremental output signals (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulStepPerRevolution*** Measuring step length or measuring steps per revolution with serial data transfer (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulNumberOfRevolution*** Distinguishable revolutions - only for multiturn encoders (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulScalingFactor*** Scaling factor for resolution (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulAdditionalData*** Supported additional data (see page 85/121 of EnDat specifications)
    - **Bit 0** "Position value 2" is available (MRS-Code: 0x42, 0x43, 0x44)
    - **Bit 1** "Test values" is available (MRS-Code: 0x49, 0x4A, 0x4B)
    - **Bit 2** "Temperature sensor 1 (external)" is available (MRS-Code: 0x4C)
    - **Bit 3** "Temperature sensor 2 (external)" is available (MRS-Code: 0x4D)
    - **Bit 4** "Additional sensors" is available (MRS-Code: 0x4E)
    - **Bit 16** "Commutation" is available (MRS-Code: 0x51)
    - **Bit 17** "Acceleration" is available (MRS-Code: 0x52)
    - **Bit 18** "Limit position signals" is available (MRS-Code: 0x54)
    - **Bit 19** "Asynchronous position value" is available (MRS-Code: 0x56, 0x57, 0x58)
    - **Bit 20** "Operating status error sources" is available (MRS-Code: 0x59)
    - **Bit 23** "Timestamp" is available (MRS-Code: 0x5B)

Example with ROQ 437 2048 5XS08-C4 Sensor

- **Serial Number**

    - **Sensor Serial number** = 35549793 in hex -> 0x**20**021e7261**20**
        - In bold, ASCII character. In this case spaces (0x20)
        - 0x0201e7261 = 35549793 in decimal
    - **ulSerialNumberLsb** = 0x1e726120
    - **ulSerialNumberMsb** = 0x00002002

- **ID**

    - **ID number** = 693 641-05 in hex -> 0xA6E79**3035**

    - ♦ In bold, ASCII character. In this case (0 (0x30) and 5 (0x35))
    - ♦ 0xA6E79 = 683641 in decimal
- **ulIDNumberLsb** = 0x6E793035
- **ulIDNumberMsb** = 0xA

Example with EQN 1135 512 Sensor

- **Serial Number Example**

    - **Sensor Serial number** = 38473098A in hex -> 0x**20**024B0D8A**41**
        - ♦ In bold, ASCII character. In this case (0x20 and 0x41 (A))
        - ♦ 0x024B0D8A = 38473098 in decimal
    - **ulSerialNumberLsb** = 0x4B0D8A41
    - **ulSerialNumberMsb** = 0x00002002

- **ID**

    - **ID number** = 743 587-01 in hex -> 0xB58A3**3031**
        - ♦ In bold, ASCII character. In this case (0 (0x30) and 1 (0x31))
        - ♦ 0xB58A3 = 743 587 in decimal
    - **ulIDNumberLsb** = 0x58A33031
    - **ulIDNumberMsb** = 0xB

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while selecting memory area 0xA3
- **-8** Error while reading ID number (address 0x8)
- **-9** Error while reading ID number (address 0x9)
- **-10** Error while reading ID number (address 0xA)
- **-11** Error while reading Serialnumber (address 0xB)
- **-12** Error while reading Serialnumber (address 0xC)
- **-13** Error while reading Serialnumber (address 0xD)
- **-14** Error while selecting memory area 0xA1
- **-15** Error while reading encoder model (address 0xE)
- **-16** Error while reading signal period length or signal periods per revolution for incremental output signals (address 0xF)
- **-17** Error while getting the position
- **-18** Error while selecting memory area 0xA3
- **-19** Error while reading signal period length or signal periods per revolution for incremental output signals (address 0x0)
- **-20** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)

- **-21** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-22** Error while selecting memory area 0xBD
- **-23** Error while reading scaling factor for resolution (address 0x1B)
- **-24** Error while reading measuring step, or measuring steps per revolution or subdivision values of a grating period (address 0x1C)
- **-25** Error while reading measuring step, or measuring steps per revolution or subdivision values of a grating period (address 0x1D)
- **-26** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-27** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-28** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-29** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-30** Error while reading distinguishable revolutions (address 0x1)
- **-31** Error while selecting memory area 0xBD
- **-32** Error while reading number of distinguishable revolutions with scaling factor (address 0x22)
- **-33** Error while selecting memory area 0xBD
- **-34** Error while reading status of additional datum 1 (address 0x0)
- **-35** Error while reading status of additional datum 2 (address 0x1)
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	3500	0xAC0D	0x0DAC
word count	2		24	0x1800	0x0018

		16-bit integer			

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	51	0x3300	0x0033
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	48	0x30	0x30
ulIDNumberLsb	4	32-bit integer	See the description above	0x????????	0x????????
ulIDNumberMsb	4	32-bit integer	See the description above	0x????????	0x????????
ulSerialNumberLsb	4	32-bit integer	See the description above	0x????????	0x????????
ulSerialNumberMsb	4	32-bit integer	See the description above	0x????????	0x????????
ulModel	4	32-bit integer	See the description above	0x????????	0x????????
ulMode	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionSize	4	32-bit integer	See the description above	0x????????	0x????????
ulSignalPeriod	4	32-bit integer	See the description above	0x????????	0x????????
ulStepPerRevolution	4	32-bit integer	See the description above	0x????????	0x????????

ulNumberOfRevolution	4	32-bit integer	See the description above	0x????????	0x????????
ulScalingFactor	4	32-bit integer	See the description above	0x????????	0x????????
ulAdditionalData	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 MSXE173x__EndatGetSensorProperties1

Description

Reads the properties of the sensor 1.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Response frame layout]***ullDNumberLsb*** ID Number - low bits (see page 67/121 of EnDat specifications)
  - **bits 0 -> 15** Word 24 (last 2 digits of the ID number in ASCII format)
  - **bits 16 -> 31** Word 25 (first part of the first 6 digits of the ID number)
- [Response frame layout]***ullDNumberMsb*** ID Number - high bits (see page 67/121 of EnDat specifications)
  - **bits 0 -> 15** Word 26 (last part of the first 6 digits of the ID number)

- [Response frame layout]***ulSerialNumberLsb*** Serialnumber - low bits (see page 68/121 of EnDat specifications)
    - **bits 0 -> 15** Word 27
        - **bits 0 -> 7** ASCII digit
        - **bits 8 -> 15** binary coded
    - **bits 16 -> 31** Word 28 binary coded
- [Response frame layout]***ulSerialNumberMsb*** Serialnumber - high bits (see page 68/121 of EnDat specifications)
    - **bits 0 -> 15** Word 29
        - **bits 0 -> 7** binary coded
        - **bits 8 -> 15** ASCII digit
- [Response frame layout]***ulModel*** Model/Type of the sensor (see page 79/84 of EnDat application notes 722024)
    - **0, 1, 2, 3** Incremental linear encoder
    - **4, 6** Absolute linear encoder
    - **8, 9, 10, 11** Incremental rotary encoder or angle encoder
    - **12** Single-turn encoder
    - **13, 14** Multi-turn encoder
- [Response frame layout]***ulMode*** Provide if sensor allow to use EnDat 2.2 commands
    - **0** Sensor does not support EnDat 2.2. commands
    - **1** Sensor supports EnDat 2.2 commands
- [Response frame layout]***ulPositionSize*** Size of the position value send by the sensor (in bits)
- [Response frame layout]***ulSignalPeriod*** Signal period length or signal periods per revolution for incremental output signals (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulStepPerRevolution*** Measuring step length or measuring steps per revolution with serial data transfer (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulNumberOfRevolution*** Distinguishable revolutions - only for multiturn encoders (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulScalingFactor*** Scaling factor for resolution (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulAdditionalData*** Supported additional data (see page 85/121 of EnDat specifications)
    - **Bit 0** "Position value 2" is available (MRS-Code: 0x42, 0x43, 0x44)
    - **Bit 1** "Test values" is available (MRS-Code: 0x49, 0x4A, 0x4B)
    - **Bit 2** "Temperature sensor 1 (external)" is available (MRS-Code: 0x4C)
    - **Bit 3** "Temperature sensor 2 (external)" is available (MRS-Code: 0x4D)
    - **Bit 4** "Additional sensors" is available (MRS-Code: 0x4E)
    - **Bit 16** "Commutation" is available (MRS-Code: 0x51)
    - **Bit 17** "Acceleration" is available (MRS-Code: 0x52)
    - **Bit 18** "Limit position signals" is available (MRS-Code: 0x54)
    - **Bit 19** "Asynchronous position value" is available (MRS-Code: 0x56, 0x57, 0x58)
    - **Bit 20** "Operating status error sources" is available (MRS-Code: 0x59)
    - **Bit 23** "Timestamp" is available (MRS-Code: 0x5B)

Example with ROQ 437 2048 5XS08-C4 Sensor

- **Serial Number**

- **Sensor Serial number** = 35549793 in hex -> 0x**20**021e7261**20**
    - In bold, ASCII character. In this case spaces (0x20)
    - 0x0201e7261 = 35549793 in decimal

- **ulSerialNumberLsb** = 0x1e726120
- **ulSerialNumberMsb** = 0x00002002

- **ID**

  - **ID number** = 693 641-05 in hex -> 0xA6E79**3035**
    - ♦ In bold, ASCII character. In this case (0 (0x30) and 5 (0x35))
    - ♦ 0xA6E79 = 683641 in decimal
  - **ulIDNumberLsb** = 0x6E793035
  - **ulIDNumberMsb** = 0xA

Example with EQN 1135 512 Sensor

- **Serial Number Example**

  - **Sensor Serial number** = 38473098A in hex -> 0x**20**024B0D8A**41**
    - ♦ In bold, ASCII character. In this case (0x20 and 0x41 (A))
    - ♦ 0x024B0D8A = 38473098 in decimal
  - **ulSerialNumberLsb** = 0x4B0D8A41
  - **ulSerialNumberMsb** = 0x00002002

- **ID**

  - **ID number** = 743 587-01 in hex -> 0xB58A3**3031**
    - ♦ In bold, ASCII character. In this case (0 (0x30) and 1 (0x31))
    - ♦ 0xB58A3 = 743 587 in decimal
  - **ulIDNumberLsb** = 0x58A33031
  - **ulIDNumberMsb** = 0xB

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while selecting memory area 0xA3
- **-8** Error while reading ID number (address 0x8)
- **-9** Error while reading ID number (address 0x9)
- **-10** Error while reading ID number (address 0xA)
- **-11** Error while reading Serialnumber (address 0xB)
- **-12** Error while reading Serialnumber (address 0xC)
- **-13** Error while reading Serialnumber (address 0xD)
- **-14** Error while selecting memory area 0xA1
- **-15** Error while reading encoder model (address 0xE)
- **-16** Error while reading signal period length or signal periods per revolution for incremental output signals (address 0xF)
- **-17** Error while getting the position

- **-18** Error while selecting memory area 0xA3
- **-19** Error while reading signal period length or signal periods per revolution for incremental output signals (address 0x0)
- **-20** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-21** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-22** Error while selecting memory area 0xBD
- **-23** Error while reading scaling factor for resolution (address 0x1B)
- **-24** Error while reading measuring step, or measuring steps per revolution or subdivision values of a grating period (address 0x1C)
- **-25** Error while reading measuring step, or measuring steps per revolution or subdivision values of a grating period (address 0x1D)
- **-26** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-27** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-28** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-29** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-30** Error while reading distinguishable revolutions (address 0x1)
- **-31** Error while selecting memory area 0xBD
- **-32** Error while reading number of distinguishable revolutions with scaling factor (address 0x22)
- **-33** Error while selecting memory area 0xBD
- **-34** Error while reading status of additional datum 1 (address 0x0)
- **-35** Error while reading status of additional datum 2 (address 0x1)
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	1	8-bit integer	0x03	0x03	0x03

code					
Reference number (=register)	2	16-bit integer	3600	0x100E	0x0E10
word count	2	16-bit integer	24	0x1800	0x0018

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	51	0x3300	0x0033
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	48	0x30	0x30
ulIDNumberLsb	4	32-bit integer	See the description above	0x????????	0x????????
ulIDNumberMsb	4	32-bit integer	See the description above	0x????????	0x????????
ulSerialNumberLsb	4	32-bit integer	See the description above	0x????????	0x????????
ulSerialNumberMsb	4	32-bit integer	See the description above	0x????????	0x????????
ulModel	4	32-bit integer	See the description above	0x????????	0x????????
ulMode	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionSize	4	32-bit integer	See the description above	0x????????	0x????????
ulSignalPeriod	4	32-bit	See the	0x????????	0x????????

		integer	description above		
ulStepPerRevolution	4	32-bit integer	See the description above	0x????????	0x????????
ulNumberOfRevolution	4	32-bit integer	See the description above	0x????????	0x????????
ulScalingFactor	4	32-bit integer	See the description above	0x????????	0x????????
ulAdditionalData	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 MSXE173x__EndatGetSensorProperties2

Description

Reads the properties of the sensor 2.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Response frame layout]***ullDNumberLsb*** ID Number - low bits (see page 67/121 of EnDat specifications)

    - ♦ **bits 0 -> 15** Word 24 (last 2 digits of the ID number in ASCII format)
    - ♦ **bits 16 -> 31** Word 25 (first part of the first 6 digits of the ID number)
- [Response frame layout]***ulIDNumberMsb*** ID Number - high bits (see page 67/121 of EnDat specifications)
    - ♦ **bits 0 -> 15** Word 26 (last part of the first 6 digits of the ID number)
- [Response frame layout]***ulSerialNumberLsb*** Serialnumber - low bits (see page 68/121 of EnDat specifications)
    - ♦ **bits 0 -> 15** Word 27
        - ◊ **bits 0 -> 7** ASCII digit
        - ◊ **bits 8 -> 15** binary coded
    - ♦ **bits 16 -> 31** Word 28 binary coded
- [Response frame layout]***ulSerialNumberMsb*** Serialnumber - high bits (see page 68/121 of EnDat specifications)
    - ♦ **bits 0 -> 15** Word 29
        - ◊ **bits 0 -> 7** binary coded
        - ◊ **bits 8 -> 15** ASCII digit
- [Response frame layout]***ulModel*** Model/Type of the sensor (see page 79/84 of EnDat application notes 722024)
    - ♦ **0, 1, 2, 3** Incremental linear encoder
    - ♦ **4, 6** Absolute linear encoder
    - ♦ **8, 9, 10, 11** Incremental rotary encoder or angle encoder
    - ♦ **12** Single-turn encoder
    - ♦ **13, 14** Multi-turn encoder
- [Response frame layout]***ulMode*** Provide if sensor allow to use EnDat 2.2 commands
    - ♦ **0** Sensor does not support EnDat 2.2. commands
    - ♦ **1** Sensor supports EnDat 2.2 commands
- [Response frame layout]***ulPositionSize*** Size of the position value send by the sensor (in bits)
- [Response frame layout]***ulSignalPeriod*** Signal period length or signal periods per revolution for incremental output signals (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulStepPerRevolution*** Measuring step length or measuring steps per revolution with serial data transfer (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulNumberOfRevolution*** Distinguishable revolutions - only for multiturn encoders (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulScalingFactor*** Scaling factor for resolution (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulAdditionalData*** Supported additional data (see page 85/121 of EnDat specifications)
    - ♦ **Bit 0** "Position value 2" is available (MRS-Code: 0x42, 0x43, 0x44)
    - ♦ **Bit 1** "Test values" is available (MRS-Code: 0x49, 0x4A, 0x4B)
    - ♦ **Bit 2** "Temperature sensor 1 (external)" is available (MRS-Code: 0x4C)
    - ♦ **Bit 3** "Temperature sensor 2 (external)" is available (MRS-Code: 0x4D)
    - ♦ **Bit 4** "Additional sensors" is available (MRS-Code: 0x4E)
    - ♦ **Bit 16** "Commutation" is available (MRS-Code: 0x51)
    - ♦ **Bit 17** "Acceleration" is available (MRS-Code: 0x52)
    - ♦ **Bit 18** "Limit position signals" is available (MRS-Code: 0x54)
    - ♦ **Bit 19** "Asynchronous position value" is available (MRS-Code: 0x56, 0x57, 0x58)
    - ♦ **Bit 20** "Operating status error sources" is available (MRS-Code: 0x59)
    - ♦ **Bit 23** "Timestamp" is available (MRS-Code: 0x5B)

Example with ROQ 437 2048 5XS08-C4 Sensor

- **Serial Number**

    - **Sensor Serial number** = 35549793 in hex -> 0x**20**021e7261**20**
        - ♦ In bold, ASCII character. In this case spaces (0x20)
        - ♦ 0x0201e7261 = 35549793 in decimal
    - **ulSerialNumberLsb** = 0x1e726120
    - **ulSerialNumberMsb** = 0x00002002

- **ID**

    - **ID number** = 693 641-05 in hex -> 0xA6E79**3035**
        - ♦ In bold, ASCII character. In this case (0 (0x30) and 5 (0x35))
        - ♦ 0xA6E79 = 683641 in decimal
    - **ulIDNumberLsb** = 0x6E793035
    - **ulIDNumberMsb** = 0xA

Example with EQN 1135 512 Sensor

- **Serial Number Example**

    - **Sensor Serial number** = 38473098A in hex -> 0x**20**024B0D8A**41**
        - ♦ In bold, ASCII character. In this case (0x20 and 0x41 (A))
        - ♦ 0x024B0D8A = 38473098 in decimal
    - **ulSerialNumberLsb** = 0x4B0D8A41
    - **ulSerialNumberMsb** = 0x00002002

- **ID**

    - **ID number** = 743 587-01 in hex -> 0xB58A3**3031**
        - ♦ In bold, ASCII character. In this case (0 (0x30) and 1 (0x31))
        - ♦ 0xB58A3 = 743 587 in decimal
    - **ulIDNumberLsb** = 0x58A33031
    - **ulIDNumberMsb** = 0xB

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while selecting memory area 0xA3
- **-8** Error while reading ID number (address 0x8)
- **-9** Error while reading ID number (address 0x9)
- **-10** Error while reading ID number (address 0xA)
- **-11** Error while reading Serialnumber (address 0xB)
- **-12** Error while reading Serialnumber (address 0xC)
- **-13** Error while reading Serialnumber (address 0xD)

- **-14** Error while selecting memory area 0xA1
- **-15** Error while reading encoder model (address 0xE)
- **-16** Error while reading signal period length or signal periods per revolution for incremental output signals (address 0xF)
- **-17** Error while getting the position
- **-18** Error while selecting memory area 0xA3
- **-19** Error while reading signal period length or signal periods per revolution for incremental output signals (address 0x0)
- **-20** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-21** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-22** Error while selecting memory area 0xBD
- **-23** Error while reading scaling factor for resolution (address 0x1B)
- **-24** Error while reading measuring step, or measuring steps per revolution or subdivision values of a grating period (address 0x1C)
- **-25** Error while reading measuring step, or measuring steps per revolution or subdivision values of a grating period (address 0x1D)
- **-26** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-27** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-28** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-29** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-30** Error while reading distinguishable revolutions (address 0x1)
- **-31** Error while selecting memory area 0xBD
- **-32** Error while reading number of distinguishable revolutions with scaling factor (address 0x22)
- **-33** Error while selecting memory area 0xBD
- **-34** Error while reading status of additional datum 1 (address 0x0)
- **-35** Error while reading status of additional datum 2 (address 0x1)
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

		16-bit integer			
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	3700	0x740E	0x0E74
word count	2	16-bit integer	24	0x1800	0x0018

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	51	0x3300	0x0033
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	48	0x30	0x30
ulIDNumberLsb	4	32-bit integer	See the description above	0x????????	0x????????
ulIDNumberMsb	4	32-bit integer	See the description above	0x????????	0x????????
ulSerialNumberLsb	4	32-bit integer	See the description above	0x????????	0x????????
ulSerialNumberMsb	4	32-bit integer	See the description above	0x????????	0x????????
ulModel	4	32-bit integer	See the description above	0x????????	0x????????
ulMode	4	32-bit	See the	0x????????	0x????????

		integer	description above		
ulPositionSize	4	32-bit integer	See the description above	0x????????	0x????????
ulSignalPeriod	4	32-bit integer	See the description above	0x????????	0x????????
ulStepPerRevolution	4	32-bit integer	See the description above	0x????????	0x????????
ulNumberOfRevolution	4	32-bit integer	See the description above	0x????????	0x????????
ulScalingFactor	4	32-bit integer	See the description above	0x????????	0x????????
ulAdditionalData	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 MSXE173x__EndatGetSensorProperties3

Description

Reads the properties of the sensor 3.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Response frame layout]***ulIDNumberLsb*** ID Number - low bits (see page 67/121 of EnDat specifications)
    - **bits 0 -> 15** Word 24 (last 2 digits of the ID number in ASCII format)
    - **bits 16 -> 31** Word 25 (first part of the first 6 digits of the ID number)
- [Response frame layout]***ulIDNumberMsb*** ID Number - high bits (see page 67/121 of EnDat specifications)
    - **bits 0 -> 15** Word 26 (last part of the first 6 digits of the ID number)
- [Response frame layout]***ulSerialNumberLsb*** Serialnumber - low bits (see page 68/121 of EnDat specifications)
    - **bits 0 -> 15** Word 27
        - **bits 0 -> 7** ASCII digit
        - **bits 8 -> 15** binary coded
    - **bits 16 -> 31** Word 28 binary coded
- [Response frame layout]***ulSerialNumberMsb*** Serialnumber - high bits (see page 68/121 of EnDat specifications)
    - **bits 0 -> 15** Word 29
        - **bits 0 -> 7** binary coded
        - **bits 8 -> 15** ASCII digit
- [Response frame layout]***ulModel*** Model/Type of the sensor (see page 79/84 of EnDat application notes 722024)
    - **0, 1, 2, 3** Incremental linear encoder
    - **4, 6** Absolute linear encoder
    - **8, 9, 10, 11** Incremental rotary encoder or angle encoder
    - **12** Single-turn encoder
    - **13, 14** Multi-turn encoder
- [Response frame layout]***ulMode*** Provide if sensor allow to use EnDat 2.2 commands
    - **0** Sensor does not support EnDat 2.2. commands
    - **1** Sensor supports EnDat 2.2 commands
- [Response frame layout]***ulPositionSize*** Size of the position value send by the sensor (in bits)
- [Response frame layout]***ulSignalPeriod*** Signal period length or signal periods per revolution for incremental output signals (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulStepPerRevolution*** Measuring step length or measuring steps per revolution with serial data transfer (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulNumberOfRevolution*** Distinguishable revolutions - only for multiturn encoders (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulScalingFactor*** Scaling factor for resolution (see page 79/84 of EnDat application notes 722024)
- [Response frame layout]***ulAdditionalData*** Supported additional data (see page 85/121 of EnDat specifications)
    - **Bit 0** "Position value 2" is available (MRS-Code: 0x42, 0x43, 0x44)
    - **Bit 1** "Test values" is available (MRS-Code: 0x49, 0x4A, 0x4B)
    - **Bit 2** "Temperature sensor 1 (external)" is available (MRS-Code: 0x4C)
    - **Bit 3** "Temperature sensor 2 (external)" is available (MRS-Code: 0x4D)
    - **Bit 4** "Additional sensors" is available (MRS-Code: 0x4E)
    - **Bit 16** "Commutation" is available (MRS-Code: 0x51)
    - **Bit 17** "Acceleration" is available (MRS-Code: 0x52)
    - **Bit 18** "Limit position signals" is available (MRS-Code: 0x54)

- **Bit 19** "Asynchronous position value" is available (MRS-Code: 0x56, 0x57, 0x58)
- **Bit 20** "Operating status error sources" is available (MRS-Code: 0x59)
- **Bit 23** "Timestamp" is available (MRS-Code: 0x5B)

Example with ROQ 437 2048 5XS08-C4 Sensor

- **Serial Number**
  - **Sensor Serial number** = 35549793 in hex -> 0x**20**021e7261**20**
    - In bold, ASCII character. In this case spaces (0x20)
    - 0x0201e7261 = 35549793 in decimal
  - **ulSerialNumberLsb** = 0x1e726120
  - **ulSerialNumberMsb** = 0x00002002
- **ID**
  - **ID number** = 693 641-05 in hex -> 0xA6E79**3035**
    - In bold, ASCII character. In this case (0 (0x30) and 5 (0x35))
    - 0xA6E79 = 683641 in decimal
  - **ulIDNumberLsb** = 0x6E793035
  - **ulIDNumberMsb** = 0xA

Example with EQN 1135 512 Sensor

- **Serial Number Example**
  - **Sensor Serial number** = 38473098A in hex -> 0x**20**024B0D8A**41**
    - In bold, ASCII character. In this case (0x20 and 0x41 (A))
    - 0x024B0D8A = 38473098 in decimal
  - **ulSerialNumberLsb** = 0x4B0D8A41
  - **ulSerialNumberMsb** = 0x00002002
- **ID**
  - **ID number** = 743 587-01 in hex -> 0xB58A3**3031**
    - In bold, ASCII character. In this case (0 (0x30) and 1 (0x31))
    - 0xB58A3 = 743 587 in decimal
  - **ulIDNumberLsb** = 0x58A33031
  - **ulIDNumberMsb** = 0xB

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while selecting memory area 0xA3

- **-8** Error while reading ID number (address 0x8)
- **-9** Error while reading ID number (address 0x9)
- **-10** Error while reading ID number (address 0xA)
- **-11** Error while reading Serialnumber (address 0xB)
- **-12** Error while reading Serialnumber (address 0xC)
- **-13** Error while reading Serialnumber (address 0xD)
- **-14** Error while selecting memory area 0xA1
- **-15** Error while reading encoder model (address 0xE)
- **-16** Error while reading signal period length or signal periods per revolution for incremental output signals (address 0xF)
- **-17** Error while getting the position
- **-18** Error while selecting memory area 0xA3
- **-19** Error while reading signal period length or signal periods per revolution for incremental output signals (address 0x0)
- **-20** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-21** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-22** Error while selecting memory area 0xBD
- **-23** Error while reading scaling factor for resolution (address 0x1B)
- **-24** Error while reading measuring step, or measuring steps per revolution or subdivision values of a grating period (address 0x1C)
- **-25** Error while reading measuring step, or measuring steps per revolution or subdivision values of a grating period (address 0x1D)
- **-26** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-27** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-28** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x4)
- **-29** Error while reading measuring step length or measuring steps per revolution with serial data transfer (address 0x5)
- **-30** Error while reading distinguishable revolutions (address 0x1)
- **-31** Error while selecting memory area 0xBD
- **-32** Error while reading number of distinguishable revolutions with scaling factor (address 0x22)
- **-33** Error while selecting memory area 0xBD
- **-34** Error while reading status of additional datum 1 (address 0x0)
- **-35** Error while reading status of additional datum 2 (address 0x1)
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

			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	3800	0xD80E	0x0ED8
word count	2	16-bit integer	24	0x1800	0x0018

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	51	0x3300	0x0033
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	48	0x30	0x30
ulIDNumberLsb	4	32-bit integer	See the description above	0x????????	0x????????
ulIDNumberMsb	4	32-bit integer	See the description above	0x????????	0x????????
ulSerialNumberLsb	4	32-bit integer	See the description above	0x????????	0x????????
ulSerialNumberMsb	4	32-bit integer	See the description	0x????????	0x????????

			above		
ulModel	4	32-bit integer	See the description above	0x????????	0x????????
ulMode	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionSize	4	32-bit integer	See the description above	0x????????	0x????????
ulSignalPeriod	4	32-bit integer	See the description above	0x????????	0x????????
ulStepPerRevolution	4	32-bit integer	See the description above	0x????????	0x????????
ulNumberOfRevolution	4	32-bit integer	See the description above	0x????????	0x????????
ulScalingFactor	4	32-bit integer	See the description above	0x????????	0x????????
ulAdditionalData	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 MSXE173x__EndatGetPositionWithAddData0

Description

Reads the position of the sensor 0 with additional data.

Parameters

- [Response frame layout]***ulPositionLow*** Position - low bits (LSB)
- [Response frame layout]***ulPositionHigh*** Position - high bits (MSB)
- [Response frame layout]***ulAddData1*** Value of the additional data 1
- [Response frame layout]***ulAddData2*** Value of the additional data 2

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reading the position
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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
	2		3900	0x3C0F	0x0F3C

Reference number (=register)		16-bit integer			
word count	2	16-bit integer	8	0x0800	0x0008

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	16	0x10	0x10
ulPositionLow	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionHigh	4	32-bit integer	See the description above	0x????????	0x????????
ulAddData1	4	32-bit integer	See the description above	0x????????	0x????????
ulAddData2	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 MSXE173x__EndatGetPositionWithAddData1

Description

Reads the position of the sensor 1 with additional data.

Parameters

- [Response frame layout]***ulPositionLow*** Position - low bits (LSB)
- [Response frame layout]***ulPositionHigh*** Position - high bits (MSB)
- [Response frame layout]***ulAddData1*** Value of the additional data 1
- [Response frame layout]***ulAddData2*** Value of the additional data 2

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reading the position
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	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	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	4000	0xA00F	0x0FA0
word count	2	16-bit integer	8	0x0800	0x0008

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	16	0x10	0x10
ulPositionLow	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionHigh	4	32-bit integer	See the description above	0x????????	0x????????
ulAddData1	4	32-bit integer	See the description above	0x????????	0x????????
ulAddData2	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 MSXE173x__EndatGetPositionWithAddData2

Description

Reads the position of the sensor 2 with additional data.

Parameters

- [Response frame layout]***ulPositionLow*** Position - low bits (LSB)
- [Response frame layout]***ulPositionHigh*** Position - high bits (MSB)
- [Response frame layout]***ulAddData1*** Value of the additional data 1
- [Response frame layout]***ulAddData2*** Value of the additional data 2

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reading the position
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	4100	0x0410	0x1004
word count	2	16-bit integer	8	0x0800	0x0008

Response frame layout

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

			above		
ulPositionHigh	4	32-bit integer	See the description above	0x????????	0x????????
ulAddData1	4	32-bit integer	See the description above	0x????????	0x????????
ulAddData2	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 MSXE173x__EndatGetPositionWithAddData3

Description

Reads the position of the sensor 3 with additional data.

Parameters

- [Response frame layout]***ulPositionLow*** Position - low bits (LSB)
- [Response frame layout]***ulPositionHigh*** Position - high bits (MSB)
- [Response frame layout]***ulAddData1*** Value of the additional data 1
- [Response frame layout]***ulAddData2*** Value of the additional data 2

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reading the position
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	4200	0x6810	0x1068
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		19	0x1300	0x0013

		16-bit integer			
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	16	0x10	0x10
ulPositionLow	4	32-bit integer	See the description above	0x????????	0x????????
ulPositionHigh	4	32-bit integer	See the description above	0x????????	0x????????
ulAddData1	4	32-bit integer	See the description above	0x????????	0x????????
ulAddData2	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 MSXE173x__EndatGetErrorSources0

Description

Reads the error sources of the sensor 0.

The error are reseted after a call to MSXE173x__MFEndatResetErrorBits. If a function returns an error, please use this function to check if it is not a communication error.

Parameters

- [Response frame layout]***ulErrorSrc*** Mask of bits that give the current error sources. Each bit represents an error. If the bit is set to 1, the error is present.
  - **Bit 0** Invalid mode command
  - **Bit 1** Invalid MRS-Code
  - **Bit 2** Transmission is not completed
  - **Bit 3** Communication command is not supported
  - **Bit 4** MRS-Code is not allowed
  - **Bit 6** Invalid address is selected or sensor's EEPROM is written while being busy
  - **Bit 7** Try to write a protected memory place
  - **Bit 8** Write-Protect configuration is tried to be reset (if a memory place is write-protected, it cannot be reset)
  - **Bit 9** Block address is not available
  - **Bit 10** Invalid address for the communication command
  - **Bit 11** Invalid additional data (or additional data not supported by the sensor)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-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
	1		0x03	0x03	0x03

MODBUS Function code		8-bit integer			
Reference number (=register)	2	16-bit integer	4300	0xCC10	0x10CC
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
ulErrorSrc	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

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

Function MSXE173x__EndatGetErrorSources1

Description

Reads the error sources of the sensor 1.

The error are reseted after a call to MSXE173x__MFEndatResetErrorBits. If a function returns an error, please use this function to check if it is not a communication error.

Parameters

- [Response frame layout]***ulErrorSrc*** Mask of bits that give the current error sources. Each bit represents an error. If the bit is set to 1, the error is present.
  - **Bit 0** Invalid mode command
  - **Bit 1** Invalid MRS-Code
  - **Bit 2** Transmission is not completed
  - **Bit 3** Communication command is not supported
  - **Bit 4** MRS-Code is not allowed
  - **Bit 6** Invalid address is selected or sensor's EEPROM is written while being busy
  - **Bit 7** Try to write a protected memory place
  - **Bit 8** Write-Protect configuration is tried to be reset (if a memory place is write-protected, it cannot be reset)
  - **Bit 9** Block address is not available
  - **Bit 10** Invalid address for the communication command
  - **Bit 11** Invalid additional data (or additional data not supported by the sensor)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-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	4400	0x3011	0x1130
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
ulErrorSrc	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 MSXE173x__EndatGetErrorSources2

Description

Reads the error sources of the sensor 2.

The error are reseted after a call to MSXE173x__MFEndatResetErrorBits. If a function returns an error, please use this function to check if it is not a communication error.

Parameters

- [Response frame layout]***ulErrorSrc*** Mask of bits that give the current error sources. Each bit represents an error. If the bit is set to 1, the error is present.
  - **Bit 0** Invalid mode command
  - **Bit 1** Invalid MRS-Code
  - **Bit 2** Transmission is not completed
  - **Bit 3** Communication command is not supported
  - **Bit 4** MRS-Code is not allowed
  - **Bit 6** Invalid address is selected or sensor's EEPROM is written while being busy
  - **Bit 7** Try to write a protected memory place
  - **Bit 8** Write-Protect configuration is tried to be reset (if a memory place is write-protected, it cannot be reset)
  - **Bit 9** Block address is not available
  - **Bit 10** Invalid address for the communication command
  - **Bit 11** Invalid additional data (or additional data not supported by the sensor)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-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	4500	0x9411	0x1194
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
ulErrorSrc	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 MSXE173x__EndatGetErrorSources3

Description

Reads the error sources of the sensor 3.

The error are reseted after a call to MSXE173x__MFEndatResetErrorBits. If a function returns an error, please use this function to check if it is not a communication error.

Parameters

- [Response frame layout]***ulErrorSrc*** Mask of bits that give the current error sources. Each bit represents an error. If the bit is set to 1, the error is present.
  - **Bit 0** Invalid mode command
  - **Bit 1** Invalid MRS-Code
  - **Bit 2** Transmission is not completed

- **Bit 3** Communication command is not supported
- **Bit 4** MRS-Code is not allowed
- **Bit 6** Invalid address is selected or sensor's EEPROM is written while being busy
- **Bit 7** Try to write a protected memory place
- **Bit 8** Write-Protect configuration is tried to be reset (if a memory place is write-protected, it cannot be reset)
- **Bit 9** Block address is not available
- **Bit 10** Invalid address for the communication command
- **Bit 11** Invalid additional data (or additional data not supported by the sensor)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-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	4600	0xF811	0x11F8
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
ulErrorSrc	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 MSXE173x__EndatModbusGetParameter0

Description

Reads the parameter send by the sensor 0.

Before calling this function call the MSXE173x__SensorSendParameter function to select the parameter that you want to read.

Parameters

- [Response frame layout]***ulParam*** Value of the parameter

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	4700	0x5C12	0x125C

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
ulParam	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 MSXE173x__EndatModbusGetParameter1

Description

Reads the parameter send by the sensor 1.

Before calling this function call the MSXE173x__SensorSendParameter function to select the parameter that you want to read.

Parameters

- [Response frame layout]***ulParam*** Value of the parameter

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	4800	0xC012	0x12C0

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
ulParam	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 MSXE173x__EndatModbusGetParameter2

Description

Reads the parameter send by the sensor 2.

Before calling this function call the MSXE173x__SensorSendParameter function to select the parameter that you want to read.

Parameters

- [Response frame layout]***ulParam*** Value of the parameter

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	4900	0x2413	0x1324

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
ulParam	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 MSXE173x__EndatModbusGetParameter3

Description

Reads the parameter send by the sensor 3.

Before calling this function call the MSXE173x__SensorSendParameter function to select the parameter that you want to read.

Parameters

- [Response frame layout]***ulParam*** Value of the parameter

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	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	5000	0x8813	0x1388

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
ulParam	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.

- MXCommon__SetHardwareTriggerFilterTime Register: **100**
- MXCommon__SetHardwareTriggerFilterTimeEx Register: **11000**
- MXCommon__InitAndStartSynchroTimer Register: **101**
- MXCommon__InitAndStartSynchroTimerEx Register: **11050**
- MXCommon__StopAndReleaseSynchroTimer Register: **102**
- MXCommon__StopAndReleaseSynchroTimerEx Register: **11100**
- MXCommon__Reboot Register: **103**
- MXCommon__RebootEx Register: **11150**
- MXCommon__SetCustomerKey Register: **104**
- MXCommon__SetCustomerKeyEx Register: **11200**
- MXCommon__SetFilterChannels Register: **105**
- MXCommon__SetFilterChannelsEx Register: **11250**
- MSXE17xx__MFCommonSetInputsFilter Register: **6000**
- MSXE17xx__MFCommonReferenceVoltageActivation Register: **6050**
- MSXE17xx__MFCommonSetFIFO0Level Register: **6100**
- MSXE17xx__DigitalIOWriteAllChannelsValue Register: **7100**
- MSXE17xx__DigitalIORearmShortCircuit Register: **7150**
- MSXE17xx__DigitalIOInitPort Register: **7200**
- MSXE17xx__IOWatchdogInitAndStart Register: **8050**
- MSXE17xx__IOWatchdogStopAndRelease Register: **8100**

- MSXE173x__EndatInitSensor Register: **2000**
- MSXE173x__EndatSensorReceiveParameter Register: **2100**
- MSXE173x__EndatSelectMemoryArea Register: **2200**
- MSXE173x__EndatSensorSendParameter Register: **2300**
- MSXE173x__EndatSelectAdditionalData Register: **2500**
- MSXE173x__EndatInitAndEnableLatchPositionValues Register: **2600**
- MSXE173x__EndatDisableAndReleaseLatchPositionValues Register: **2700**
- MSXE173x__EndatResetErrorBits Register: **2900**
- MSXE173x__EndatSensorSendPosAndRecvSelMemArea Register: **3000**
- MSXE173x__EndatSensorReceiveReset Register: **5100**

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

Function MXCommon__InitAndStartSynchroTimer

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

Description

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

Parameters:

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

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

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

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

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

- 0: Wait the time overflow to set the synchronisation trigger
- 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	0x????????	0x????????

			above		

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	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-bit integer	16	0x1000	0x0010
byte count	1	8-bit integer	32	0x20	0x20
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulReloadValue	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrOfCycle	4	32-bit integer	See the description above	0x????????	0x????????
ulGenerateTriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????
ulOption03	4	32-bit integer	See the description above	0x????????	0x????????
ulOption04	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
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			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	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	1	8-bit	See	0x??	0x??

code		integer	corresponding chapter		

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

		integer	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	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	0x??	0x??

			chapter		

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	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	103	0x6700	0x0067
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

code		integer	corresponding chapter		

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	0x????????	0x????????

			above		

Response frame layout

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

Exception frame layout

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

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

Function 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
	2		11200	0xC02B	0x2BC0

Reference number (=register)		16-bit integer			
word count	2	16-bit integer	24	0x1800	0x0018
byte count	1	8-bit integer	48	0x30	0x30
bKey	32	8-bit integer array	See the description above	0x??[32]	0x??[32]
bPublicKey	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11200	0xC02B	0x2BC0
word count	2	16-bit integer	24	0x1800	0x0018

Exception frame layout

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

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

Function MXCommon__SetFilterChannels

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

Description

Permit to set a filter per channel

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	24	0x1800	0x0018

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

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	105	0x6900	0x0069
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

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

Function MXCommon__SetFilterChannelsEx

Description

Permit to set a filter per channel

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	23	0x1700	0x0017
unit	1	8-bit	0 or 1	0x00 or	0x00 or

identifier		integer		0x01	0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11250	0xF22B	0x2BF2
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ChannelList	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11250	0xF22B	0x2BF2
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

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

Function MSXE17xx__MFCommonSetInputsFilter

Description

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

Parameters:

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

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

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

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

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

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

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

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

- 0: Filter nicht benutzt
- 1: 100 ns

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

Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - 0 : No error.
  - -1 : Means an system error occured
  - -2 : Multifunction sub module index selection error
  - -3 : Input A filter value selection error
  - -4 : Input B filter value selection error
  - -5 : Input C filter value selection error
  - -6 : Input D filter value selection error
  - -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	27	0x1B00	0x001B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6000	0x7017	0x1770
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulInputAFilterValue	4	32-bit integer	See the description above	0x????????	0x????????
ulInputBFilterValue	4	32-bit integer	See the description above	0x????????	0x????????

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

Response frame layout

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

Exception frame layout

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

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

Function MSXE17xx__MFCommonReferenceVoltageActivation

Description

Permit to activate the reference voltage to pin D-

Parameters:

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

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

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

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

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

Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - ♦ 0 : No error.
  - ♦ -1 : Means an system error occured
  - ♦ -2 : Multifunction sub module index selection error
  - ♦ -3 : Activation flag selection error
  - ♦ -100 : Kernel function error (see syserrno).

Query frame layout

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

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6050	0xA217	0x17A2
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulActivationFlag	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6050	0xA217	0x17A2
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function MSXE17xx__MFCommonSetFIFO0Level

Description

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

Parameters:

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

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

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

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

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

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

Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - ♦ 0 : No error.
  - ♦ -1 : Means an system error occured

- -2 : Multifunction sub module index selection error
- -3 : FIFO level value is wrong
- -4 : Time out time base selection error
- -5 : Time out value can not be null, if a time base is selected
- -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6100	0xD417	0x17D4
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulMFModuleIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulFIFOLevel	4	32-bit integer	See the description above	0x????????	0x????????
ulTimeOutTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulReloadValue	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Exception frame layout

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

Function MSXE17xx__DigitalIOWriteAllChannelsValue

Description

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

Parameters:

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

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7100	0xBC1B	0x1BBC
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulValue	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Exception frame layout

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

Function MSXE17xx__DigitalIORearmShortCircuit

Description

Rearm digital outputs after a short circuit happened.

Parameters:

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

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7150	0xEE1B	0x1BEE
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Exception frame layout

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

Function MSXE17xx__DigitalIOInitPort

Description

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

Parameters:

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

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

- 0 : input
- 1 : output

Returns:

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

Query frame layout

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

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

Response frame layout

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

Exception frame layout

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

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

Function MSXE17xx__IOWatchdogInitAndStart

Description

Init and start the digital output IO watchdog.

Parameters:

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

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

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

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

Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
  - ♦ 0 : No error.
  - ♦ -1 : Means an system error occured
  - ♦ -2:: Time base selection error
  - ♦ -3:: Time value selection error
  - ♦ -100 : Kernel function error (see syserrno).

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function	1	8-bit integer	0x10	0x10	0x10

code					
Reference number (=register)	2	16-bit integer	8050	0x721F	0x1F72
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulTimeValue	4	32-bit integer	See the description above	0x????????	0x????????
ulOption1	4	32-bit integer	See the description above	0x????????	0x????????
ulOption2	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Exception frame layout

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

Function MSXE17xx__IOWatchdogStopAndRelease

Description

Stop and release the digital output watchdog.

Parameters:

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

Returns:

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

Query frame layout

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

			usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	8100	0xA41F	0x1FA4
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Exception frame layout

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

Function MSXE173x__EndatInitSensor

Description

Initialises an EnDat sensor. This function should be called once, in order to call the other EnDat functions.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0
- [Query frame layout]***ulFrequency*** Frequency to use in kHz (500, 900, 1500, 2500, 4500)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The driver is in a wrong state (must be INITIALISED or UNINITIALISED)
- **-6** The component is not programmed as EnDat
- **-7** Error while reseting sensor. Note: If no sensor is plugged on the selected channel, you will get this error.
- **-8** Error while selecting memory area 0xB9
- **-9** Error while reading alarm space (address 0x0)

- **-10** Error while reading warning space (address 0x1)
- **-11** Error while clearing errors (write 0 at address 0x0)
- **-12** Error while clearing warnings (write 0 at address 0x1)
- **-13** Error while selecting memory area 0xA1
- **-14** Error while reading number of clock pulses for transfer of position value (address 0x0D)
- **-15** Error while selecting memory area 0xA5
- **-16** Error while reading EnDat command set (address 0x5)
- **-17** Error while reading support of error messages 1 (address 0x3)
- **-18** Error while reading support of warnings (address 0x4)
- **-19** Error while reading EnDat ordering designation (address 0x8)
- **-20** ulFrequency parameter is too high for current sensor
- **-21** The ulFrequency parameter is wrong
- **-22** The ulFrequency parameter is wrong
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	19	0x1300	0x0013
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2000	0xD007	0x07D0
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulFrequency	4	32-bit integer	See the description	0x????????	0x????????

			above		

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2000	0xD007	0x07D0
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 MSXE173x__EndatSensorReceiveParameter

Description

Writes a parameter to the memory area that was last selected.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function to initialise the sensor, and then MSXE173x__MFEndatSelectMemoryArea, or MSXE173x__MFEndatSensorSendPosAndRecvSelMemArea to select the memory area that contains the parameter you want to write.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0
- [Query frame layout]***ulMrsCode*** The MRS-code corresponding to the last memory area that you have selected (see MSXE173x__MFEndatSelectMemoryArea or MSXE173x__MFEndatSensorSendPosAndRecvSelMemArea)
- [Query frame layout]***ulAddress*** The address of the parameter that you want to write (0x0-0xFF)
- [Query frame layout]***ulParam*** The new value of the parameter

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** The ulMrsCode parameter is wrong
- **-8** The ulAddress parameter is wrong
- **-9** The last selected memory area do not corresponds to the parameter ulMrsCode. Please call MSXE173x__MFEndatSelectMemoryArea or MSXE173x__MFEndatSensorSendPosAndRecvSelMemArea with the wished ulMrsCode before
- **-10** Error while writing parameter
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

			server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	27	0x1B00	0x001B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2100	0x3408	0x0834
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulMrsCode	4	32-bit integer	See the description above	0x????????	0x????????
ulAddress	4	32-bit integer	See the description above	0x????????	0x????????
ulParam	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	2100	0x3408	0x0834
word count	2	16-bit integer	10	0x0A00	0x000A

Exception frame layout

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

Function MSXE173x__EndatSelectMemoryArea

Description

Selects a memory area (see page 31/121 of EnDat specifications).

In order to send or read parameters, the memory area must first be selected.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0
- [Query frame layout]***ulMrsCode*** The MRS-code corresponding to the memory area that you want to

select (see page 31/121 and 84/121 of EnDat specifications)

- **0xB9** Operating status (address area: 0x0 - 0x3)
- **0xA1** Parameters of the encoder manufacturer - first part (address area: 0x4 - 0xF)
- **0xA3** Parameters of the encoder manufacturer - second part (address area: 0x0 - 0xF)
- **0xA5** Parameters of the encoder manufacturer - third part (address area: 0x0 - 0xF)
- **0xA7** Operating parameters (address area: 0x0 - 0xF)
- **0xA9** Parameters of the OEM - first part (address area: depending on the sensor)
- **0xAB** Parameters of the OEM - second part (address area: depending on the sensor)
- **0xAD** Parameters of the OEM - third part (address area: depending on the sensor)
- **0xAF** Parameters of the OEM - fourth part (address area: depending on the sensor)
- **0xB1** Compensation values of the encoder manufacturer - first part (address area: depending on the sensor)
- **0xB3** Compensation values of the encoder manufacturer - second part (address area: depending on the sensor)
- **0xB5** Compensation values of the encoder manufacturer - third part (address area: depending on the sensor)
- **0xB7** Compensation values of the encoder manufacturer - fourth part (address area: depending on the sensor)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** The ulMrsCode parameter is wrong
- **-8** Error while selecting the memory area
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2200	0x9808	0x0898
word count	2	16-bit integer	6	0x0600	0x0006
byte count	1	8-bit integer	12	0x0C	0x0C
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulMrsCode	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	2200	0x9808	0x0898
word count	2	16-bit integer	6	0x0600	0x0006

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

Function MSXE173x__EndatSensorSendParameter

Description

Reads a parameter from the memory area that was last selected.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function to initialise the sensor, and then MSXE173x__MFEndatSelectMemoryArea, or MSXE173x__MFEndatSensorSendPosAndRecvSelMemArea to select the memory area that contains the parameter you want to read.

Then call the function MSXE173x__EndatModbusGetParameter0, MSXE173x__EndatModbusGetParameter1, MSXE173x__EndatModbusGetParameter2 or MSXE173x__EndatModbusGetParameter3 to read the value of the parameter.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0
- [Query frame layout]***ulMrsCode*** The MRS-code corresponding to the last memory area that you have selected (see MSXE173x__MFEndatSelectMemoryArea or MSXE173x__MFEndatSensorSendPosAndRecvSelMemArea)
- [Query frame layout]***ulAddress*** The address of the parameter that you want to read (0x0-0xFF)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working

- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** The ulMrsCode parameter is wrong
- **-8** The ulAddress parameter is wrong
- **-9** Your sensor is not compatible with EnDat 2.2, but the parameter ulMrsCode is only available for EnDat 2.2
- **-10** The last selected memory area do not corresponds to the parameter ulMrsCode. Please call MSXE173x__MFEndatSelectMemoryArea or MSXE173x__MFEndatSensorSendPosAndRecvSelMemArea with the wished ulMrsCode before
- **-11** Error while reading parameter
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2300	0xFC08	0x08FC
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulMrsCode	4	32-bit integer	See the description above	0x????????	0x????????

ulAddress	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	2300	0xFC08	0x08FC
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

code		integer	corresponding chapter		

Function MSXE173x__EndatSelectAdditionalData

Description

Selects the additional data that will be sent by the sensor.

Some additional data are not available on all sensors. To get the available additional data of your sensor, please use the function MSXE173x__MFEndatGetSensorProperties.

If you select an additional data that is not available on your sensor, you will get the parameter ucErrorSrc13 set to 1 when calling the function MSXE173x__MFEndatGetErrorSources.

The additional data are extra values that the sensor can send (in the same cycle as its position value).

This function is reserved for EnDat 2.2 sensors. It will returns an error if the sensor does not support EnDat 2.2 commands.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0
- [Query frame layout]***ulAddDataCount*** The number of selected additional data (0 to 2)
- [Query frame layout]***ulMrsCodeAD1***The MRS-Code for the first additional data
  - **0x40** Send additional info 1 without data contents
  - **0x42** Position value 2 word 1 LSB
  - **0x43** Position value 2 word 2
  - **0x44** Position value 2 word 3 MSB
  - **0x49** Test values word 1 LSB
  - **0x4A** Test values word 2
  - **0x4B** Test values word 3 MSB
  - **0x4C** Temperature sensor 1 (external)
  - **0x4D** Temperature sensor 2 (external)
  - **0x4E** Additional sensors
- [Query frame layout]***ulMrsCodeAD2*** The MRS-Code for the second additional data
  - **0x50** Send additional datum 2 without data contents
  - **0x51** Commutation
  - **0x52** Acceleration
  - **0x54** Limit position signals
  - **0x56** Asynchronous position value word 1 LSB
  - **0x57** Asynchronous position value word 2
  - **0x58** Asynchronous position value word 3 MSB
  - **0x59** Operating status error sources
  - **0x5B** Timestamp

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** The ulAddDataCount parameter is wrong
- **-8** The ulMrsCodeAD1 parameter is wrong
- **-9** The ulMrsCodeAD2 parameter is wrong
- **-10** The sensor is not compatible with EnDat 2.2
- **-11** Error while deactivating additional data 2
- **-12** Error while deactivating additional data 1
- **-13** Error while activating additional data 1
- **-14** Error while deactivating additional data 2
- **-15** Error while deactivating additional data 1
- **-16** Selected additional data 1 is wrong or not available on this sensor. Please call MSXE173x__MFEndatGetErrorSources to get more information
- **-17** Error while activating additional data 1. Please call MSXE173x__MFEndatGetErrorSources to get more information
- **-18** Error while getting the current position value
- **-19** Error while activating additional data 2
- **-20** Error while deactivating additional data 2
- **-21** Error while deactivating additional data 1
- **-22** Selected additional data 2 is wrong or not available on this sensor. Please call MSXE173x__MFEndatGetErrorSources to get more information
- **-23** Error while activating additional data 2. Please call MSXE173x__MFEndatGetErrorSources to get more information
- **-24** Error while getting the current position value
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

MODBUS Function code		8-bit integer			
Reference number (=register)	2	16-bit integer	2500	0xC409	0x09C4
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulAddDataCount	4	32-bit integer	See the description above	0x????????	0x????????
ulMrsCodeAD1	4	32-bit integer	See the description above	0x????????	0x????????
ulMrsCodeAD2	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	2500	0xC409	0x09C4
word count	2		10	0x0A00	0x000A

		16-bit integer			

Exception frame layout

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

Function MSXE173x__EndatInitAndEnableLatchPositionValues

Description

Initialises and enables the latch logic to get the position and additional informations from the EnDat sensor using a trigger source.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

When the selected trigger occurs, the position value of the selected EnDat channel will be measured and send through the dataserver.

If you chose additional data using the function MSXE173x__MFEndatSelectAdditionalData and if your selected ulDataFormat enables it, you will also receive the value of the selected additional data.

Note: using a synchro timer (and activating the synchro trigger as latch source) enables to get postion of the EnDat sensor at a defined rate.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0

- [Query frame layout]***ulLatchSource*** Mask of bits, that defines the trigger source.
  - *Bit 0: Hardware trigger* If you select the hardware trigger, you must also choose the edge detection (Bit 2 and 3). Of course, you can choose both.
  - *Bit 1 : Synchro trigger*
  - *Bit 2 : Trigger rising edge* (only if hardware trigger is selected)
  - *Bit 3 : Trigger falling edge* (only if hardware trigger is selected)
- [Query frame layout]***ulTriggerEdgeCount*** Number of edges required to detect an hardware trigger (only if hardware trigger is selected)
- [Query frame layout]***ulDataFormat*** Mask of bits, that defines the format of data that will be sent by the data server.
  - *Bit 0: Timestamp*
  - *Bit 1: Digital I/Os state*
  - *Bit 2: Additional data 1*
  - *Bit 3: Additional data 2*
  - *Bit 4: Format of the value*
    - **0** Raw value (as sent by the sensor)
    - **1** Standardised value. The format of the frame will then depends on the model of the sensor. See system documentation.

Frame sent by the dataserver depending on dataformat

Basic frame (ulDataformat = 0)

- unsigned long eventsrc
  - High 16 bits (MSB) : Channel concerned (0 to 3)
  - Low 16 bits (LSB) : Bit mask representing the source of the event unsigned long eventsrc
    - Bit 0: Hardware trigger
    - Bit 1: Synchro trigger
    - Bit 2: Compare
- unsigned long positionLow (Low bits of the position)
- unsigned long positionHigh (High bits of the position)
- unsigned long error (Status of the communication. Same value as the value returned by the function MSXE173x__MFEncatGetErrorSources)

If ulDataformat bit 0 is set (send timestamp). At the end of the Basic frame, we add the timestamp

- unsigned long ts (Second part of the timestamp)
- unsigned long tus (Microsecond part of the timestamp)

If ulDataformat bit 1 is set (send digital I/Os state). At the end of the Basic frame, we add the state of the digital I/Os.

- unsigned long digiostate

If ulDataformat bit 2 is set (send additional data 1). At the end of the Basic frame, we add the state of the first additional data.

- unsigned long ad1value

If ulDataformat bit 3 is set (send additional data 2). At the end of the Basic frame, we add the state of the second additional data.

- unsigned long ad2value

The bit 4 of the ulDataFormat is more complex. The format of the frame that is sent by the dataserver will depend on the model of the sensor used. See the system documentation for more information, or the "Acquisition" menu of the web site.

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** The ulLatchSource parameter is wrong
- **-8** The ulDataFormat parameter is wrong
- **-9** Error while getting sensor properties
- **-10** The multiturn part size of the sensor is 0
- **-11** The singleturn part size of the sensor is 0
- **-12** The step per revolution properties of the sensor is 0
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	27	0x1B00	0x001B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2600	0x280A	0x0A28
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulConnectorIndex	4	32-bit integer	See the description	0x????????	0x????????

			above		
ulChannelIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulLatchSource	4	32-bit integer	See the description above	0x????????	0x????????
ulTriggerEdgeCount	4	32-bit integer	See the description above	0x????????	0x????????
ulDataFormat	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	2600	0x280A	0x0A28
word count	2	16-bit integer	10	0x0A00	0x000A

Exception frame layout

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

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

Function MSXE173x__EndatDisableAndReleaseLatchPositionValues

Description

Disables and releases the latch logic started with the function MSXE173x__MFEndatInitAndEnableLatchPositionValues.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be LATCH_RUNNING or LATCH_FIFO_OVERFLOW)
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

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

			usually 0		
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	2700	0x8C0A	0x0A8C
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	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	2700	0x8C0A	0x0A8C
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 MSXE173x__EndatResetErrorBits

Description

Resets the error bits.

It can be used before each command in order to get (after the call of the command) the status of the system using MSXE173x__MFEndatGetErrorSources.

Once an error is detected using MSXE173x__MFEndatGetErrorSources, you have to clear it using this function.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2900	0x540B	0x0B54
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	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	2900	0x540B	0x0B54
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 MSXE173x__EndatSensorSendPosAndRecvSelMemArea

Description

Reads the position value of the sensor and selects a memory area in the same cycle.

In order to send or read parameters, the memory area must first be selected.

Before calling this function, you must call the MSXE173x__MFEndatInitSensor function.

This function differs from the MSXE173x__MFEndatSelectMemoryArea function, since it can select memory areas that are reserved for EnDat 2.2. sensors.

This function is reserved for EnDat 2.2 sensors. It will returns an error if the sensor does not support EnDat 2.2 commands.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0
- [Query frame layout]***ulMrsCode*** The MRS-code corresponding to the memory area that you want to select (see page 31/121 and 84/121 of EnDat specifications)
    - **0xB9** Operating status (address area: 0x0 - 0x3)
    - **0xA1** Parameters of the encoder manufacturer - first part (address area: 0x4 - 0xF)
    - **0xA3** Parameters of the encoder manufacturer - second part (address area: 0x0 - 0xF)
    - **0xA5** Parameters of the encoder manufacturer - third part (address area: 0x0 - 0xF)
    - **0xA7** Operating parameters (address area: 0x0 - 0xF)
    - **0xA9** Parameters of the OEM - first part (address area: depending on the sensor)
    - **0xAB** Parameters of the OEM - second part (address area: depending on the sensor)
    - **0xAD** Parameters of the OEM - third part (address area: depending on the sensor)
    - **0xAF** Parameters of the OEM - fourth part (address area: depending on the sensor)
    - **0xB1** Compensation values of the encoder manufacturer - first part (address area: depending on the sensor)
    - **0xB3** Compensation values of the encoder manufacturer - second part (address area: depending on the sensor)
    - **0xB5** Compensation values of the encoder manufacturer - third part (address area: depending on the sensor)
    - **0xB7** Compensation values of the encoder manufacturer - fourth part (address area: depending on the sensor)
    - **0xBD** Parameters of the encoder manufacturer for EnDat 2.2 (address area: 0x0 - 0x3F)
    - **0xBB** Operating parameters 2 (address area: depending on the sensor)
    - **0xBF** Parameters of the section 2 memory area (address area: depending on the sensor)
- [Query frame layout]***ulAddress*** Selected block address (only used if ulMrsCode is set to 0xBF. If ulMrsCode is not 0xBF, set it to 0). It is used to address further section 2 memory areas (see page 46/121 of EnDat specifications)

Returns

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

- **0** The remote function performed OK
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** The ulMrsCode parameter is wrong
- **-8** The ulAddress parameter is wrong
- **-9** The sensor is not compatible with EnDat 2.2
- **-10** Error while getting position and selecting memory area
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-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	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	3000	0xB80B	0x0BB8
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulMrsCode	4	32-bit integer	See the description above	0x????????	0x????????
ulAddress	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	3000	0xB80B	0x0BB8
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function MSXE173x__EndatSensorReceiveReset

Description

This function has the same effect as an hardware reboot of the sensor.

Parameters

- [Query frame layout]***ulConnectorIndex*** Index of the EnDat connector (0 to 3). See on the MSX-E system.
- [Query frame layout]***ulChannelIndex*** Index of the channel. Set to 0

Returns

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

- **0** The remote function performed OK
- **-1** System error occured
- **-2** The PLD is not working
- **-3** The ulConnectorIndex parameter is wrong
- **-4** The ulChannelIndex parameter is wrong
- **-5** The component is not programmed as EnDat
- **-6** The driver is in a wrong state (must be INITIALISED)
- **-7** Error while reseting sensor
- **-41** Transmission error. Please call MSXE173x__EndatGetErrorSourcesX to get more information
- **-100** Internal system error occurred. See value of syserrno

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	5100	0xEC13	0x13EC
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1	8-bit integer	8	0x08	0x08
ulConnectorIndex	4	32-bit integer	See the description above	0x????????	0x????????
ulChannelIndex	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	5100	0xEC13	0x13EC
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??

FC23 (read/write registers) Functions

Top

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

Query frame layout

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

Response frame layout

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

Exception frame layout

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

Exception code description

Top

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

Siemens Step 7 compatibility information (AWL/SDF code)

Top

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

Function/Parameter	Renamed as
MXCommon__GetModuleType	GetModuleType
MXCommon__GetTime	GetTime
MXCommon__TestCustomerID	TestCustomerID
MSXE17xx__DigitalIOReadAllChannelsValue	17xx_DigIOReadAll
MSXE17xx__DigitalIOTestShortCircuit	17xx_DigIOTestShortC
MSXE17xx__IOWatchdogGetStatusAndValue	17xx_IOWatchdogGet
MSXE173x__EndatGetPosition0	173MFEndGetPos0
MSXE173x__EndatGetPosition1	173MFEndGetPos1
MSXE173x__EndatGetPosition2	173MFEndGetPos2
MSXE173x__EndatGetPosition3	173MFEndGetPos3
MSXE173x__EndatGetSensorProperties0	173MFEndGetSensProp0
MSXE173x__EndatGetSensorProperties1	173MFEndGetSensProp11
MSXE173x__EndatGetSensorProperties2	173MFEndGetSensProp2
MSXE173x__EndatGetSensorProperties3	173MFEndGetSensProp3
MSXE173x__EndatGetPositionWithAddData0	173MFEndGetPWAddData0
MSXE173x__EndatGetPositionWithAddData1	173MFEndGetPWAddData1
MSXE173x__EndatGetPositionWithAddData2	173MFEndGetPWAddData2
MSXE173x__EndatGetPositionWithAddData3	173MFEndGetPWAddData3
MSXE173x__EndatGetErrorSources0	173MFEndGetErrSrc0
MSXE173x__EndatGetErrorSources1	173MFEndGetErrSrc1
MSXE173x__EndatGetErrorSources2	173MFEndGetErrSrc2
MSXE173x__EndatGetErrorSources3	173MFEndGetErrSrc3
MSXE173x__EndatModbusGetParameter0	173MFEndModGetPar0
MSXE173x__EndatModbusGetParameter1	173MFEndModGetPar1
MSXE173x__EndatModbusGetParameter2	173MFEndModGetPar2
MSXE173x__EndatModbusGetParameter3	173MFEndModGetPar3
MXCommon__SetHardwareTriggerFilterTime	SetHwTrigFiltTime
MXCommon__InitAndStartSynchroTimer	InitStartSyncTimer
MXCommon__StopAndReleaseSynchroTimer	StopRelSyncTimer
MXCommon__Reboot	Reboot
MXCommon__SetCustomerKey	SetCustomerKey
MXCommon__SetFilterChannels	SetFilterChannels
MSXE17xx__MFCommonSetInputsFilter	17xx_MFCSetInputFilter
MSXE17xx__MFCommonReferenceVoltageActivation	17xx_MFCRefVoltActiv
MSXE17xx__MFCommonSetFIFO0Level	17xx_MFCSetFIFO0Level
MSXE17xx__DigitalIOWriteAllChannelsValue	17xx_DigIOWriteAll

MSXE17xx__DigitalIORearmShortCircuit	17xx_DigIORearm
MSXE17xx__DigitalIOInitPort	17xx_DigIOInitPort
MSXE17xx__IOWatchdogInitAndStart	17xx_IOWatchdogStart
MSXE17xx__IOWatchdogStopAndRelease	17xx_IOWatchdogStop
MSXE173x__EndatInitSensor	173MFEndInitSens
MSXE173x__EndatSensorReceiveParameter	173MFEndSensRcvParam
MSXE173x__EndatSelectMemoryArea	173MFEndSelectMemArea
MSXE173x__EndatSensorSendParameter	173MFEndSenParameter
MSXE173x__EndatSelectAdditionalData	173MFEndSelectAddData
MSXE173x__EndatInitAndEnableLatchPositionValues	173MFEndIniEnaPosVal
MSXE173x__EndatDisableAndReleaseLatchPositionValues	173MFEndDisRelPosVal
MSXE173x__EndatResetErrorBits	173MFEndResErrSou
MSXE173x__EndatSensorSendPosAndRecvSelMemArea	173MFEndPosRcvMemAre
MSXE173x__EndatSensorReceiveReset	173MFEndRcvReset